Головоломки на PHP для хакера

Максим Кузнецов, Игорь Симдянов

+CD

Максим Кузнецов
Игорь Симдянов

ГОЛОВОЛОМКИ на PHP для ХАКЕРА

Санкт-Петербург
«БХВ-Петербург»
2006

УДК 681.3.06
ББК 32.973.26-018.1
К89

Кузнецов, М. В.

К89 Головоломки на PHP для хакера / М. В. Кузнецов, И. В. Симдянов. — СПб.: БХВ-Петербург, 2006. — 464 с.: ил.

ISBN 5-94157-837-7

Книга представляет собой задачник по Web-технологиям с уклоном в защиту Web-приложений от злоумышленников. Цель книги — помочь Web-разработчику научиться самостоятельно обнаруживать и устранять уязвимости в своем коде. На компакт-диске, поставляемом вместе с книгой, приведены скрипты, являющиеся ответами на предлагаемые задачи.

Для программистов и Web-разработчиков

УДК 681.3.06
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Григорий Добин*
Редактор	*Ирина Иноземцева*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Виктория Пиотровская*
Дизайн серии	*Инны Тачиной*
Оформление обложки и фото	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 24.04.06.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 37,41.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-837-7

Оглавление

Введение

Предлагаемая книга является сборником задач по PHP с уклоном в защиту сайта и Web-приложений от злоумышленников.

Основная проблема создателей Web-приложений заключается в том, что они мыслят совсем другими категориями, нежели злоумышленники. Кроме того, Web-разработчики редко прибегают к тестированию своих разработок на предмет уязвимости, так как им подсознательно не хочется ломать свои собственные Web-приложения. Снять такой настрой поможет эта книга, где, наряду с задачами по защите Web-приложений, будет предложено большое количество задач по взлому сайта с применением самых различных технологий, от межсайтового скриптинга и SQL-инъекций до подбора паролей при помощи словаря. Это позволит читателю убедиться в том, как легко может быть нарушена работа Web-сайта и как дорого может обернуться беспечность при его разработке.

Наряду с "деструктивными" задачами будет предложено большое количество заданий, направленных на построение обороны сайта. Выполнив задания, вы получите в руки мощную систему защиты собственного сайта, которая будет отличаться от коммерческих и свободных аналогов тем, что вы будете знать в ней каждый винтик и сможете легко модернизировать ее, быстро устранять последствия взлома и находить уязвимости.

Книга разбита на две части: непосредственно задачник и ответы на задачи. Вы можете решать все задачи последовательно или, если вам необходимо срочно защитить свой сайт, можете воспользоваться готовыми кодами, находящимися на прилагаемом к книге компакт-диске. Коды примеров можно также загрузить с сайта IT-студии SoftTime по адресу **http://www.softtime.ru/security/**.

По всем вопросам, возникающим по мере чтения книги, вы можете обращаться на форум, расположенный на Web-сайте IT-студии SoftTime, сотрудниками которой являются авторы книги (**http://www.softtime.ru/forum/**).

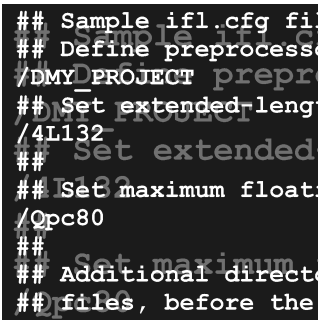
Авторы присутствуют на форуме каждый день и с удовольствием ответят на ваши вопросы.

Благодарности

Авторы благодарят сотрудника отдела разработки программного обеспечения средств связи IT-студии SoftTime Ломалова В. П. за помощь в написании *Приложения 1.*

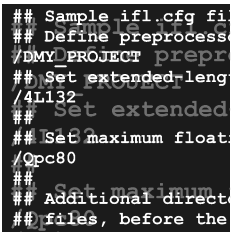
Авторы также выражают большую признательность следователю УФСБ РФ по Нижегородской области Зайченко Д. А. и старшему оперуполномоченному УФСБ РФ по Нижегородской области Новикову В. Б. за ценные консультации, которые они оказывали авторам при написании *Приложения 2.*

Авторы благодарны сотрудникам издательства "БХВ-Петербург", усилиями которых эта рукопись увидела свет, и посетителям форума **http://www.softtime.ru/forum/** за интересные вопросы и конструктивное обсуждение.



Часть I

ЗАДАЧИ



Глава I.1

Строки

Работа со строками составляет основу любого программирования. Виртуозное манипулирование строками позволит программисту создавать более короткие и эффективные программы. Исследования показали, что плотность ошибок в программах не зависит от языка программирования, а зависит только от квалификации программиста. Чем короче будут программы, тем меньше ошибок и уязвимостей в них будет. Хорошее знание особенностей строк позволяет безошибочно определять возможные проблемные с точки зрения безопасности места в коде. Данная глава содержит задачи на знание строковых функций PHP и умение обращаться с ними.

Замечание

Все примеры из данной главы можно найти в каталоге scripts\1 компакт-диска, поставляемого вместе с книгой.

I.1.1. Количество и имена файлов в произвольном каталоге

Определите количество и имена файлов в каталоге, не прибегая к функциям работы с каталогами. Решение задачи основано на том факте, что в PHP существует несколько видов кавычек, каждый из которых обладает своими свойствами.

I.1.2. Выравнивание по правому краю

Пусть есть список файлов в массиве (листинг I.1.1). У имен файлов может быть различная длина, и необходимо выровнять их по правому краю так, как это изображено на рис. I.1.1. Для решения задачи не разрешается

прибегать к атрибуту `align` и CSS, можно использовать только теги `<pre>` и `</pre>`.

Листинг I.1.1. Массив `$filename` с именами файлов

```
<?php

  $filename = array("all.php", "auth.php",
                    "auth.txt", "base.txt",
                    "chat.html", "config.php",
                    "count.txt", "count_new.txt",
                    "counter.dat", "counter.php",
                    "create.php", "dat.db");

?>
```

Замечание

Файл с массивом можно найти на прилагаемом к книге компакт-диске (scripts\1\1.2\1.php).

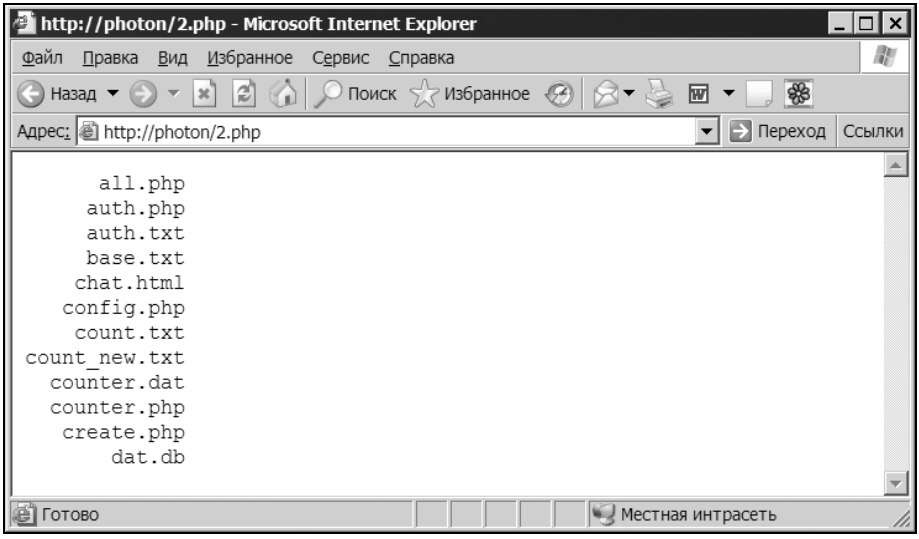


Рис. I.1.1. Выравнивание имен файлов по правому краю

I.1.3. Выравнивание по левому и правому краям

Необходимо разбить массив `$filename` (листинг I.1.1) на две части и вывести в виде двух колонок так, как это представлено на рис. I.1.2. Для решения задачи не разрешается прибегать к атрибуту `align` и CSS, можно использовать только теги `<pre>` и `</pre>`.

Рис. I.1.2. Выравнивание имен файлов по левому и правому краям

I.1.4. Вывод данных в три столбца

Часто перед Web-разработчиками встает задача вывода таблицы, содержащей несколько столбцов. Выведите имена файлов из массива `$filename` (листинг I.1.1) двумя способами, представленными на рис. I.1.3 и I.1.4 соответственно. При решении этой задачи необходимо динамически сформировать HTML-таблицу.

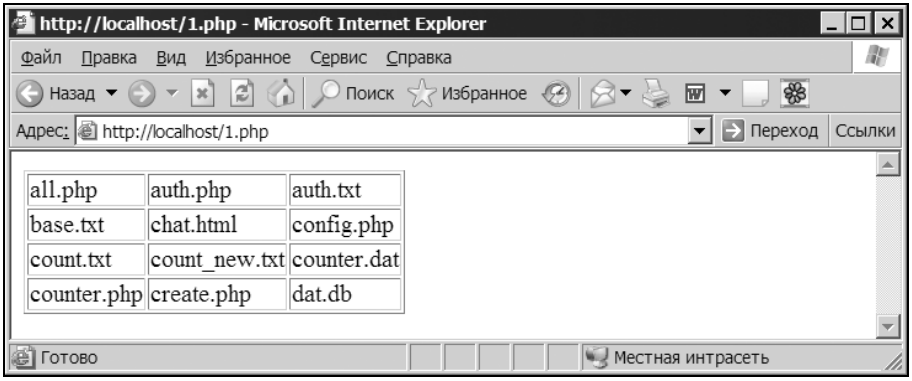


Рис. I.1.3. Первый вариант вывода массива в три столбца

Рис. I.1.4. Второй вариант вывода массива в три столбца

I.1.5. Передача массива между двумя страницами

Пусть массив `$filename`, представленный в листинге I.1.1, определен на странице first.php. Отобразите его на странице second.php, используя инструкцию `include`.

I.1.6. Передача массива методом GET

Пусть массив `$filename`, представленный в листинге I.1.1, определен на странице first.php. Отобразите его на странице second.php, используя для передачи метод GET.

I.1.7. Передача массива методом POST

Пусть массив `$filename`, представленный в листинге I.1.1, определен на странице first.php. Отобразите его на странице second.php, используя для передачи метод POST.

I.1.8. Передача массива через сессии

Пусть массив `$filename`, представленный в листинге I.1.1, определен на странице first.php. Отобразите его на странице second.php, используя сессии.

I.1.9. Передача массива через cookies

Пусть массив `$filename`, представленный в листинге I.1.1, определен на странице first.php. Отобразите его на странице second.php, используя cookies.

I.1.10. Календарь

Создайте календарь на текущий месяц в двух форматах: американском (рис. I.1.5) и российском (рис. I.1.6).

Субботу и воскресенье необходимо подсветить красным цветом.

Рис. I.1.5. Календарь на текущий месяц в американском формате

Рис. I.1.6. Календарь на текущий месяц в российском формате

I.1.11. Вертикальный вывод строки

Выведите строку "Hello world!" вертикально, так, как это представлено на рис. I.1.7.

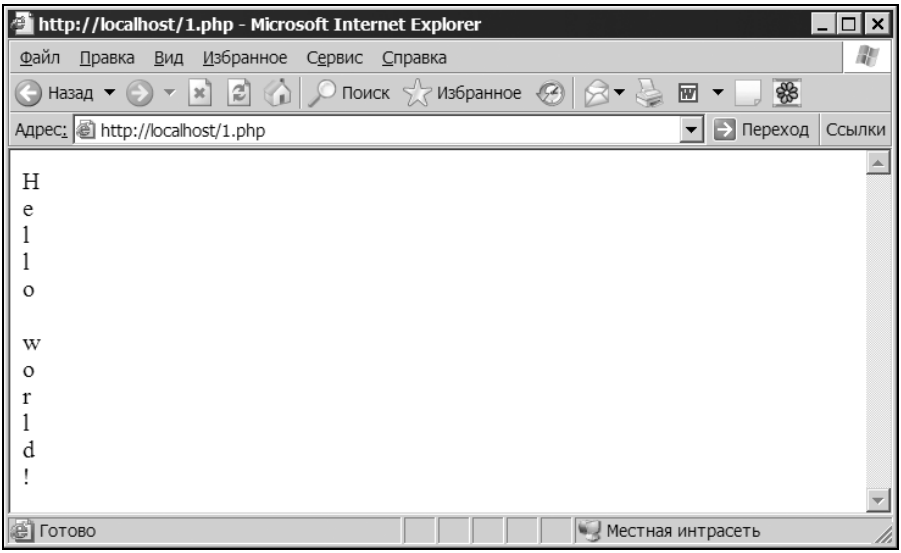


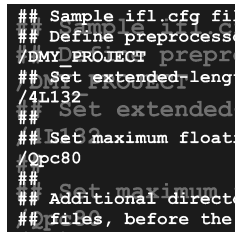
Рис. I.1.7. Вертикальный вывод строки

I.1.12. Число в денежном формате

Пусть имеется число 18439529234.5678, его необходимо представить в денежном формате, т. е. чтобы после запятой осталось только два знака, а триады были бы разделены пробелом — 18 439 529 234.57.

I.1.13. Замена символов bbCode

Замените в тексте "Очень [b]жирный[/b], жирный [b]текст" символы bbCode [b] и [/b] на их HTML-эквиваленты и , не прибегая к регулярным выражениям. То есть для решения задачи должны быть использованы только строковые функции.



Глава I.2

Регулярные выражения

Регулярные выражения являются мини-языком. Сложную задачу можно решить двумя способами: либо создав сложное решение, используя простые технологии, либо создав простое решение, используя сложную технологию. Точно так же и с регулярными выражениями — изучить их достаточно сложно, но, поняв их один раз, далее в одну строку можно решать задачи, для решения которых при помощи строковых функций может понадобиться сотня строк. В *главе 1* было сказано, что плотность ошибок тем меньше, чем короче программа — регулярные выражения позволяют создавать не просто короткие программы, а очень короткие. Данная глава содержит задачи на различные регулярные выражения.

Замечание

Все примеры из данной главы можно найти в каталоге scripts\2 компакт-диска, поставляемого вместе с книгой.

I.2.1. Удаление всех тегов из HTML-страницы

На компакт-диске найдите HTML-страницу scripts\2\index.htm. Прочитайте содержимое страницы и удалите все HTML-теги, оставив только полезный текст. Текст необходимо вывести в окно браузера (рис. I.2.1).

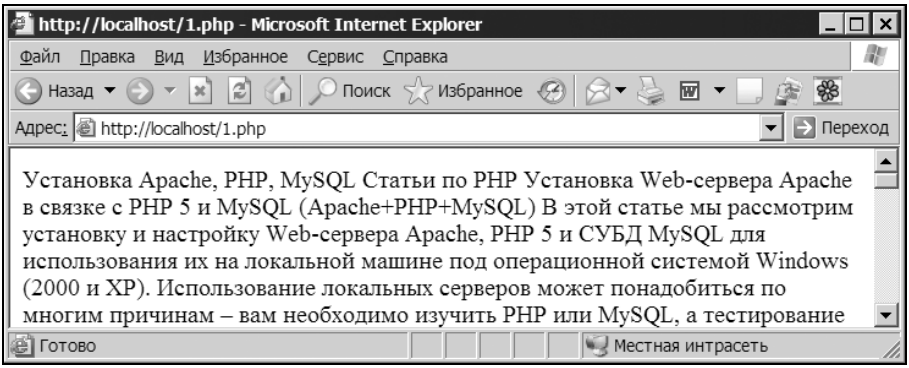


Рис. I.2.1. Чистый текст, извлеченный из HTML-страницы index.htm

I.2.2. Удаление изображений из HTML-страницы

На компакт-диске найдите HTML-страницу scripts\2\index.htm. Прочитайте содержимое страницы и удалите HTML-теги `<img>`.

I.2.3. Извлечение названия HTML-страницы

На компакт-диске найдите HTML-страницу scripts\2\index.htm. Извлеките название страницы, которое помещается между тегами `<title>` и `</title>`.

I.2.4. Конвертация даты из MySQL-формата в календарный формат

Используя регулярные выражения, переконвертируйте дату из формата 2003-03-21 в формат 21.03.2003.

I.2.5. Проверка корректности ввода адреса электронной почты

Разработайте HTML-форму, обработчик которой будет проверять корректность ввода адреса электронной почты (рис. I.2.2).

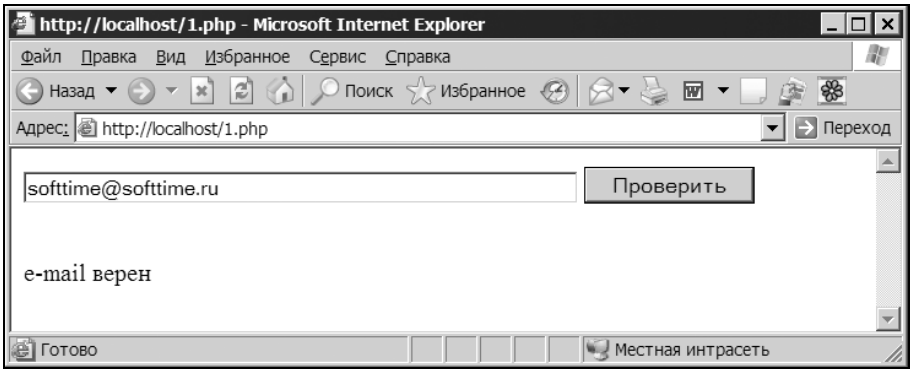


Рис. I.2.2. HTML-форма проверки адреса электронной почты

I.2.6. Проверка корректности ввода URL

Разработайте HTML-форму, обработчик которой будет проверять корректность ввода адреса Web-сайта. Допускается ввод как с указанием протокола, например, **http://www.softtime.ru**, так и без него, например, **www.softtime.ru**. Следует учитывать, что адрес может содержать путь после доменного имени, а также параметры, например, **http://www.softtime.ru/php5/index.php?id_article=43**.

I.2.7. Подсветка URL

Часто возникает задача превращения текстовой ссылки в гиперссылку. На компакт-диске найдите текстовый файл scripts\2\text.txt и выведите его содержимое в окно браузера, преобразовав все URL в гиперссылки.

I.2.8. Проверка корректности ввода чисел

Создайте HTML-форму, состоящую из двух текстовых полей, в первом из которых вводится количество товарных позиций, а во втором их цена в формате ###.##. Обработчик формы должен проверить, является ли введенная в первом поле информация целым числом, а во втором — удовлетворяющим денежному формату. Если все верно, необходимо вывести произведение этих двух чисел.

I.2.9. Изменение регистра

Пусть имеется фраза "**ПРОГРАММИРОВАНИЕ — это ИСКУССТВО. Ему и ЖИЗНЬ посвятить не жалко**". Создайте скрипт и регулярное выражение, которое заменит все слова в верхнем регистре на слова, начинающиеся с заглавной буквы: "**Программирование — это Искусство. Ему и Жизнь посвятить не жалко**".

I.2.10. Разбивка длинной строки

При построении различных Web-приложений, главным образом гостевых книг, форумов и чатов часто возникает необходимость защиты дизайна страниц от длинных последовательностей символов, которые могут исказить дизайн. Создайте функцию, разбивающую на части все слова, длина которых превышает 25 символов.

I.2.11. Разбивка HTML-страницы на предложения

На компакт-диске найдите HTML-страницу scripts\2\index.htm. Прочитайте содержимое страницы и поместите каждое предложение текста в элементы массива `$text` так, чтобы первое предложение оказалось в элементе с индексом 0 — `$text[0]`, второе в элементе с индексом 1 — `$text[1]` и т. д. После чего в цикле преобразуйте массив `$text` в двумерный массив таким образом, чтобы в элементе `$text[0][0]` хранилось первое слово первого предложения, в элементе `$text[0][1]` хранилось второе слово первого предложения и т. д. Проконтролируйте результаты работы, отправив дамп массива в окно браузера при помощи функции `print_r()`.

I.2.12. Количество слов в тексте

На компакт-диске найдите HTML-страницу scripts\2\index.htm. Прочитайте содержимое страницы и сосчитайте, сколько в нем содержится одно-, двух-, ..., десятибуквенных слов.

I.2.13. Интерпретация тегов bbCode

В Интернете большое распространение получили теги в квадратных скобках, именуемые так же, как теги в стиле phpBB (известного и широко распространенного форума). Удобство использования таких тегов заключается в том, что все теги HTML можно запретить, преобразуя их при помощи функции `htmlspecialchars()` в безопасную форму, и в то же время разрешить посетителям использовать их эквиваленты. Например, `[i]` вместо `<i>` и `[code]` вместо `<code>`. Теги в квадратных скобках можно заменить на теги в угловых скобках уже после преобразования текста при помощи функции `htmlspecialchars()`.Чаще всего прибегают к тегам `[url]`, которые имеют следующий синтаксис:

```
[url = ссылка] имя ссылки [/url]
```

При выводе на страницу этот шаблон следует преобразовать в

```
<a href=ссылка>имя ссылки</a>
```

Если используется форма тега

```
[url]ссылка[/url]
```

то на страницу выводится гиперссылка вида:

```
<a href=ссылка>ссылка</a>
```

На компакт-диске найдите HTML-страницу scripts\2\bb.txt, содержимое этой страницы представлено на рис. I.2.3.

Необходимо преобразовать все имеющиеся на странице теги в их HTML-эквиваленты (рис. I.2.4).

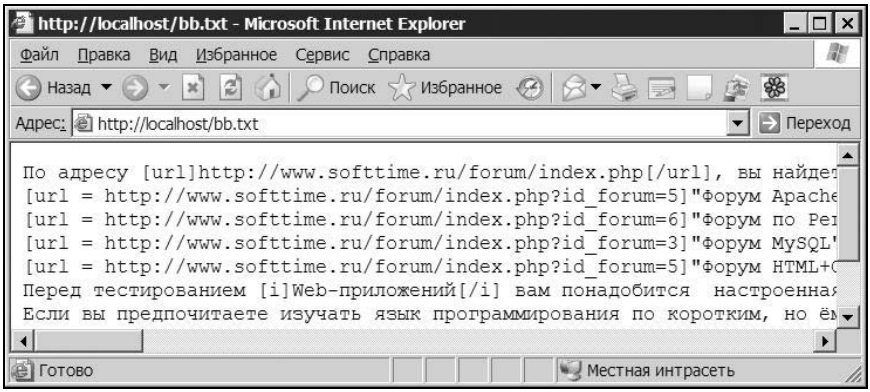


Рис. I.2.3. Содержимое файла bb.txt

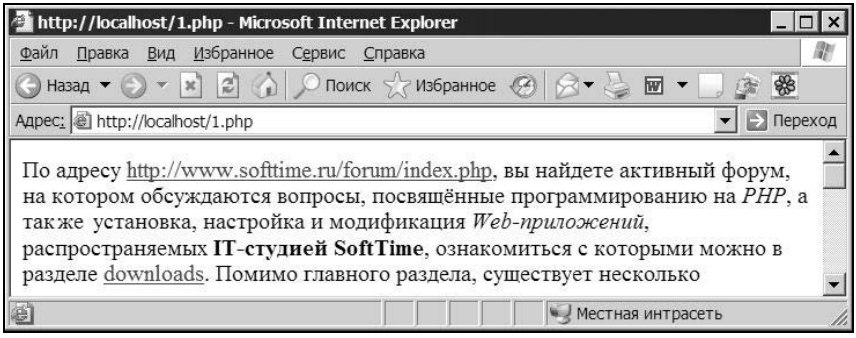
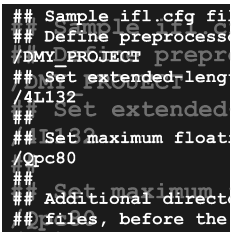


Рис. I.2.4. Преобразованное содержимое файла bb.txt

I.2.14. Подсветка PHP-кода

В PHP есть две стандартные функции для подсветки кода: `highlight_string()` и `highlight_file()`. Данные функции имеют два серьезных недостатка: поддерживается только подсветка PHP-кода и только кода, размещенного между тегами `<?php` и `?>` (а также `<?` и `?>`). Создайте собственную функцию подсветки синтаксиса, лишенную этого недостатка.



Глава I.3

Файлы

Работа с файлами является неотъемлемой частью Web-приложений — в них хранится как информация, так и код самих Web-приложений. Поэтому от эффективности использования файлов зависит и производительность Web-приложений, и их безопасность.

Замечание

Все примеры из данной главы можно найти в каталоге scripts\3 компакт-диска, поставляемого вместе с книгой.

I.3.1. Загрузка файлов на сервер

Создайте Web-приложение, позволяющее загружать на сервер произвольное количество файлов (рис. I.3.1).

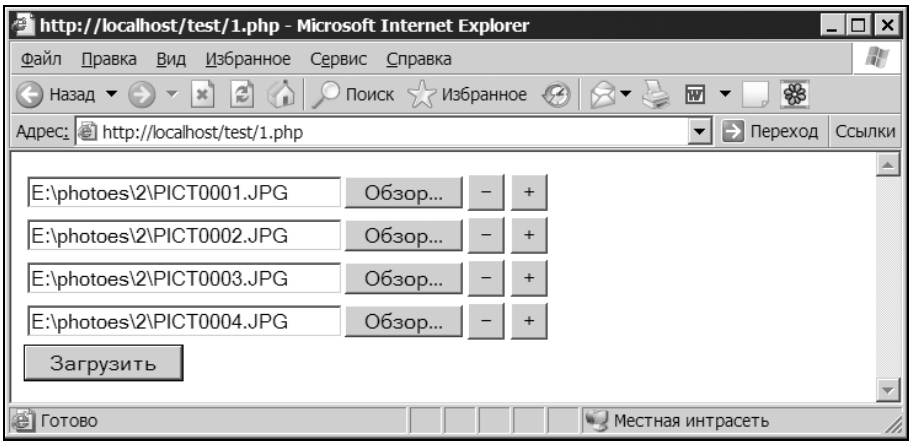


Рис. I.3.1. HTML-форма для загрузки произвольного числа файлов на сервер

I.3.2. Редактирование файлов на удаленном сервере

Создайте Web-приложение, позволяющее открывать указанный файл на сервере. Содержимое файла должно передаваться в текстовую область. После редактирования файла должна быть возможность сохранить изменения (рис. I.3.2).

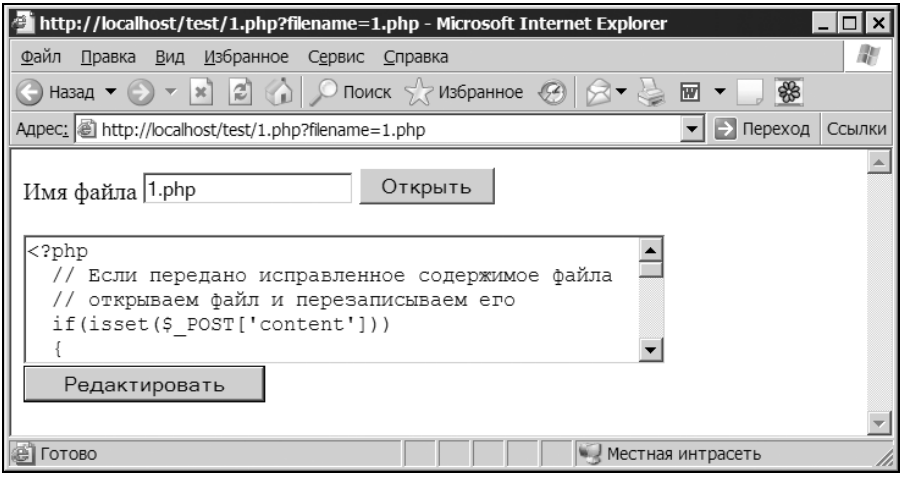


Рис. I.3.2. Редактирование файлов на удаленном сервере

I.3.3. Уязвимость скрипта загрузки

В листинге I.3.1 представлен скрипт загрузки (upload.php) — он содержит уязвимость. Используя эту уязвимость, уничтожьте файл upload.php. Разработайте скрипт загрузки файла на сервер, защищенный от этого вида уязвимости.

Замечание

Скрипт из листинга I.3.1 можно найти на компакт-диске, поставляемом вместе с книгой (scripts\3\upload.php).

Листинг I.3.1. Скрипт загрузки файла на сервер

```
<form enctype='multipart/form-data' method=post>
  <input type="file" size="32" name="filename"><br>
  <input class=button type=submit value='Загрузить'>
</form>
```

```
<?php
  // Обработчик формы
  if(!empty($_FILES['filename']['tmp_name']))
  {
     // Сохраняем файл в текущем каталоге
     if(copy($_FILES['filename']['tmp_name'],
             $_FILES['filename']['name']))
     {
       echo "Файл успешно загружен - <a href=" .
             $_FILES['filename']['name'] . ">" .
             $_FILES['filename']['name']."</a>";
     }
  }
?>
```

I.3.4. Счетчик загрузок

На сервере для свободной загрузки располагаются три файла archive1.zip, archive2.zip и archive3.zip. Необходимо создать скрипт, подсчитывающий количество загрузок файлов с сервера (рис. I.3.3).

Замечание

Файлы archive1.zip, archive2.zip и archive3.zip можно найти на компакт-диске, поставляемом вместе с книгой (scripts\3).

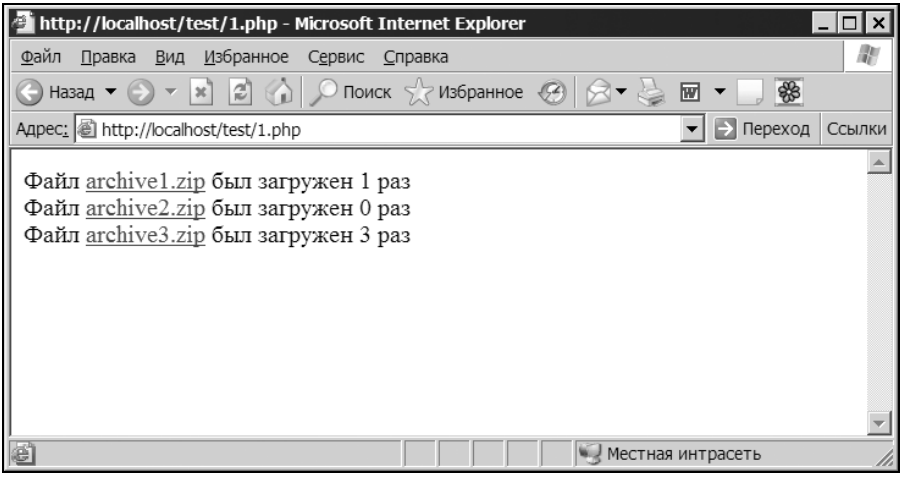


Рис. I.3.3. Счетчик загрузок файлов

I.3.5. Сохранение текстовых и графических файлов

Скрипт подсчета загрузок файлов решает еще одну проблему, связанную с безопасностью системы, — он скрывает от посетителя истинный путь к файлам. Их можно спрятать глубоко в системе, при этом посетитель будет всегда видеть только адрес страницы загрузки.

Тем не менее, если в качестве файла для загрузки указать текстовый файл, браузер не предоставит окна загрузки, а загрузит его, не только обнаружив путь к файлу, но и вынудив пользователя самостоятельно сохранять файл при помощи меню **Сохранить как**. Та же участь ожидает графические файлы и вообще любые файлы, которые браузер может отобразить. Создайте скрипт, позволяющий сохранять текстовые и графические файлы, предоставляя соответствующее окно для сохранения файлов (рис. I.3.4).

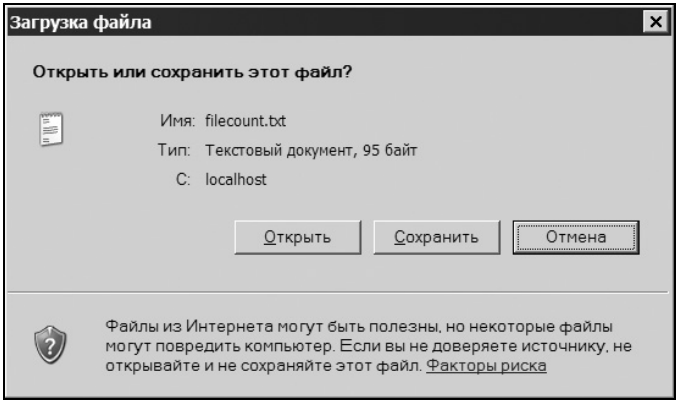


Рис. I.3.4. Диалоговое окно для загрузки файла

I.3.6. Определение размера файла

Разработайте функцию, которая принимает в качестве единственного аргумента имя файла и возвращает его размер в байтах, килобайтах или мегабайтах. Если размер файла меньше 1024 байт — функция возвращает размер в байтах, если размер меньше 1024 Кбайт — объем файла оценивается в Кбайтах, если превышен порог в 1024 Кбайт — оценка идет в Мбайтах.

I.3.7. Определение количества строк в файле

Разработайте функцию, которая принимает в качестве единственного аргумента имя файла и возвращает количество строк в нем.