



*Отчет о проведении анализа исходного кода сайта  
<http://simple.ru> на уязвимости*

## **Содержание**

1.1 Термины и обозначения.....	3
1.2 Обнаруженные уязвимости.....	3
1.3 Итоговая стоимость анализа исходного кода.....	6
1.4 Выводы и рекомендации.....	6

## 1.1 Термины и обозначения

№	Термин	Определение
1	SQL-injection	Уязвимость, возникающая как следствие недостаточной проверки принятых от пользователя значений в скрипте, с возможностью выполнения произвольных команд в БД
2	PHP-injection	Выполнение произвольных php-команд на серверной стороне
3	Remote File Include (RFI)	Использование удаленных файлов на серверной стороне
4	Local File Include (LFI)	Использование локальных файлов на серверной стороне
5	Active XSS	Вредоносный скрипт, хранящийся на сервере. Срабатывает в браузере жертвы, при открытии какой-либо страницы зараженного сайта
6	Passive XSS	Скрипт, который не хранится на сервере уязвимого сайта, либо он не может автоматически выполниться в браузере жертвы. Для срабатывания пассивной XSS, требуется некое дополнительное действие, которое должен выполнить браузер жертвы
7	File Upload	Загрузка произвольных файлов на сайт
8	Auth bypass	Обход авторизации пользователя/администратора
9	Information Leakage	Утечка конфиденциальной информации
10	Full Path Disclosure	Раскрытие полного серверного пути

## Степени критичности уязвимостей

Цвет	Уровень опасности
	низкий
	средний
	высокий

## 1.2 Обнаруженные уязвимости

### SQL-injection

#### Пример использования:

```
http://simple.ru/i.php?id=-1+UNION+SELECT+1,version()
```

#### Уязвимый код:

```
File "i.php", str 22:
```

```
$r = mysql_query("SELECT author,text from news where  
id=".$_REQUEST['id']);
```

#### Исправление уязвимости:

В файле "i.php" заменить строку 22 на

```
$r = mysql_query("SELECT author,text from news where  
id=".(int)$_REQUEST['id']);
```

### SQL-injection (blind)

#### Пример использования:

```
http://simple.ru/admin/users.php?n=')+if((substring(  
version(),1,1)=5),1,(select+1+union+select+2))+--+
```

#### Требования:

- 1) Доступ в панель администрирования
- 2) magic\_quotes\_gpc = Off

#### Уязвимый код:

```
File "admin/users.php", str 3:
```

```
$r = mysql_query("INSERT INTO authors (name) VALUES  
('".$_GET['n']."'")) or die(mysql_error());
```

#### Исправление уязвимости:

В файле " admin/users.php" заменить строку 3 на

```
$r = mysql_query("INSERT INTO authors (name) VALUES  
('".$_mysql_real_escape_string($_GET['n'])."'")) or  
die(mysql_error());
```

## Passive XSS

### Пример использования:

`http://simple.ru/logout.php?url=[XSS]`

### Уязвимый код:

```
File "logout.php", str 57:  
echo "<a href=" .$_GET['url']. ">";
```

### Исправление уязвимости:

В файле "url.php" заменить строку 57 на  
`echo "<a href=" .htmlspecialchars($_GET['url']). ">";`

## Full Path Disclosure

### Пример использования:

`http://simple.ru/index.php?i[]=1`

### Уязвимый код:

```
File "index.php", str 5:  
$pic = htmlspecialchars($_GET["pic"]);
```

### Исправление уязвимости:

В файле "index.php" заменить строку 5 на  
`$pic = htmlspecialchars((string)$_GET["pic"]);`

## Вывод ошибок не отключен

### Исправление уязвимости:

В файл ".htaccess" добавить строчку:  
`php_flag display_errors off`

### 1.3 Итоговая стоимость анализа исходного кода

№	Тип уязвимости	Коэффициент	Стоимость (в руб)
1	SQL-injection	1.45	5.800
2	Passive XSS	1	300
3	Full Path Disclosure	1	150
4	Прочие мелкие уязвимости, ошибки, недочеты	-	500
5	Устранение всех уязвимостей	-	2.600
	ИТОГО К ОПЛАТЕ:		<b>9.350</b>

### 1.4 Выводы и рекомендации

В результате проведения аудита безопасности сайта <http://simple.ru> (размер: 1,8 Мб) было обнаружено:

2 уязвимости высокой опасности  
1 уязвимость средней опасности  
3 уязвимости низкой опасности

#### Рекомендации:

- В файле robots.txt удалить путь к папке администрирования сайта /admincms/;
- Удалить файл phpinfo.php;
- Добавить basic-auth авторизацию в административную панель;

#### Наши контакты:

Site: <http://rebz.net>

E-Mail: [support@rebz.net](mailto:support@rebz.net)