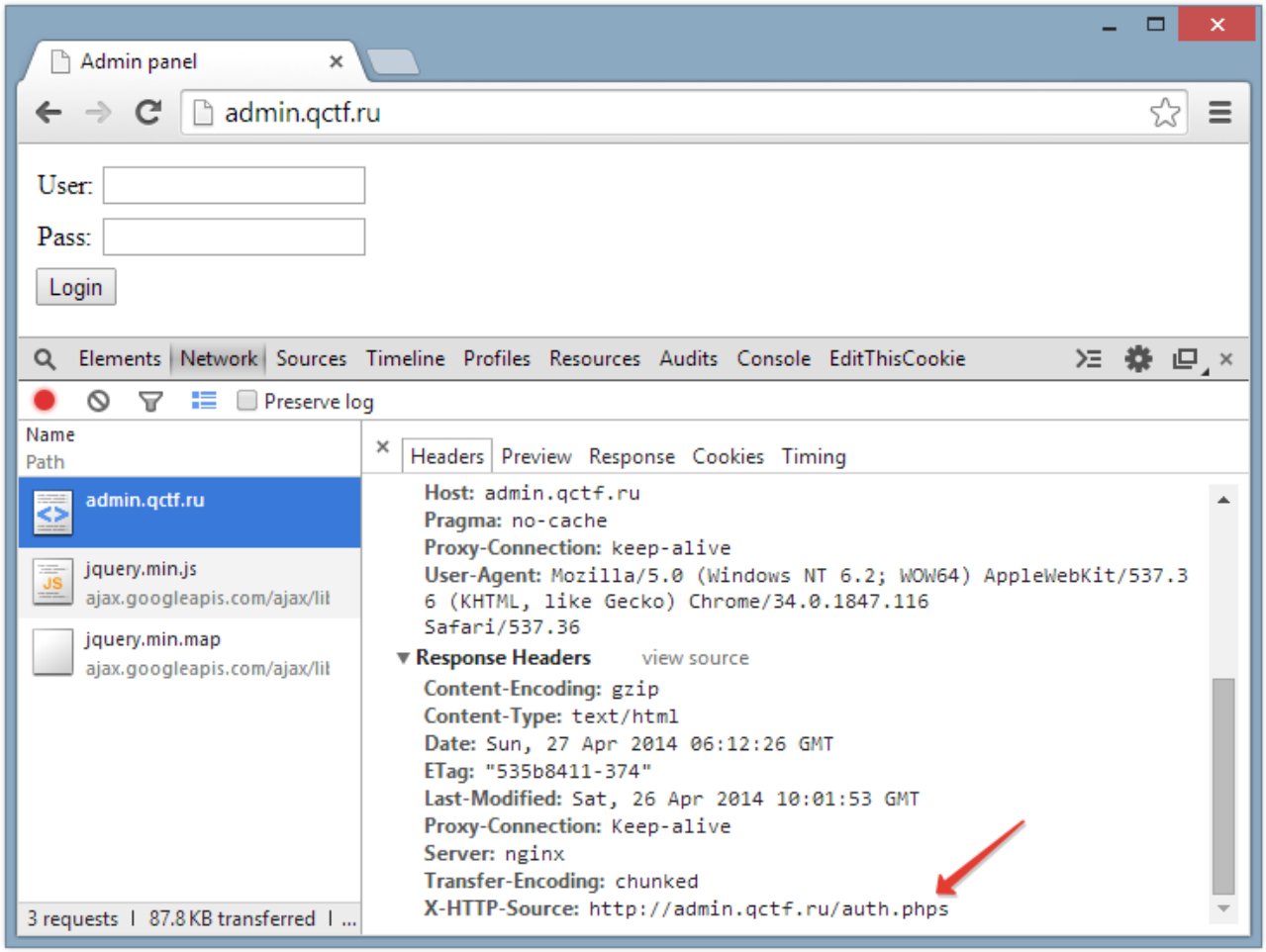
QCTF “Авторизуйся”

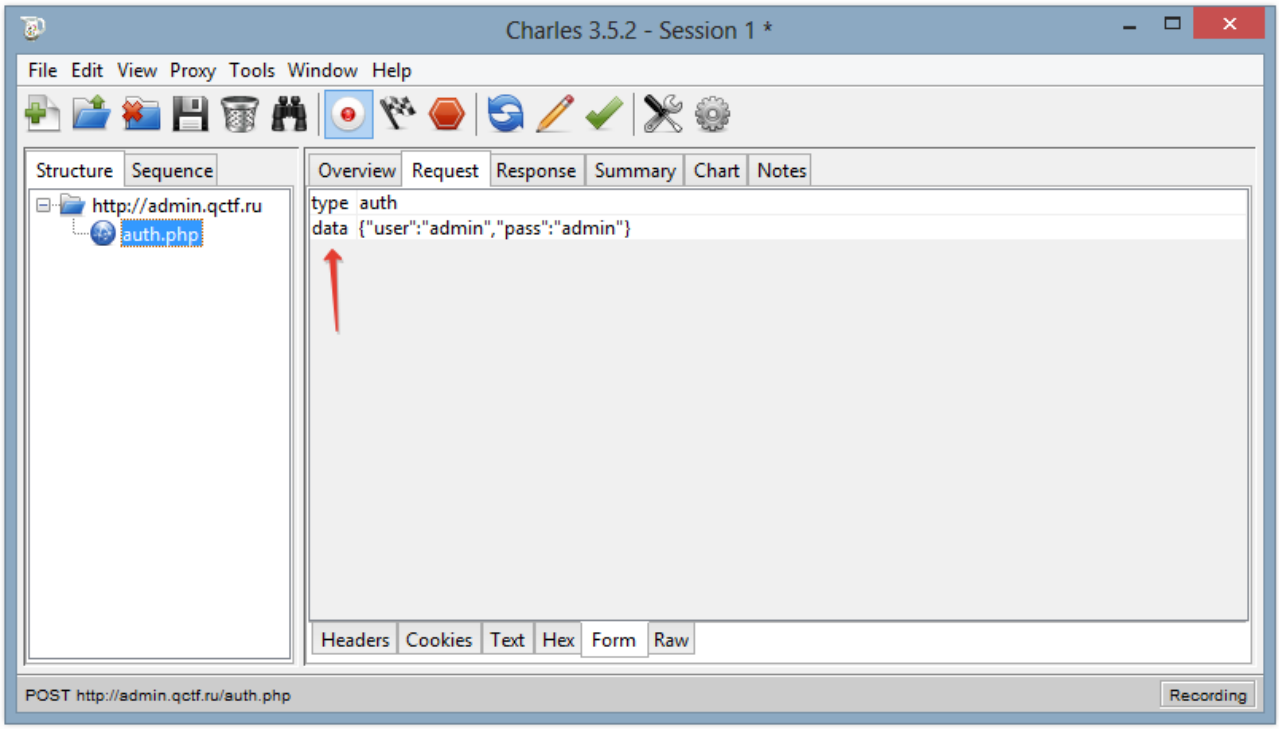
Заходим на сайт. Первым делом изучаем заголовки ответа сервера.

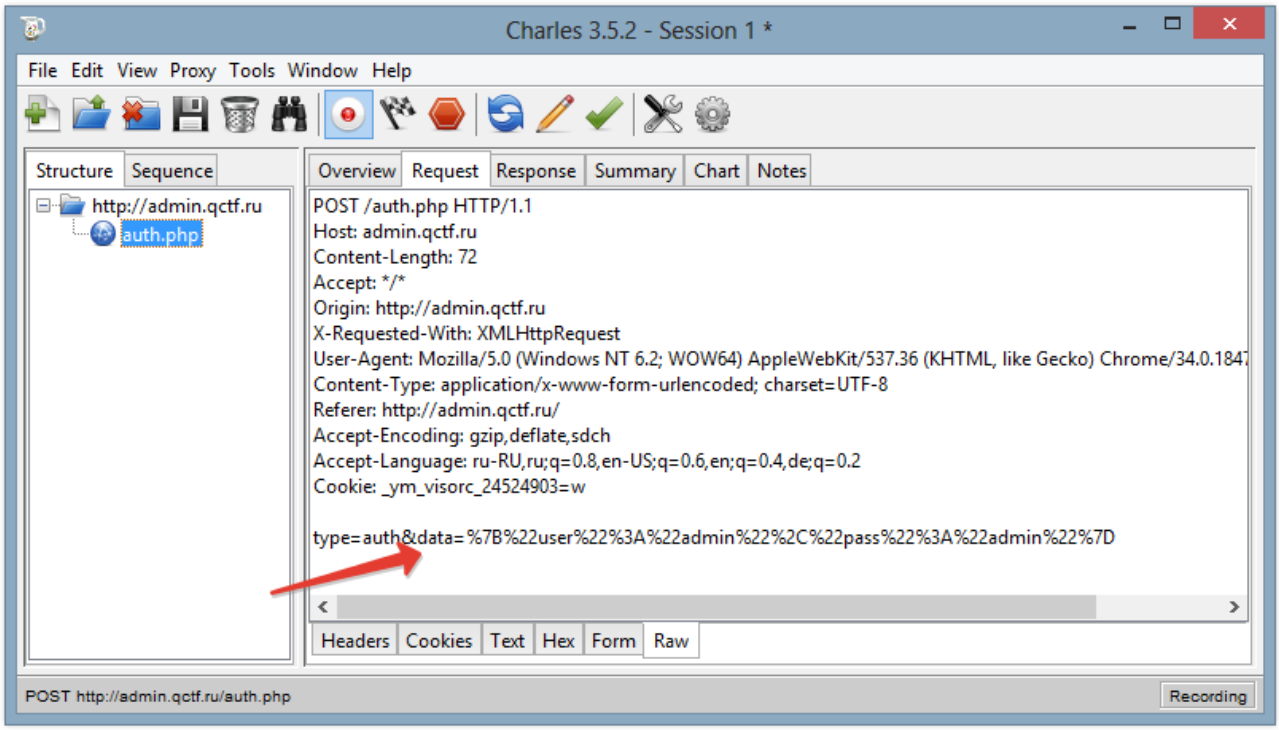


Видим странный заголовок X-HTTP-Source. Открываем - это исходник серверной части авторизации. Ок, к нему еще вернемся.
Далее откроем исходный код index.html.

```
url: '/auth.php',
type: 'POST',
data: {
        'type': 'auth',
        'data': JSON.stringify({
                "user": document.getElementById("user").value,
                "pass": document.getElementById("pass").value
        })
}
```

По нажатию на кнопку Login ajax отправляет POST запрос на сервер. Метод JSON.stringify сериализует объект в строку. Попробуем для теста отправить admin:admin, и посниффаем удобным HTTP сниффером Charles.





Ок, такой способ отправки data кажется уязвимым местом, нужно погуглить, как работает stringify, может что-то можно с этим сделать.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify

Description

`JSON.stringify` converts an object to JSON notation representing it:

- Properties of non-array objects are not guaranteed to be stringified in any particular order. Do not rely on ordering of properties within the same object within the stringification.
- Boolean, Number, and String objects are converted to the corresponding primitive values during stringification, in accord with the traditional conversion semantics.
- If `undefined`, a function, or an XML value is encountered during conversion it is either omitted (when it is found in an object) or censored to null (when it is found in an array).

```
1 JSON.stringify({});                  // '{}'
2 JSON.stringify(true);                // 'true'
3 JSON.stringify("foo");               // '"foo"'
4 JSON.stringify([1, "false", false]); // '[1,"false",false]'
5 JSON.stringify({ x: 5 });            // '{"x":5}'
6 JSON.stringify({x: 5, y: 6});        // '{"x":5,"y":6}' or '{"y":6,"x":5}'
```

Видно, что "false" и false это разные вещи, т.е stringify учитывает тип обьекта.
Взглянем на auth.phps
$data = json_decode($_POST['data'], true);
if ($data['user'] == 'admin' && $data['pass'] == PASS) {...
Гуглим json_decode

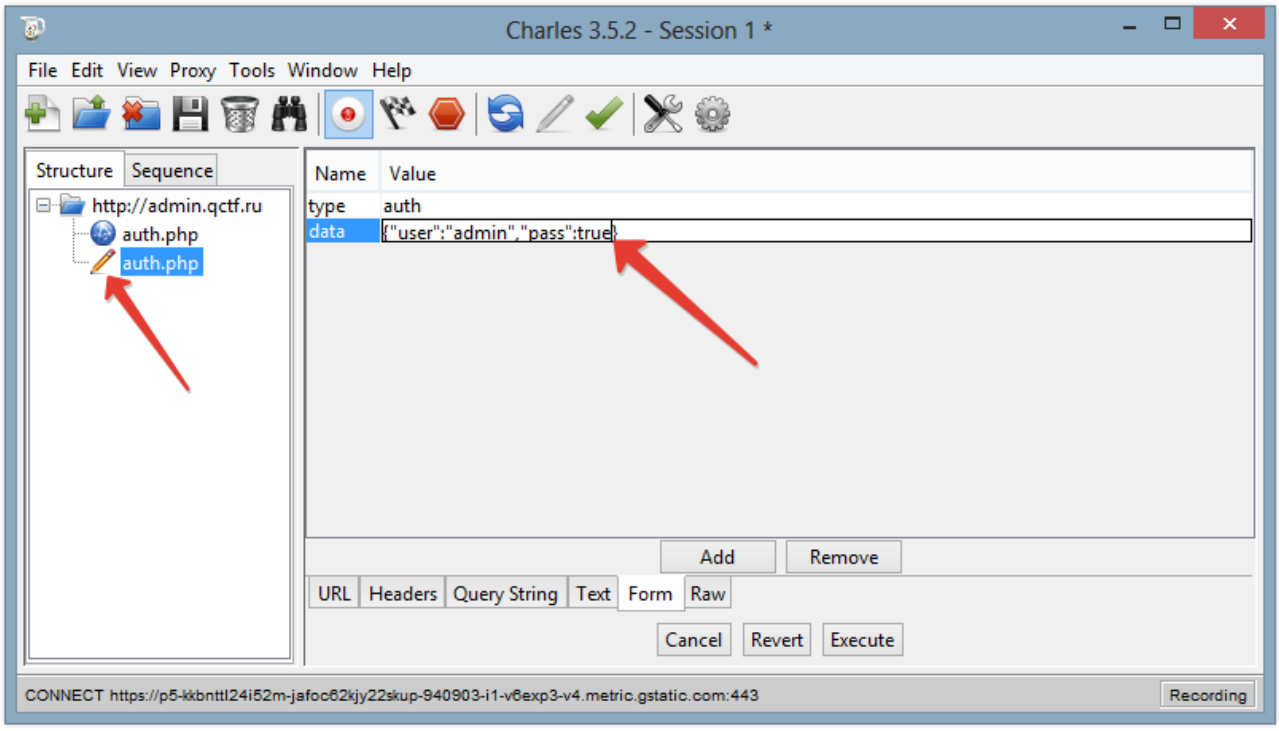
www.php.net/manual/ru/function.json-decode.php

php Downloads Documentation Get Involved Help Search

Возвращаемые значения

Возвращает данные `json` преобразованные в соответствующие типы PHP. Значения *true*, *false* и *null* (регистронезависимые) возвращаются как TRUE, FALSE и NULL соответственно. NULL также возвращается, если `json` не может быть преобразован или закодированные данные содержат вложенных уровней больше, чем допустимый предел для рекурсий.

Вспоминаем, что в php (TRUE == 'любая строка') - это TRUE
Правим запрос в Charles (правой кнопкой по запросу -> edit)



Жмем Execute. Получаем ответ сервера