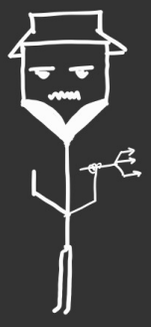
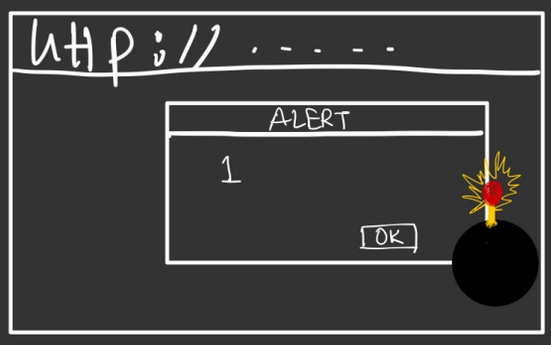
XSS



<script> alert(1) </script>



I am 'xss' attacked

Just alert, aahh!!

It's just alert, chill

Are you kidding me?
I know it is more than alert(1)
SOMEONE HELP !!!

Let me tell you what that means
I will tell you to fix Also

thx !

thank me later

Rohit

sec_r0

What is this -3

Hey !!

I am Rohit

And Today I want to tell u about

XSS

* what is xss
* How it can be exploited
* consequences
* types
* Remediation
* Best Practices.

LET's GET STARTED

sec_r0

~~what is xss ?~~

This is ~~google~~ duck-duck-go - able Question
Let me answer it in a diff way

why xss happens !

Cross site scripting

1

You Pass Some input

Can be
- → GET param
- → POST param
- → Header
- → Or Any client Side data

```
http://
<html>
  <meta> </meta>
  <script> </script>
<body>
</body>
</html>
```

2

Same input goes to

SERVER

May Store in

3

DB

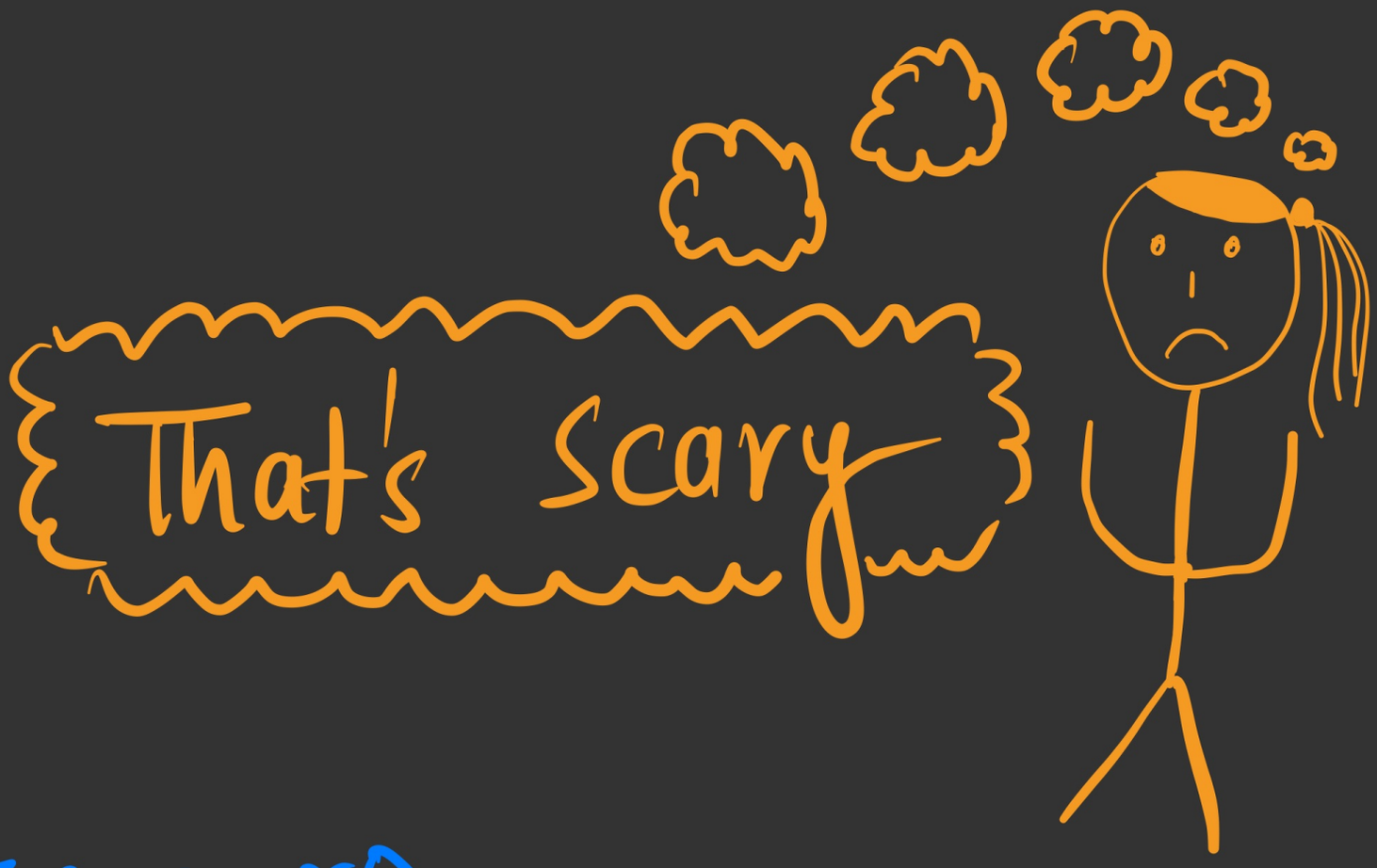
SAME data Comes back to UI

4

sec_r0

* Issue happens because server trusts the data from browser.

* If in place of data, HTML is passed, browser will trust (4) and will ~~execute~~ render it too.



sec_r0

Browser will just ~~execute~~ render HTML, and HTML is just markup

But, if I pass JavaScript code in Input ????

The js code, will go to server & when it will come from server to browser it will get ~~rendered~~ executed too .....

Yes, Exactly

sec_r0

So,
XSS Happens, when data from client side goes to server & later comes back to Client in same or other session .... unsanitized!

And if I sanitize every input ???

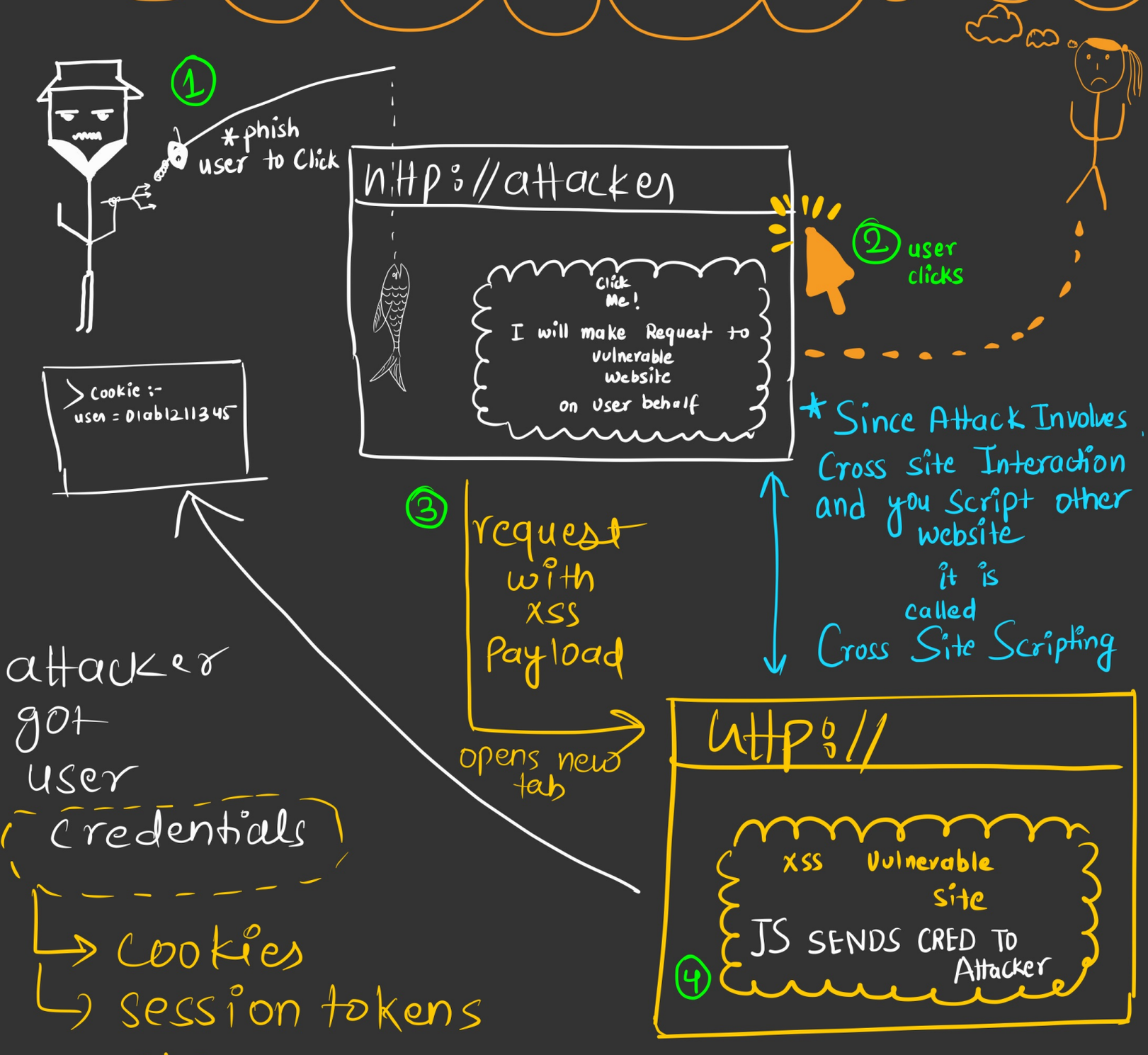
depends !!

But, I am just changing my own input!

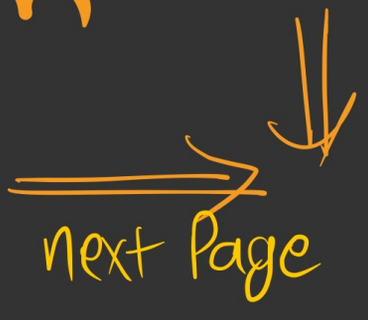
This is what you think.

sec_r0

How Is XSS Exploited ?



What all can achieve ?



Can

→ Steal session tokens.

→ Phishing to wide users.

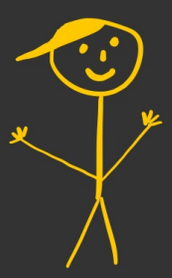
→ Keylogger injection.

→ Internal port Scanning (inside organization's firewall)

→ fetch CSRF Tokens

Consequences of Xss

what is this ??



Dont hurry wait for another zine !!

sec_r0

Types Of Xss

Reflected

* Example we saw
* when Attack payload comes in same request

Stored

* Attack payload is stored in DB
* User is exploited when he/she access resource with payload.
* Covers Variety of users ☹

Self

not at all critical

* XSS is present in website but cross site exploitation is not possible!

DOM

* Payload not at all goes to server
* Payload goes from one DOM entity to other on Client Side

* Request Goes to Server and Server is attack aware.
* WAF can prevent.

Difficult to Capture as server may not be aware of attack.

sec_r0

Remediation

It's Easy Trust Me Again!!

How Can I Fix this.....

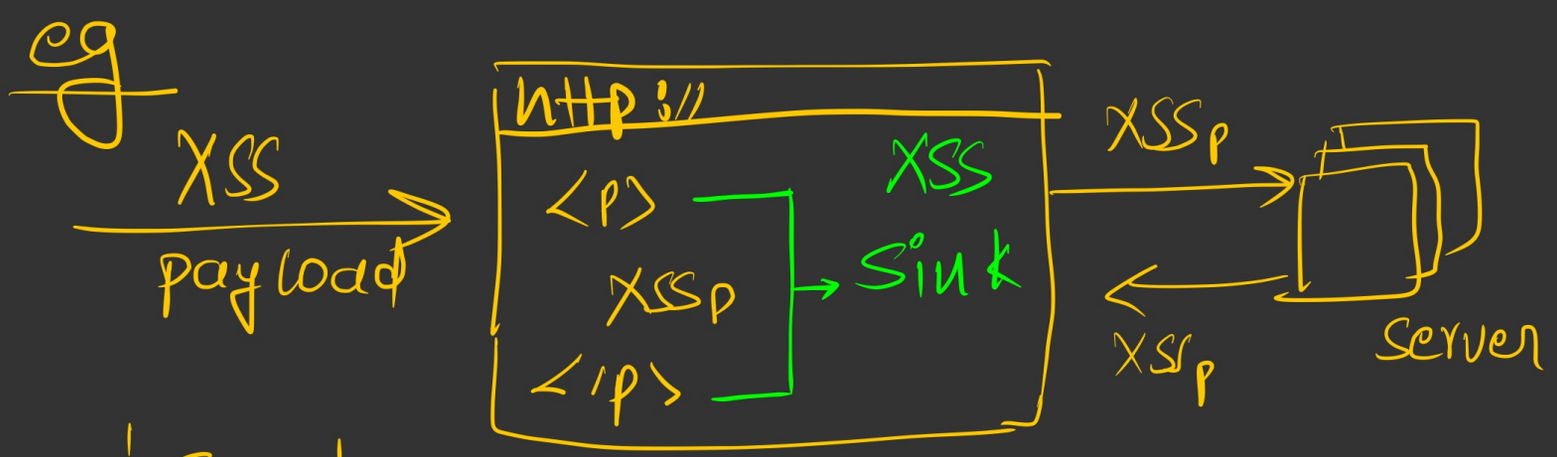
---

* Find 'SINK'
  - ↳ Location where Payload finally loads into.
* Do remediate According to Sink type.

Xss Sink

* Xss Remediation Guidelines depends on Sink

★ It's the location inside HTML page where xss payload finally loads in.



'Sink' can be

HTML Context

* Xss_p comes under any HTML entity

```
<p> Xssp </p>
<a href="Xssp"></a>
etc
```

JS context

When Xss_p falls Under some JavaScript Context.

```
eg <script>
     foo("XSSp")
   </script>
```

CSS Context

* when XSS_p falls under Some CSS attributes or Class

```
eg .body {
background: url("XSSp")
   }
```

sec_r0

Best Practices

Sink = HTML

HTML Escaping
+
HTML Encoding

eg '<' → '&lt'

eg '<' → '%3C'

Sink = JS

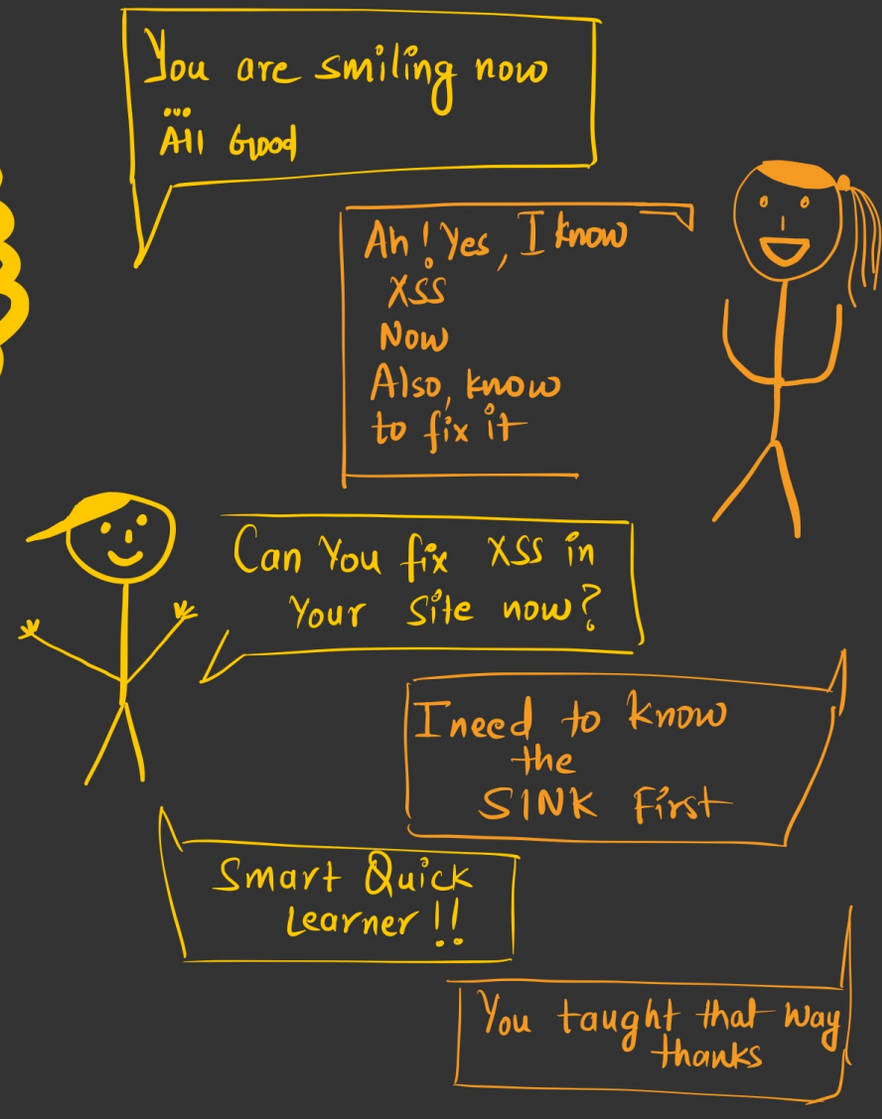
Javascript Escaping

Sink = CSS

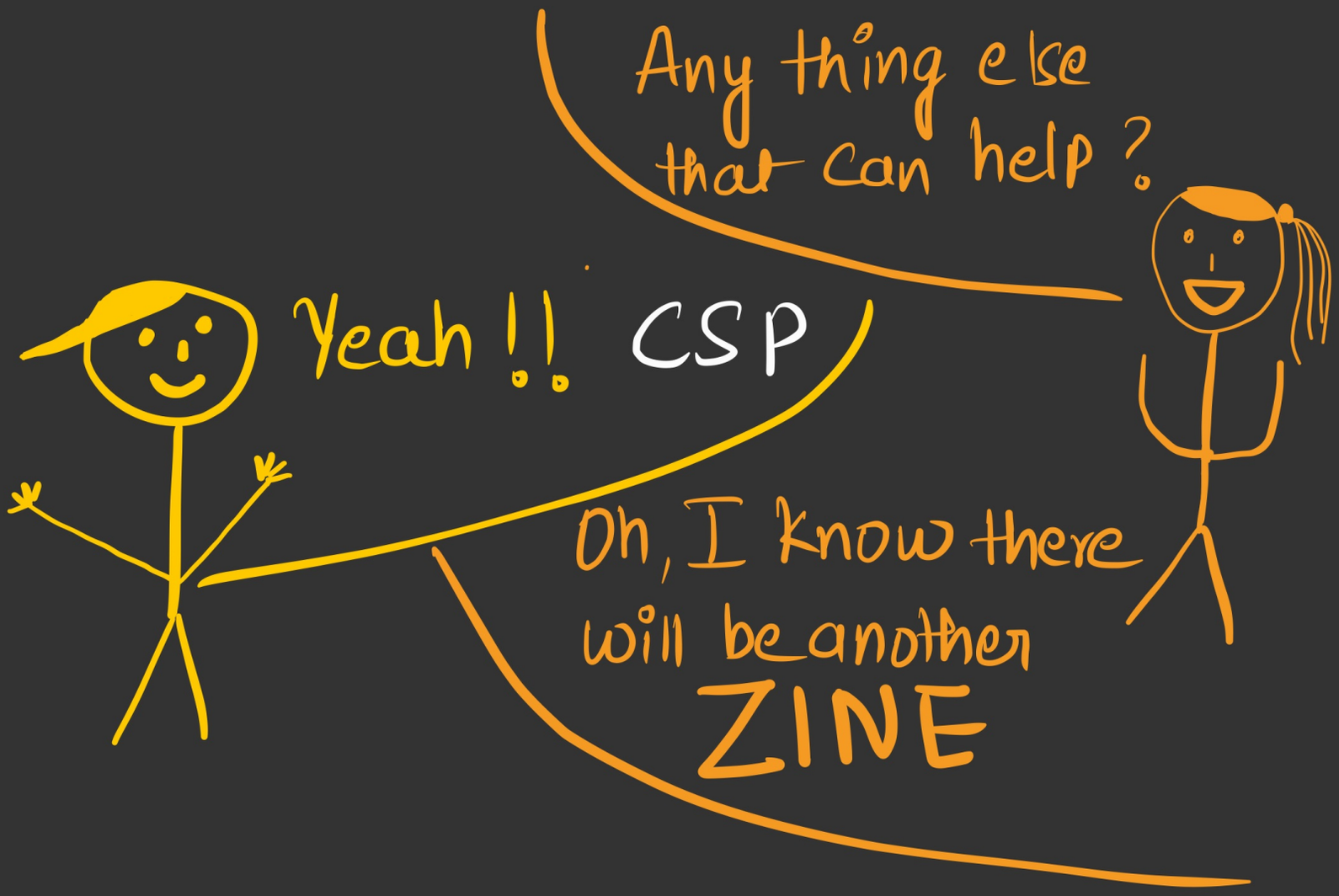
Make Sure no external urls are addressed

eg

```
.body {
background: url("xss")
}
```



sec_r0



Read more Zines @ securityzines.com

Thanks For Reading

sec_r0

↑ Join for updates