

# Web Application Security

## Winning When The Odds Are Against You

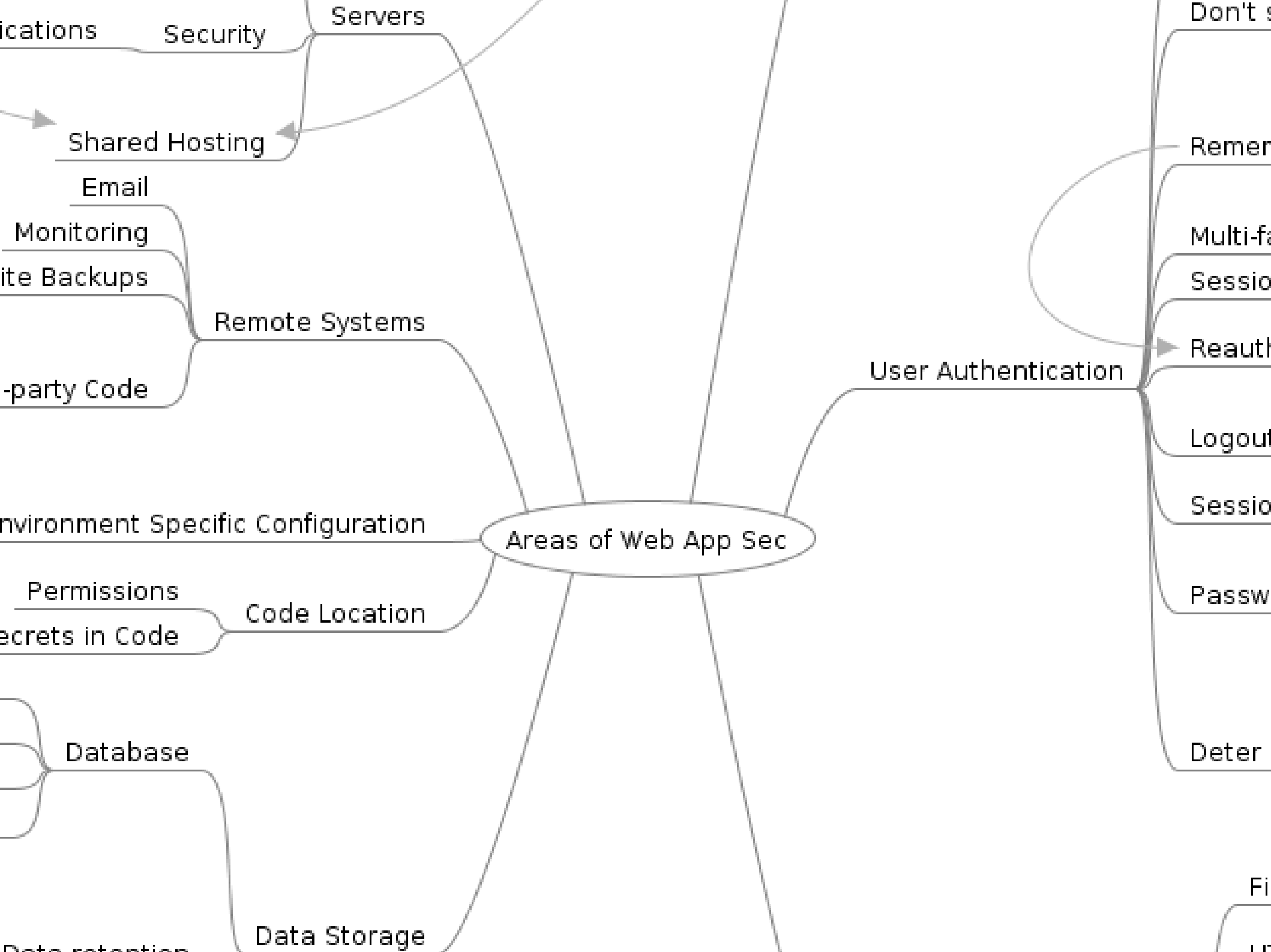


#webappsec  
#phpconfasia  
<https://joind.in/15379>

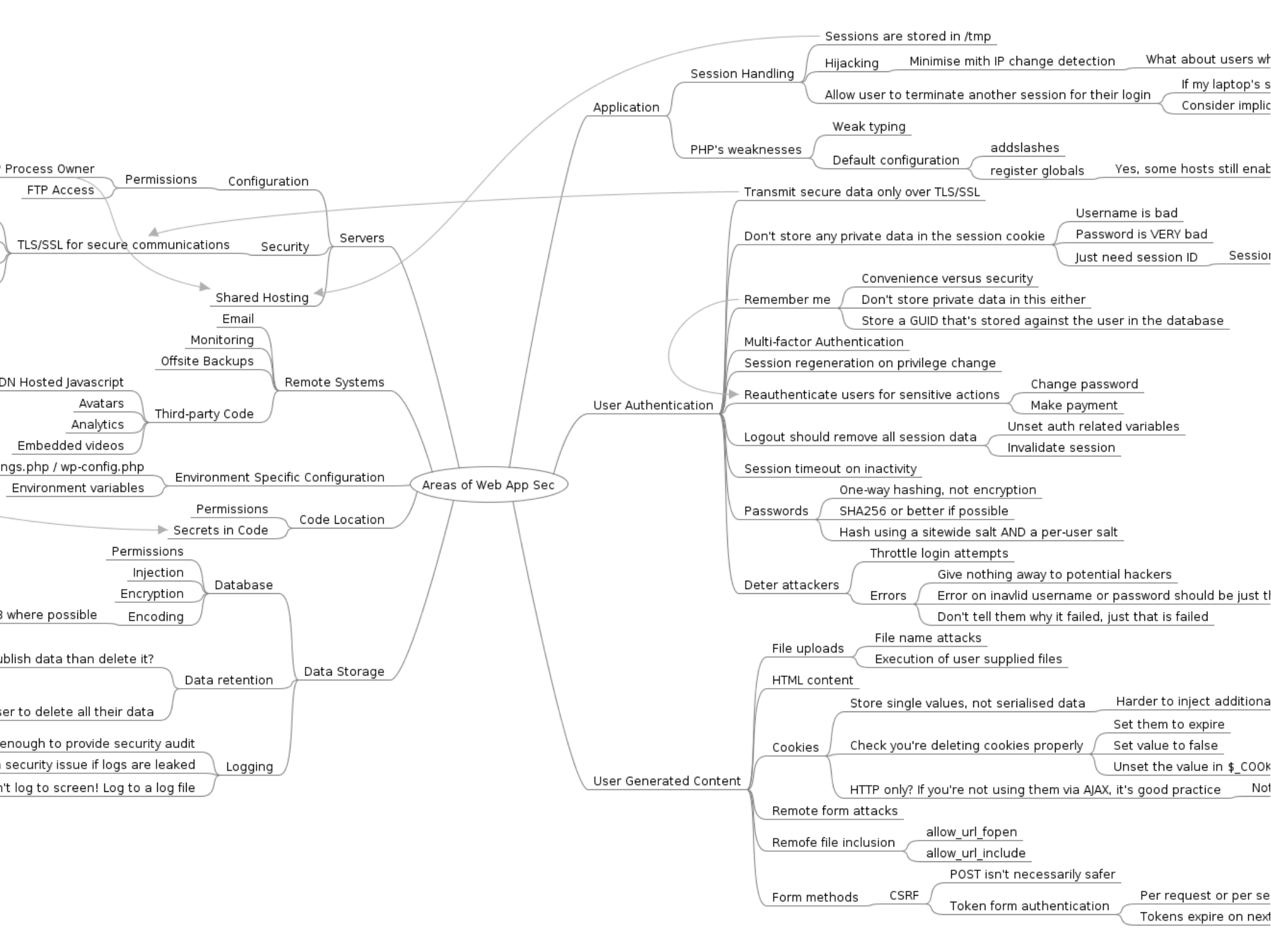
Ben Dechrai  
@bendechrai

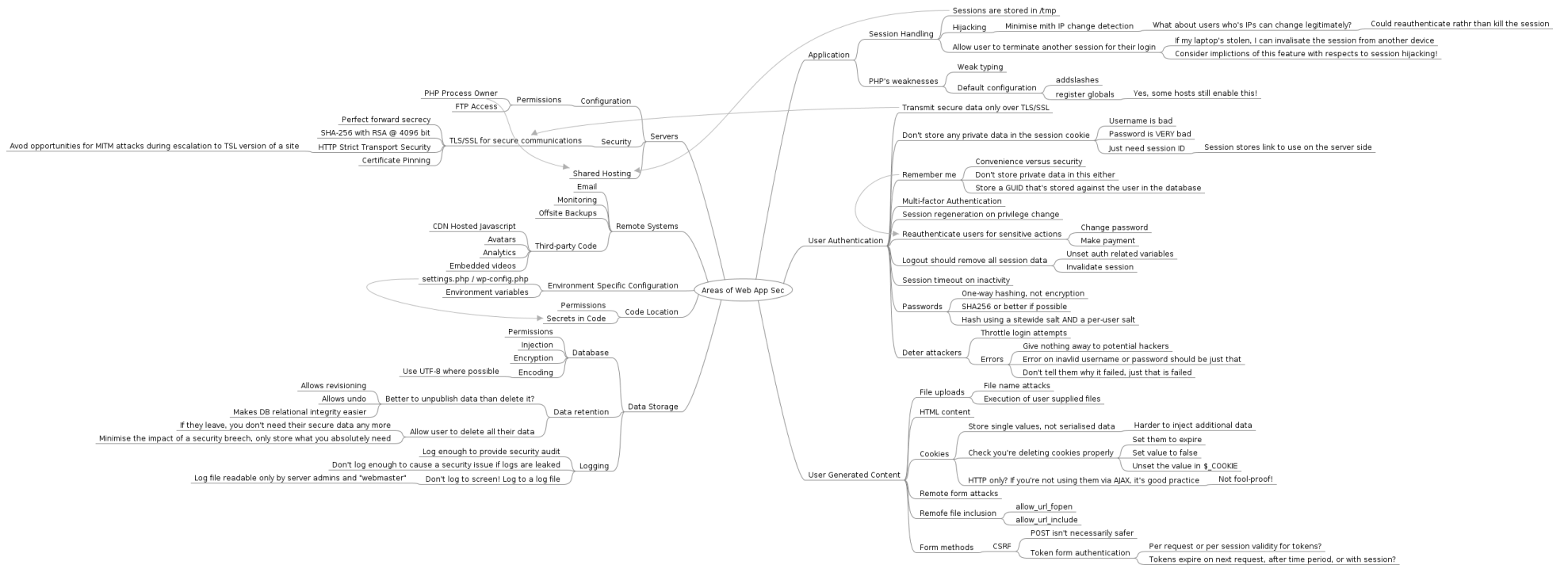


<https://cto.to/webappsec>









Where to Start?

owasp.org



# OWASP

Open Web Application  
Security Project



OWASP  
Open Web Application  
Security Project

# Top Ten Cheat Sheet

Injection

Cross Site Scripting

Weak authentication  
& session management

Insecure Direct  
Object Reference

Cross Site  
Request Forgery

Security  
Misconfiguration

Insufficient  
Cryptographic Storage

Failure to Restrict  
URL access

Insufficient Transport  
Layer Protection

Unvalidated Redirects  
and Forwards



OWASP  
Open Web Application  
Security Project

# Top Ten Cheat Sheet

Injection

Cross Site Scripting

Weak authentication  
& session management

Insecure Direct  
Object Reference

Cross Site  
Request Forgery

Security  
Misconfiguration

Insufficient  
Cryptographic Storage

Failure to Restrict  
URL access

Insufficient Transport  
Layer Protection

Unvalidated Redirects  
and Forwards



# Demo





Solutions?

Think like your favourite  
programming language...

Not in your favourite  
programming language...

Think **LIKE** your favourite  
programming language...

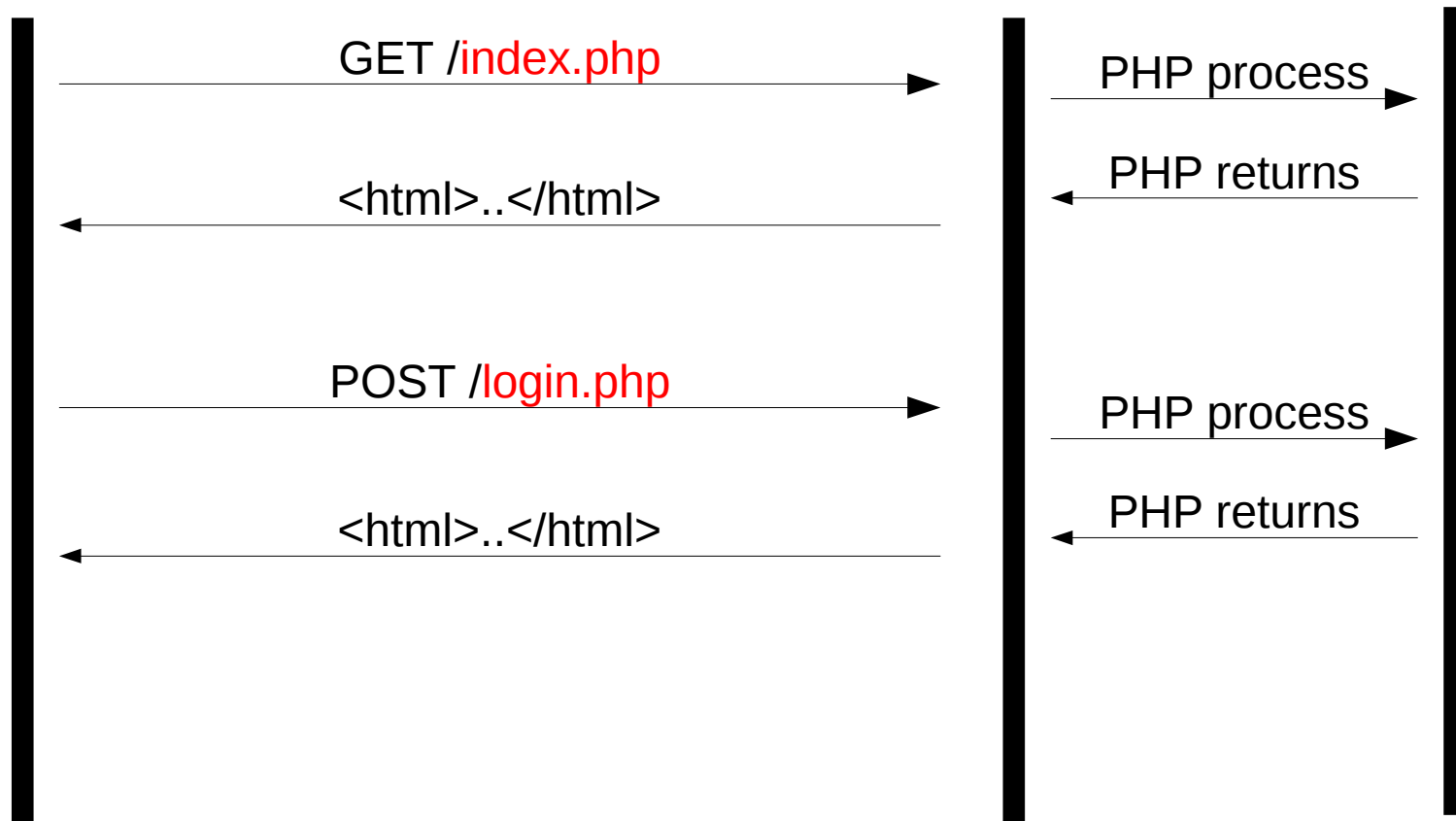


Think LIKE 

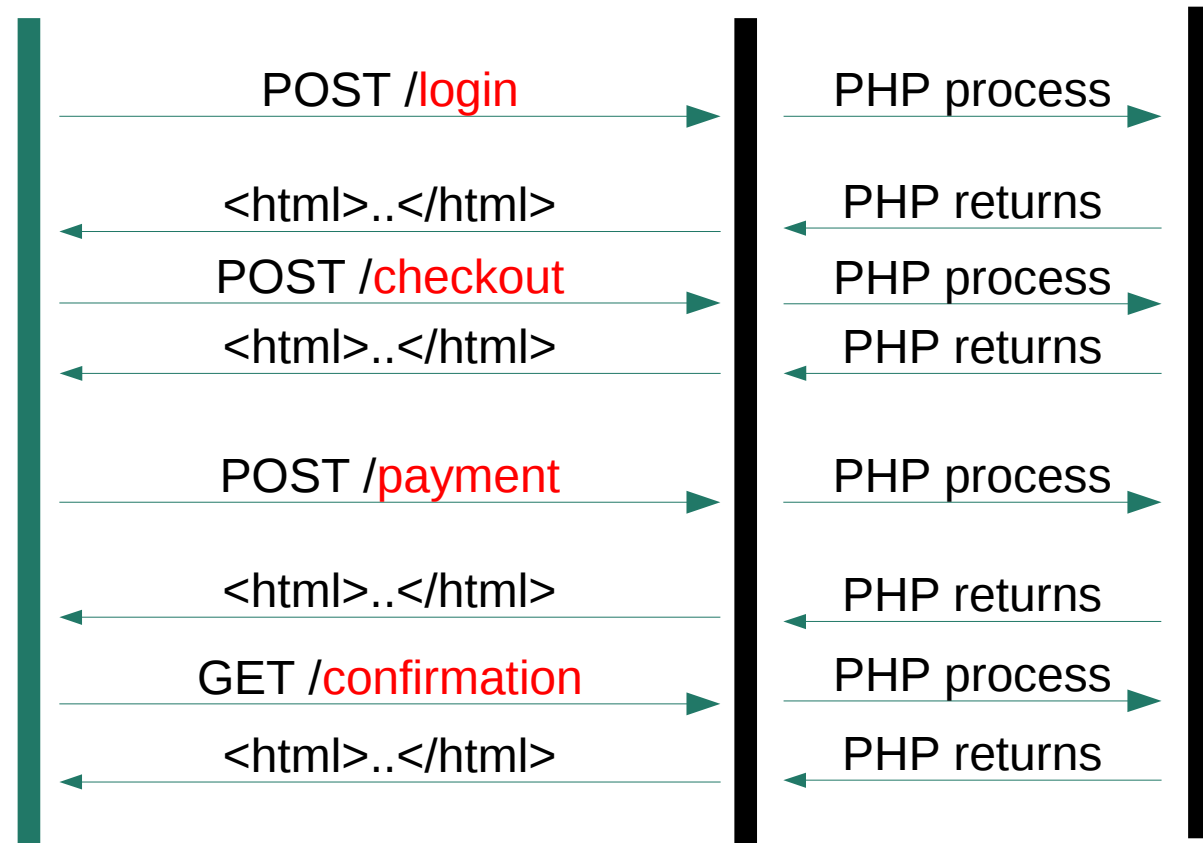
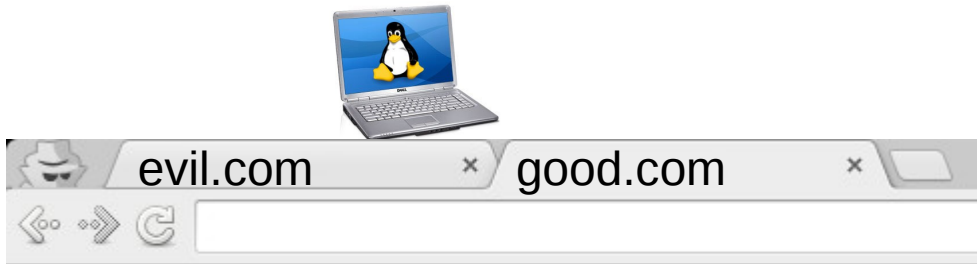
# HTTP



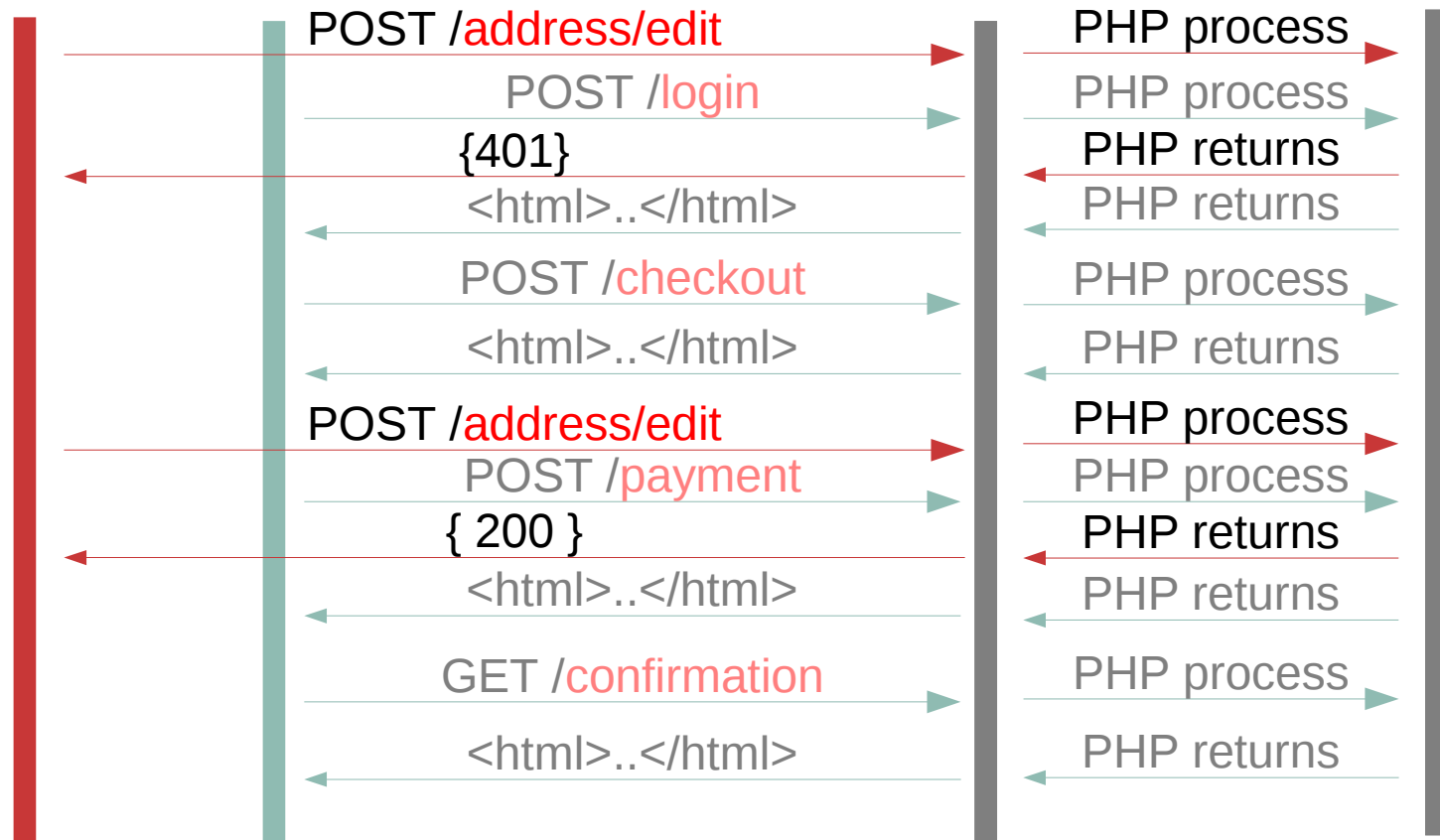
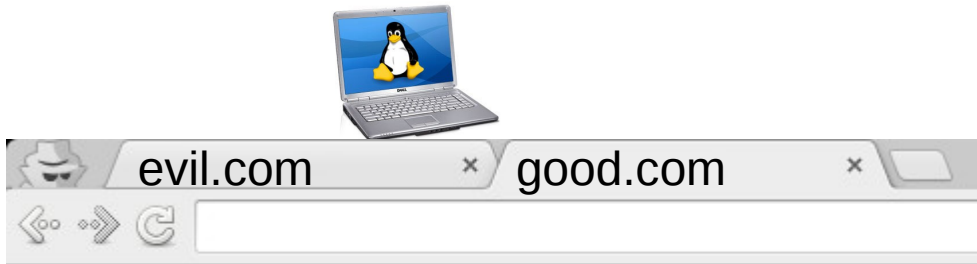
# HTTP



# Cross-Site Request Forgeries



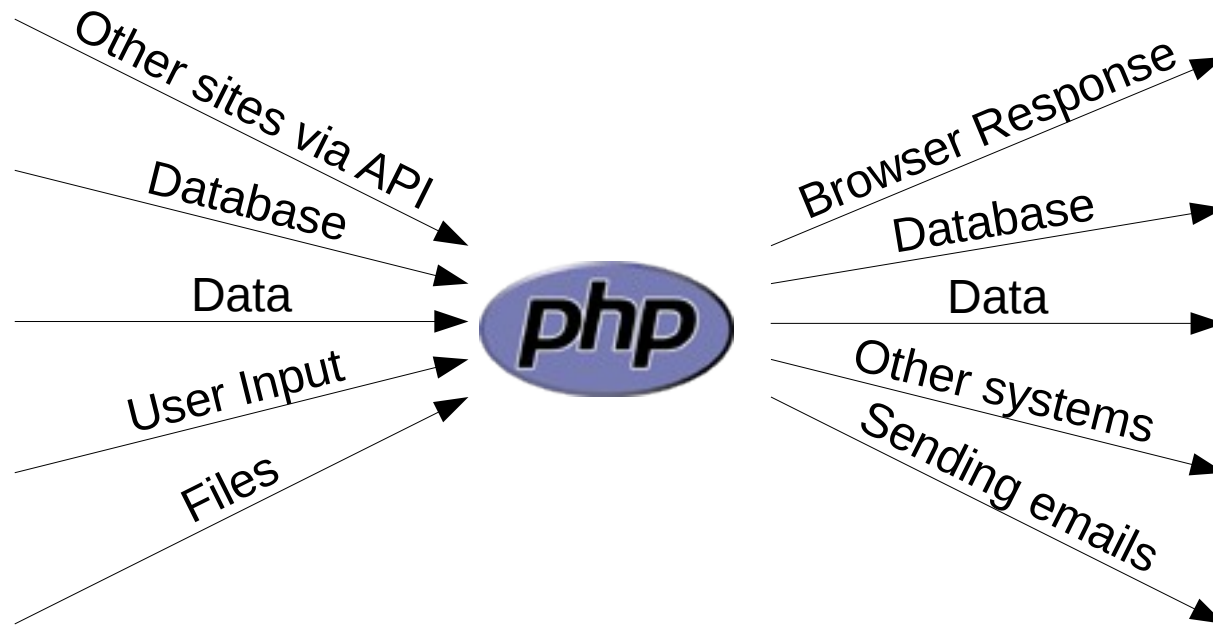
# Cross-Site Request Forgeries





# PHP Ain't Clever

(hint, not many programming languages are!)



# Piecing Data Together








`$_REQUEST` has  
no consistency!



Don't use them,  
specify the source



Even then, don't  
trust `$_POST`, etc



Consider all data  
harmful



# Validate All Incoming Data

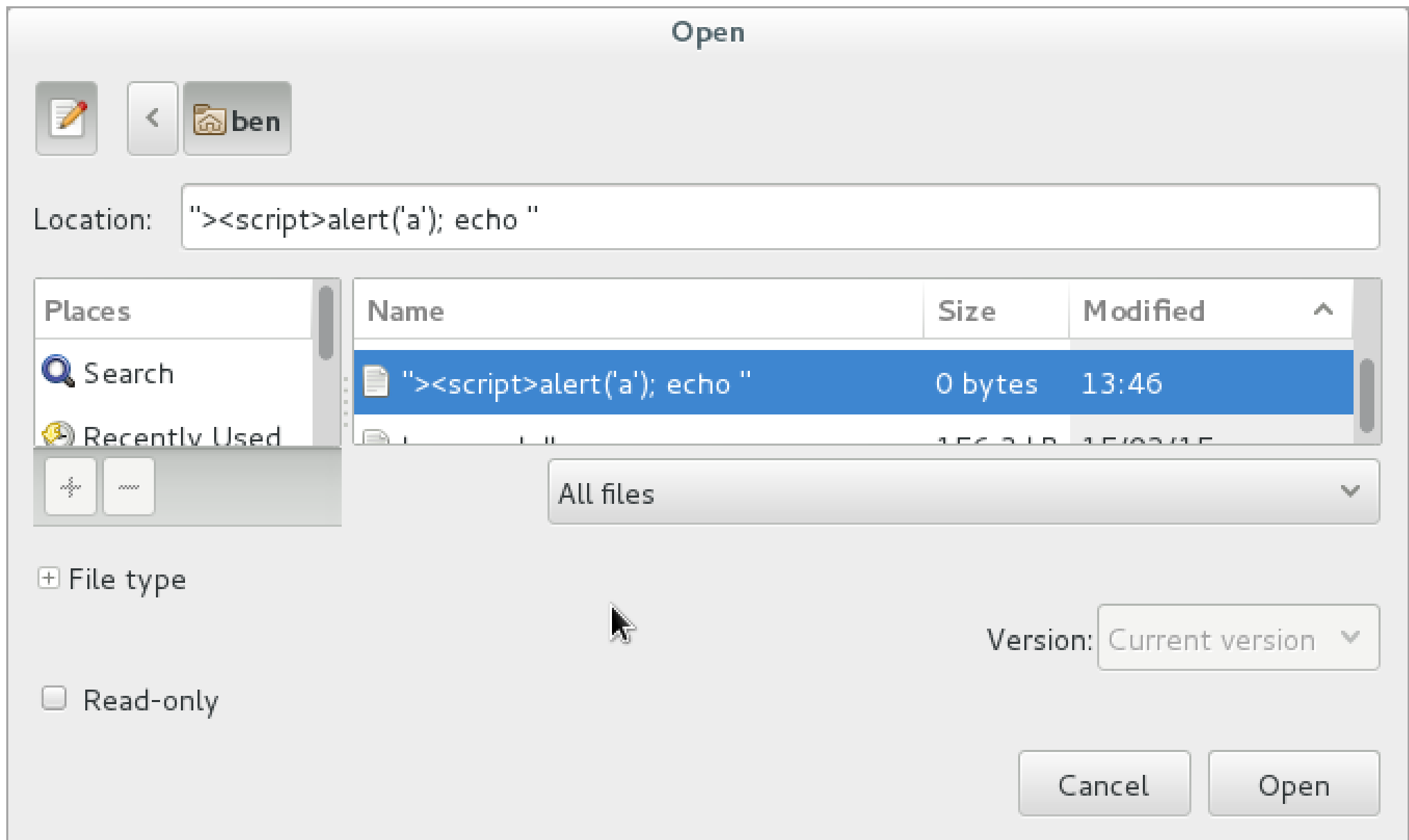


# Look Out for Odd Entry Points





# Look Out for Odd Entry Points



# Email Addresses?



ben@dechrai.com



ben+phpasia15@dechrai.com



"Ben at PHPAsia.conf"@dechrai.com



Bễn@169.254.141.29

# Email Addresses?

`/([a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\.([a-z0-9!#$%&'*/+=?^_`{|}~-]+)*)|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])*)"@([a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?|\\([0-9]|[01]?[0-9][0-9]?)\.){3}(?:[0-9]|2[0-4][0-9]|[01]?[0-9][0-9]?)|[a-z0-9-]*[a-z0-9]):([a-z0-9](?:[a-z0-9-]*[a-z0-9])?|\\([0-9]|[01]?[0-9][0-9]?)\.)+)\.)/`

# Email Addresses?

/ (?x)

(?&userpart)

@

(?&domain) | (?&ipaddress)

/

# Email Addresses?

What do you want to test?

Syntactic validity ✓

Ownership ✓

So send them an email!



# Names?

(see [http://is.qd/validating\\_names](http://is.qd/validating_names))

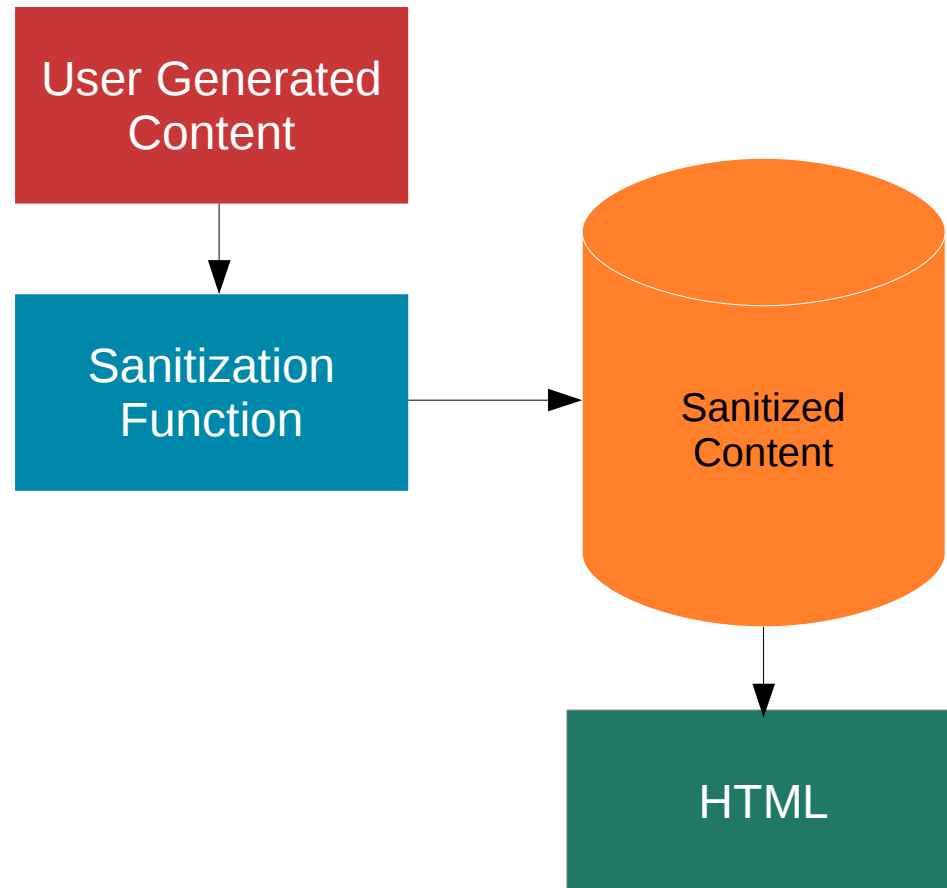
## Falsehoods Programmers Believe About Names

1. People have exactly one canonical full name.
2. People have exactly one full name which they go by.
3. People have, at this point in time, exactly one canonical full name.
4. People have, at this point in time, one full name which they go by.
5. People have exactly N names, for any value of N.
6. People's names fit within a certain defined amount of space.
7. People's names do not change.
8. People's names change, but only at a certain enumerated set of events.
9. People's names are written in ASCII.
10. People's names are written in any single character set.
11. People's names are all mapped in Unicode code points.
12. People's names are case sensitive.
13. People's names are case insensitive.
14. People's names sometimes have prefixes or suffixes, but you can safely ignore those.
15. People's names do not contain numbers.
16. People's names are not written in ALL CAPS.

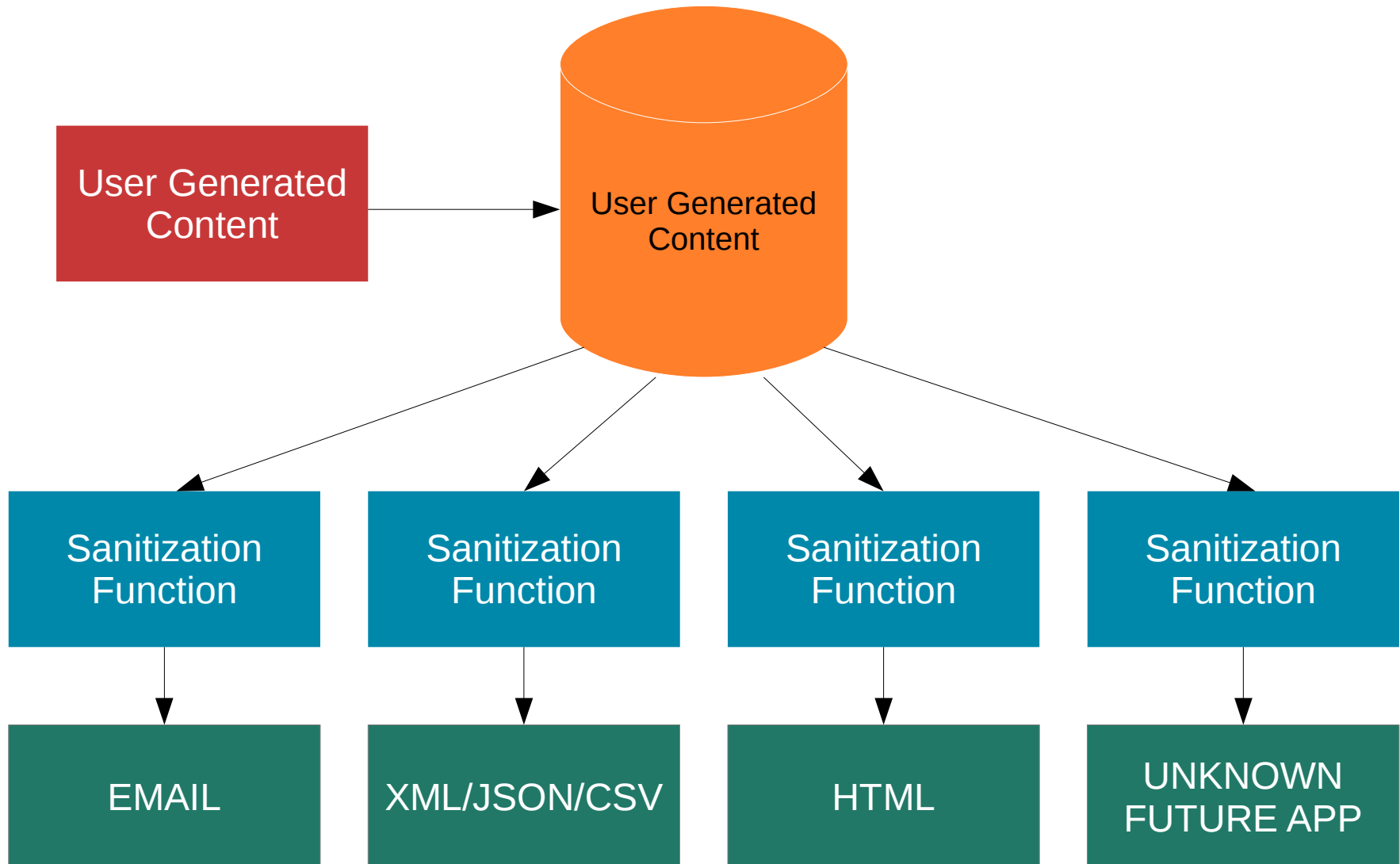
# Names?

- Who decides if a name is valid?
  - Josè Smith
  - Lazamon
  - Þórinn Eikinskjaldi
  - Πηληϊάδεω Ἀχιλῆος
  - Federico del Sagrado Corazón de Jesús García Lorca

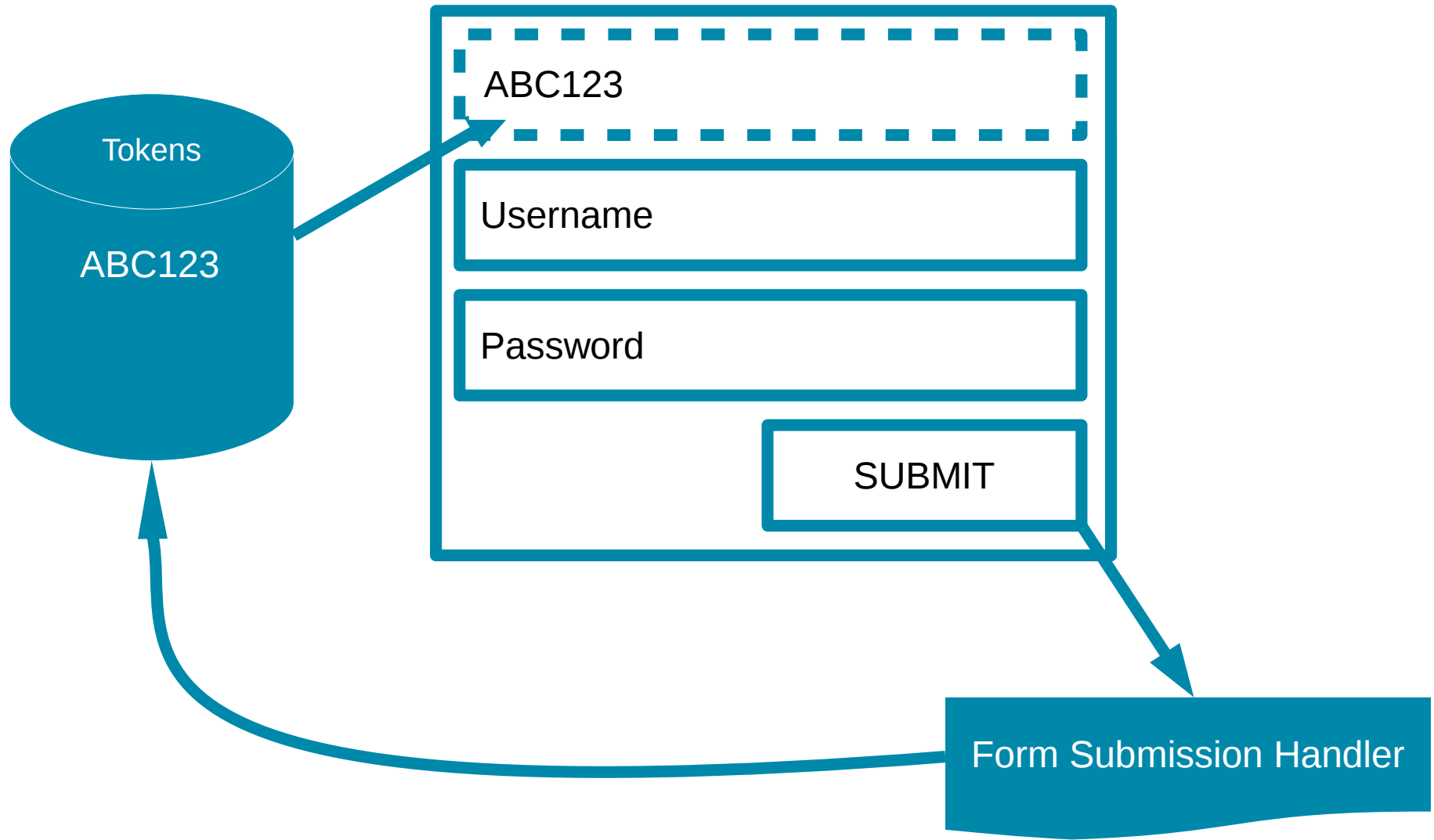
# Sanitize Data



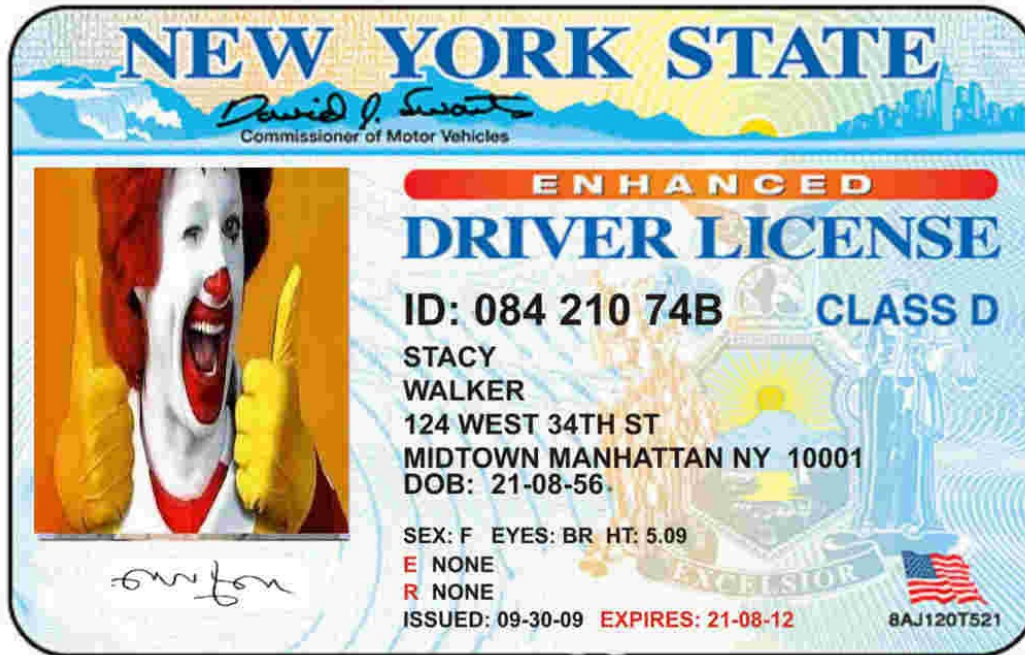
# Sanitize on Output



# Cross-Site Request Forgeries



# Cross-Site Request Forgeries



Referrers can be easily forged;  
don't rely on them

curl

```
--referer="http://good.com"  
--data "username=admin&password=complex"  
http://good.com/user/login/
```

# What Are The Odds?



# Making Prototypes Live

"That's great! Let's make it live!"

– Almost every CEO ever



# Making Prototypes Live



# References

- [OWASP Top 10 Cheat Sheet](#)

# Credits

- [Security Camera image](#) by Henning Mühlinghaus
- [Piecing Data](#) by José Manuel Ríos Valiente
- [Flip Book Animals](#) by Jamie Fingal
- [Customs and Immigration](#) From [learnz.org.nz](#)
- [Flight Tracks](#) screenshot from [flightaware.com](#)
- [Scooter Security](#) by [EpicFail.com](#)



# Thank You!

Questions?



#webappsec  
#phpconfasia  
<https://joind.in/15379>

Ben Dechrai  
@bendechrai



<https://cto.to/webappsec>