Hack2g2 - OWASP Top 10 2017

Web application security,
make web development great again.

Vannes - 21st of march 2018

Fabien Leite
https://github.com/fabienleite

HACK2G2
PARTAGEONS LA CONNAISSANCE

Summary

- What is OWASP
- What is the Top 10
- Top 10
    - A1
    - A2
    - …
    - A10
- Wild card
- Conclusion

OWASP ?

- « Open Web Application Project »
- Non profit organisation
- Theorically impartial
- For ~15 years

- Aims to spread web application security
  - Toolset
  - Guides
  - …

The Top 10

- Since 2003
- Last release was 2013

- Well known « ultimate » reference …
- Generalistic and limited …

- … But still not always understood & followed

I DON'T USUALLY CODE FOR THE WEB
BUT WHEN I DO I MAKE SURE TO DO NOTHING AGAINST XSS
imgflip.com

Top 10 2017 – A1

```
usr_name=admin&usr_password[$ne]=h4xor
```

Injections :

- NoSQL as above
- SQL

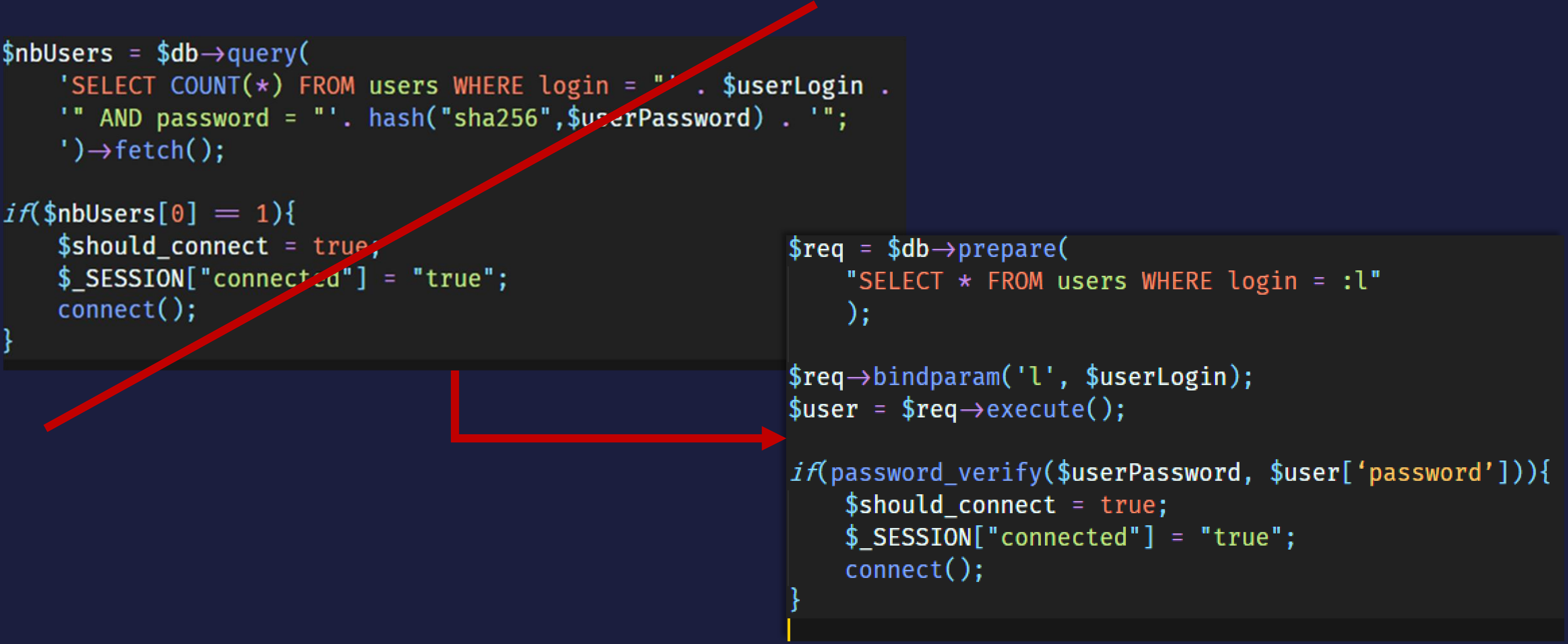
```
admin' OR '1'='1
```

- LDAP

```
admin)(&)
```

- …

Top 10 2017 – A1 : Injections



```
$nbUsers = $db->query(
    'SELECT COUNT(*) FROM users WHERE login = "' . $userLogin .
    '" AND password = "'. hash("sha256",$userPassword) . '";
    ')->fetch();

if($nbUsers[0] == 1){
    $should_connect = true;
    $_SESSION["connected"] = "true";
    connect();
}
```

```
$req = $db->prepare(
    "SELECT * FROM users WHERE login = :l"
    );

$req->bindparam('l', $userLogin);
$user = $req->execute();

if(password_verify($userPassword, $user[‘password’])){
    $should_connect = true;
    $_SESSION["connected"] = "true";
    connect();
}
```

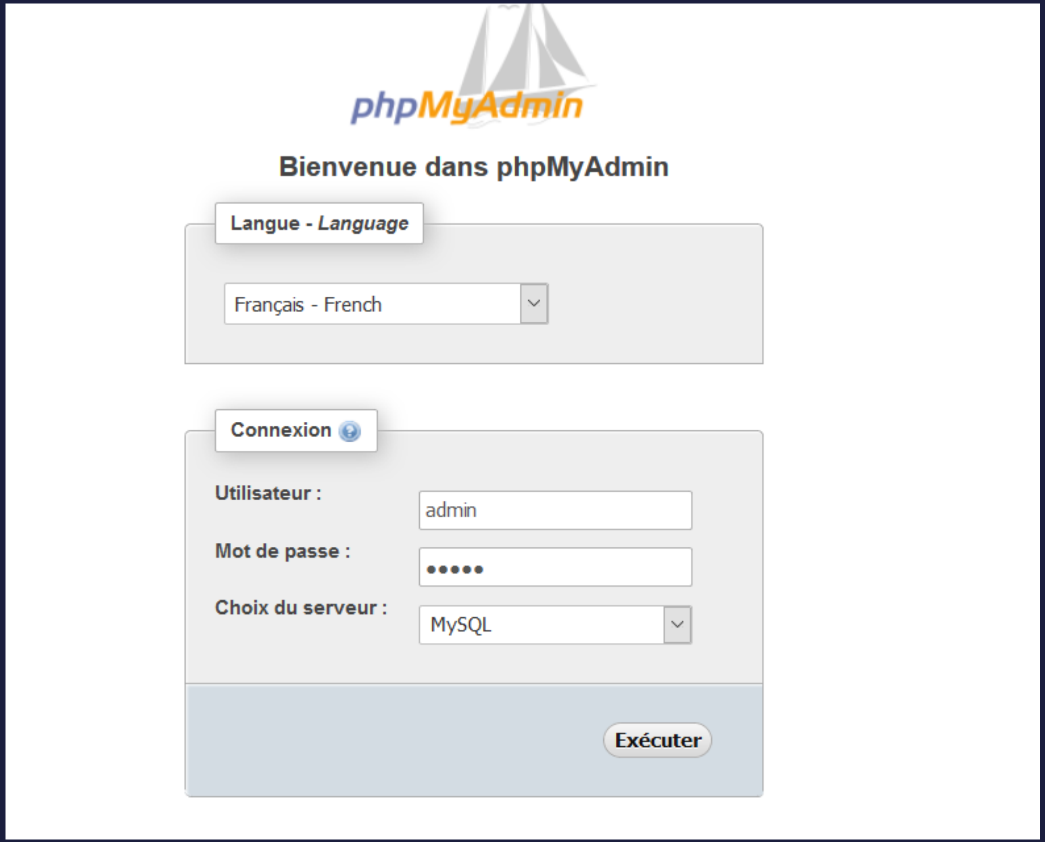
http://php.net/manual/en/pdo.prepared-statements.php

Top 10 2017 – A2

Broken Authentication :

```
https://lol.myapp.com/index.php?session=e2ace639b6848075e9401f2ad4811df2
```

- Default passwords
- Exposition of session ID
- Poor session management
- No bruteforce protection
- …



Bienvenue dans phpMyAdmin

Langue - *Language*

Français - French

Connexion

Utilisateur : admin

Mot de passe : •••••

Choix du serveur : MySQL

Top 10 2017 – A2 : Broken Authentication

- Use a two-step auth (password + OTP by SMS)
- Change default passwords (can be harder than expected)
- Use good CAPTCHA (limited)
- Timeout users after an amount of login try (make it exponential)
- Invalidate session ID on successfull login

```
if ($user_is_allowed_to_login){
    session_regenerate_id(TRUE);
}
```

Top 10 2017 – A3

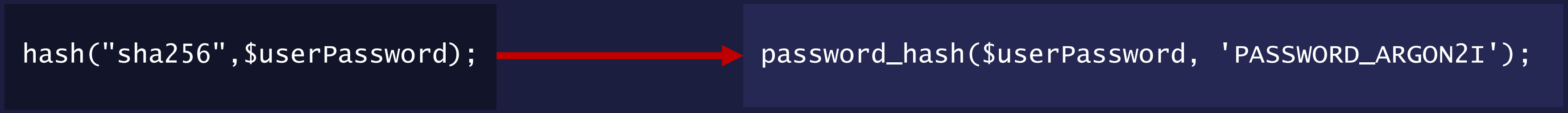
id	username	password	email	creation	lastaccess
1	admin	595ec7599e4a9c5e8a6a96f0a9fc985d	yolo@swag.com	1515760510	1515761510
2	jeanmi	f5355504c5355403b74b6fd440b96c94	jeanmi@const.fr	1515761040	1515761511

Sensitive Data Exposure :

- Bad use of passwords (bad hash, no salt)
- No encryption on sensitive data
- No encryption on communication protocol (HTTP)
- …

Top 10 2017 – A3 : Sensitive Data Exposure

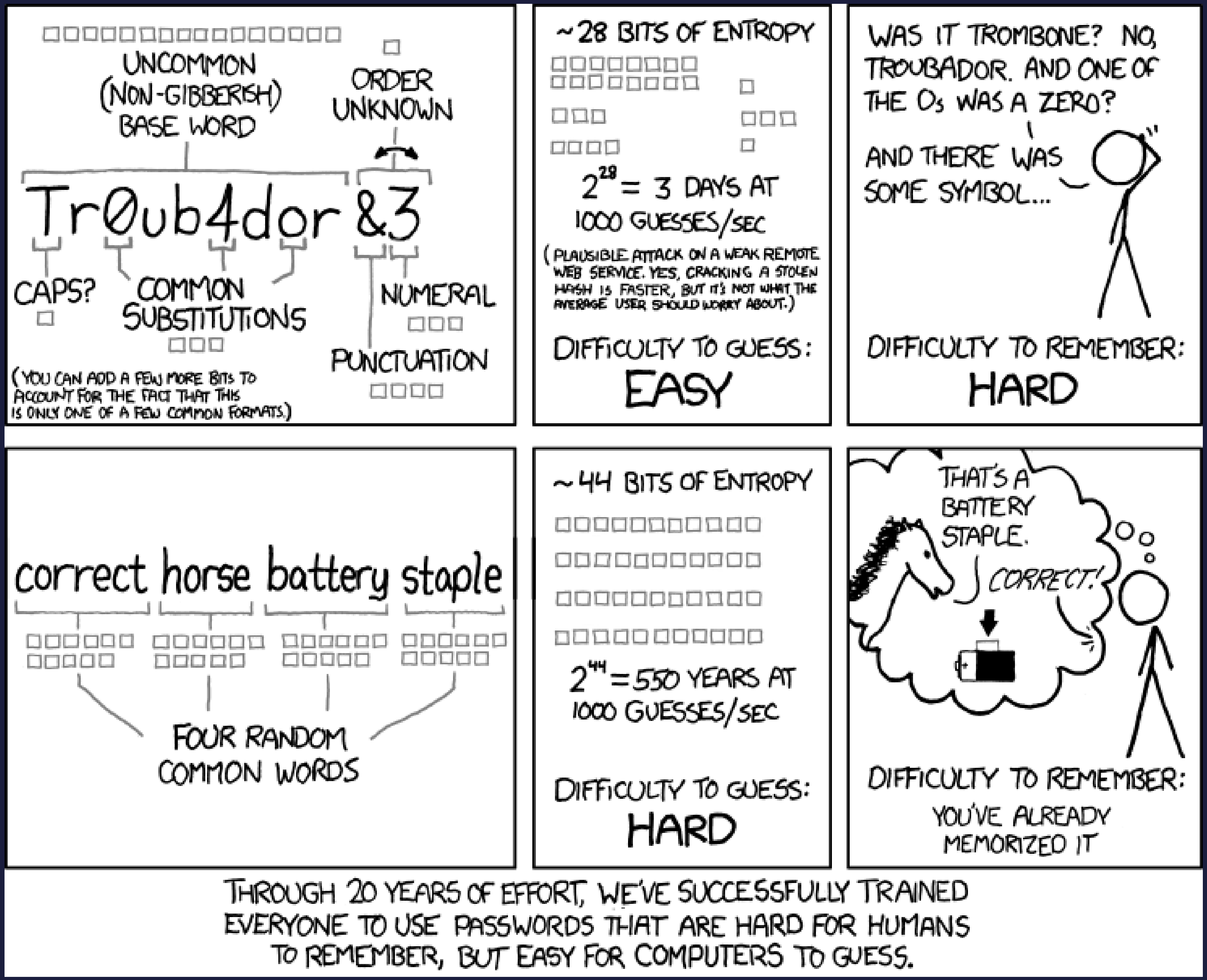
- Use strong *password specialized* hashing functions (Argon2 > Scrypt > Bcrypt > PBKDF2)



```
hash("sha256",$userPassword);
```

```
password_hash($userPassword, 'PASSWORD_ARGON2I');
```

- Use TLS with a good configuration and test it.
- Don’t let users choose bad password by testing it strength.
- Don’t show user the stack trace (printStackTrace in Java)


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor &3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Top 10 2017 – A4

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
 <!ENTITY lol "lol">
 <!ELEMENT lolz (#PCDATA)>
 <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
 <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
 <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
 <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
 <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
 <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
 <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
 <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
 <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

XXE :

- Bad implementation of XML parsers
- Spreading (API prevalence)
- Can be used to :
  - DoS (as above)
  - Read sensitive and technical datas (/etc/passwd, …)
  - …

Top 10 2017 – A4 : XXE

- Use something else than XML (JSON)
- Disable inetrnal definition (internal DTD) and define strict well-defined external DTD

```
libxml_disable_entity_loader(true);
```

```
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setExpandEntityReferences(false);
```

- NOT RECOMMENDED : filter and sanitize every XML you receive. It’s hard to do well.

Top 10 2017 – A5

`https://lol.myapp.com/invoices?id=14567` → Your invoice

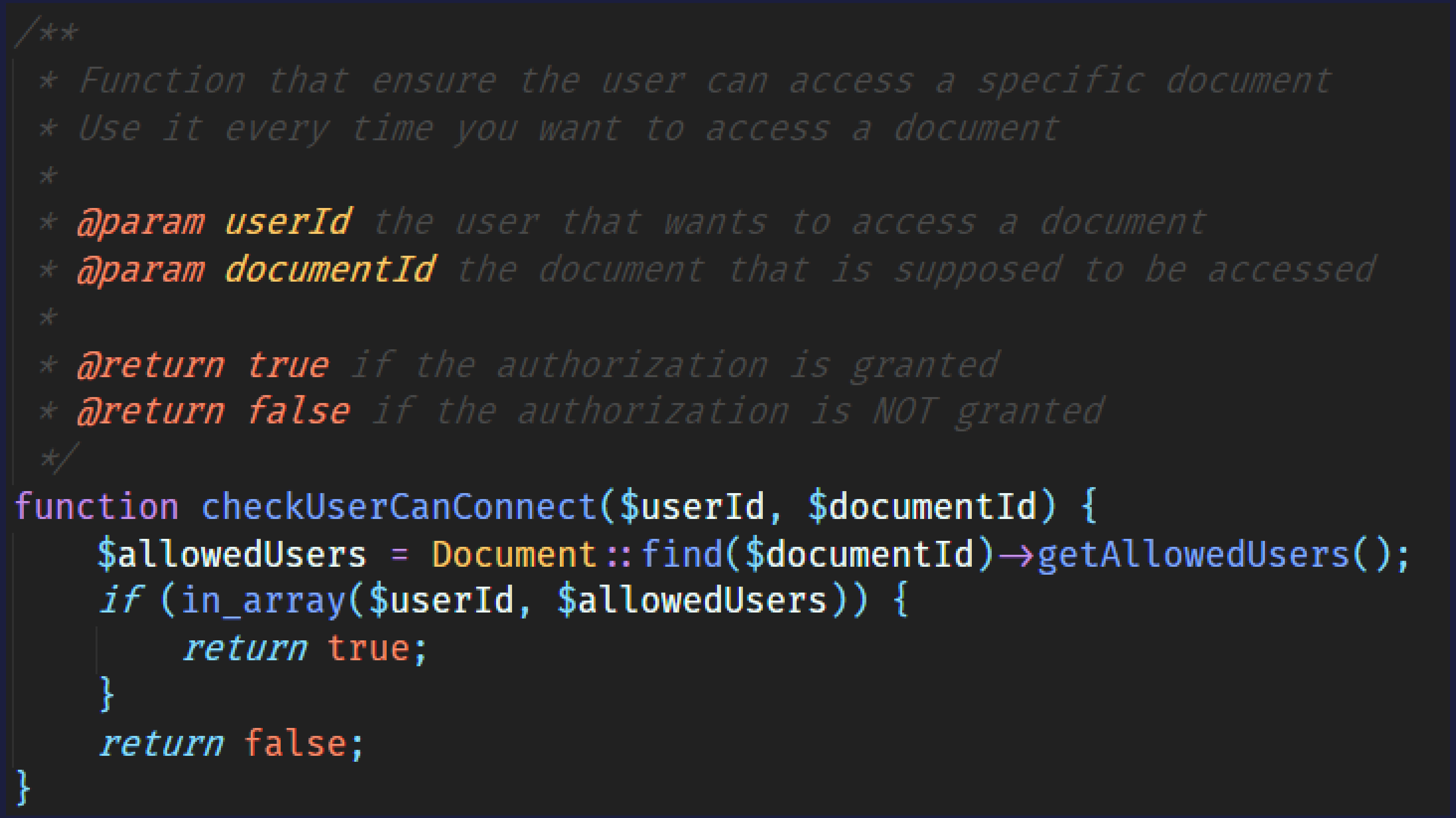
`https://lol.myapp.com/invoices?id=14569` → **Not** your invoice

Broken Access Control :

- Poor / no checking of authorization while accessing ressource
- Can be :
  - A document (see above)
  - The administration functions
  - Almost any data you access directly

Top 10 2017 – A5 : Broken Access Control

Just ensure better SERVER SIDE checking while accessing ressources

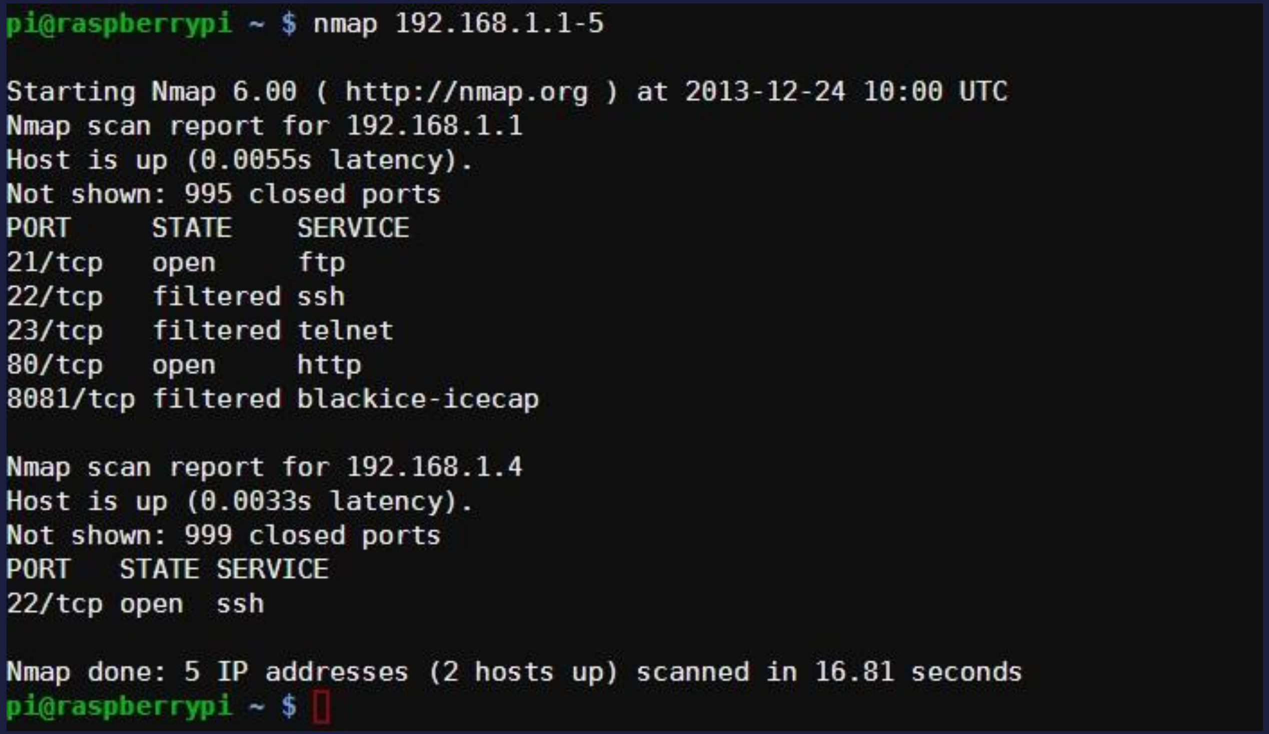


```php
/**
 * Function that ensure the user can access a specific document
 * Use it every time you want to access a document
 *
 * @param userId the user that wants to access a document
 * @param documentId the document that is supposed to be accessed
 *
 * @return true if the authorization is granted
 * @return false if the authorization is NOT granted
 */
function checkUserCanConnect($userId, $documentId) {
    $allowedUsers = Document::find($documentId)->getAllowedUsers();
    if (in_array($userId, $allowedUsers)) {
        return true;
    }
    return false;
}
```

Top 10 2017 – A6

Security Misconfiguration :

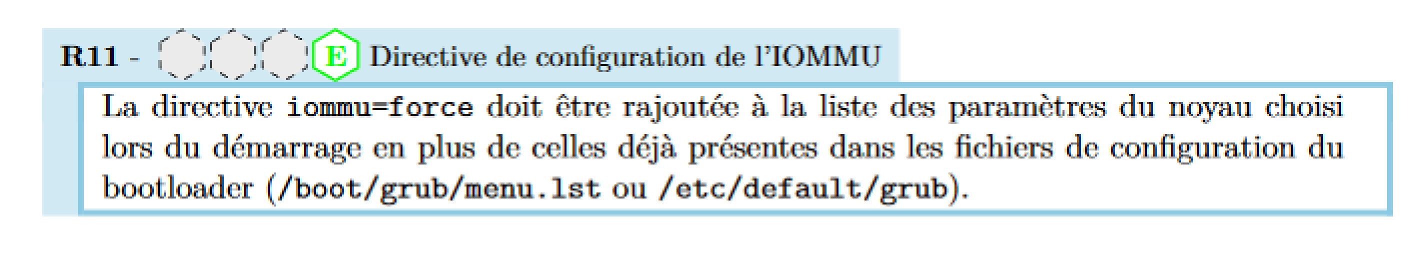
- Unused /useless services
- Bad application security settings (CSRF tokens)
- Vulnerable stack (OS, app server, …)
- …



```
pi@raspberrypi ~ $ nmap 192.168.1.1-5

Starting Nmap 6.00 ( http://nmap.org ) at 2013-12-24 10:00 UTC
Nmap scan report for 192.168.1.1
Host is up (0.0055s latency).
Not shown: 995 closed ports
PORT     STATE    SERVICE
21/tcp   open     ftp
22/tcp   filtered ssh
23/tcp   filtered telnet
80/tcp   open     http
8081/tcp filtered blackice-icecap

Nmap scan report for 192.168.1.4
Host is up (0.0033s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 5 IP addresses (2 hosts up) scanned in 16.81 seconds
pi@raspberrypi ~ $ 
```

Top 10 2017 – A6 : Security Misconfiguration

- Make sure your open ports are useful (SSH !)
- Have security configuration rules (use referencials)
- Run automated config audit
- Make sure during dev you activate built-in security features

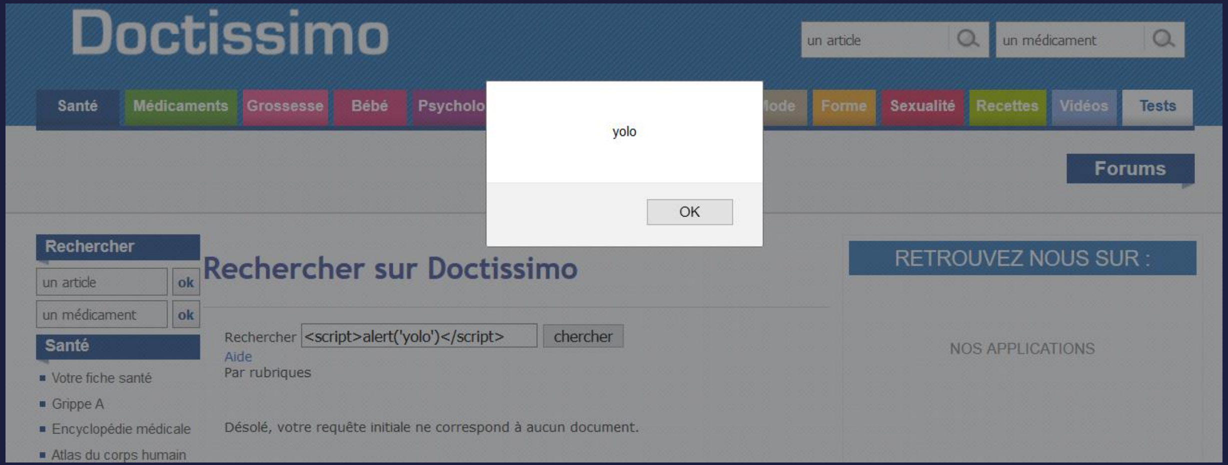


R11 - ⬡ ⬡ ⬡ E Directive de configuration de l'IOMMU

La directive `iommu=force` doit être rajoutée à la liste des paramètres du noyau choisi lors du démarrage en plus de celles déjà présentes dans les fichiers de configuration du bootloader (`/boot/grub/menu.lst` ou `/etc/default/grub`).

Top 10 2017 – A7

```
usr_name=jean-mi&title=h4xor&message=<script>stealUserSession()</script>
```



Rechercher sur Doctissimo

Rechercher <script>alert('yolo')</script>

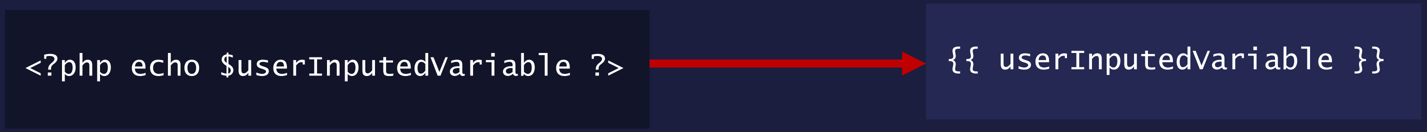
Désolé, votre requête initiale ne correspond à aucun document.

XSS :

- Insertion of elements into HTML
- Usually scripts but also can be html elements
- Can be used to :
  - Redirect users
  - Steal session IDs (with Top 10 - A2)
  - …

Top 10 2017 – A7 : XSS

- Automate secure variable display by :
  - Using template engines (Twig, Pug, Jinja2, …)

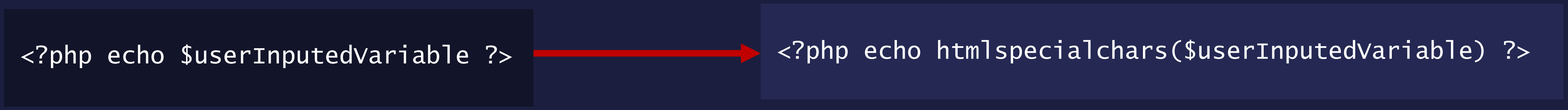


```
<?php echo $userInputedVariable ?>
```

```
{{ userInputedVariable }}
```

  - Using XSS protection systems (OWASP Java Encoder, …)
  - Using pre-built protections in client-side framework (Angular, React, Vue, …)

- NOT RECOMMANDED : for legacy code : unautomated secure display :



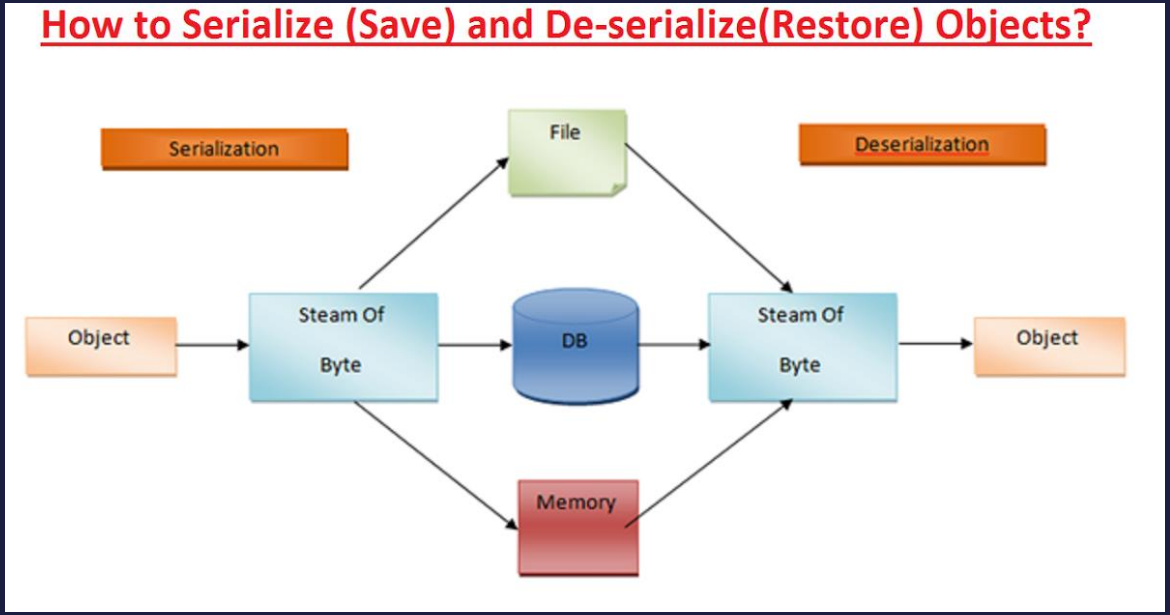
```
<?php echo $userInputedVariable ?>
```

```
<?php echo htmlspecialchars($userInputedVariable) ?>
```

Top 10 2017 – A8

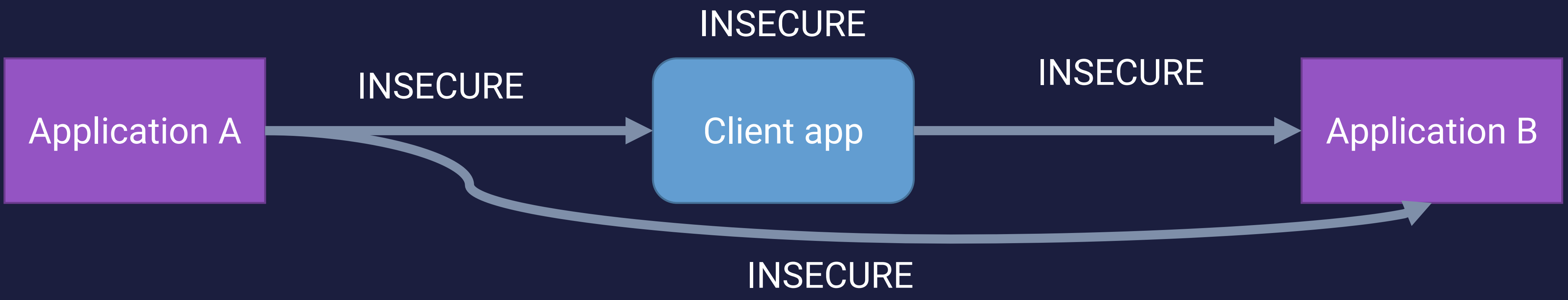
```
a:4:{i:0;i:132;i:1;s:7:"Mallory";i:2;s:4:"user";i:3;s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";}
a:4:{i:0;i:1;i:1;s:5:"Alice";i:2;s:5:"admin";i:3;s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";}
```

Insecure deserialization :



How to Serialize (Save) and De-serialize(Restore) Objects?

Top 10 2017 – A8 : Insecure deserialization



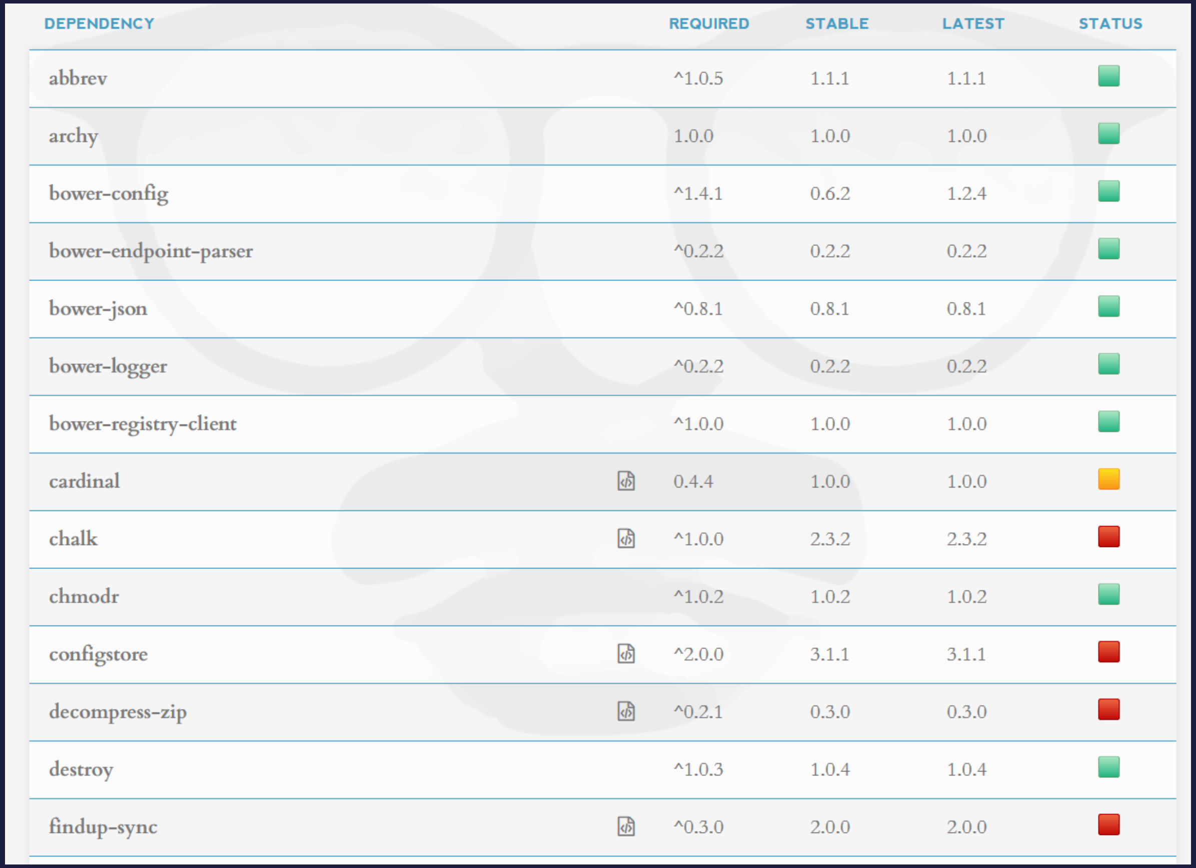
- Do not use serialization (often easy)
- Filter and sanitize user datas (not enough)
- Sign your objects to ensure integrity (HMAC)

https://www.youtube.com/watch?v=nkTBwbnfesQ
https://github.com/GrrrDog/Java-Deserialization-Cheat-Sheet

Top 10 2017 – A9

Vulnerable components :

- Using « fake » component / lib
- Using old versions of :
  - Librairies
  - Tools
  - Components (packages, bundles …)
- Can lead to any other security flaw by exploiting insecure dependency

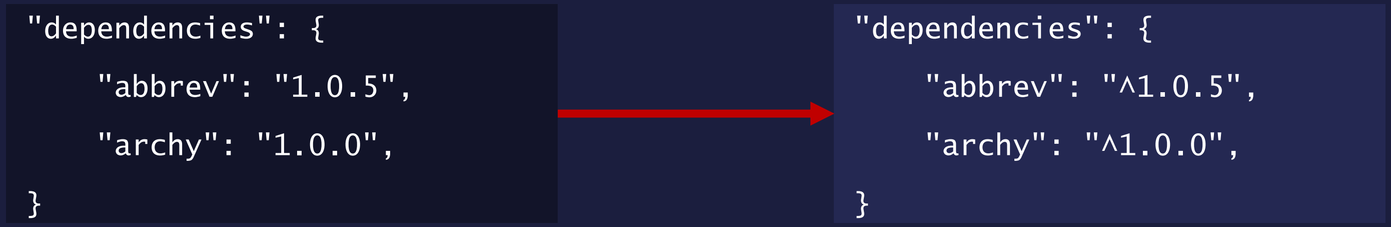


DEPENDENCY		REQUIRED	STABLE	LATEST	STATUS
abbrev		^1.0.5	1.1.1	1.1.1	🟩
archy		1.0.0	1.0.0	1.0.0	🟩
bower-config		^1.4.1	0.6.2	1.2.4	🟩
bower-endpoint-parser		^0.2.2	0.2.2	0.2.2	🟩
bower-json		^0.8.1	0.8.1	0.8.1	🟩
bower-logger		^0.2.2	0.2.2	0.2.2	🟩
bower-registry-client		^1.0.0	1.0.0	1.0.0	🟩
cardinal	📄	0.4.4	1.0.0	1.0.0	🟨
chalk	📄	^1.0.0	2.3.2	2.3.2	🟥
chmodr		^1.0.2	1.0.2	1.0.2	🟩
configstore	📄	^2.0.0	3.1.1	3.1.1	🟥
decompress-zip	📄	^0.2.1	0.3.0	0.3.0	🟥
destroy		^1.0.3	1.0.4	1.0.4	🟩
findup-sync	📄	^0.3.0	2.0.0	2.0.0	🟥

https://david-dm.org/bower/bower

Top 10 2017 – A9 : Vulnerable components

- Remove useless dependencies
- Set dependencies as relatives



```
"dependencies": {
    "abbrev": "1.0.5",
    "archy": "1.0.0",
}
```

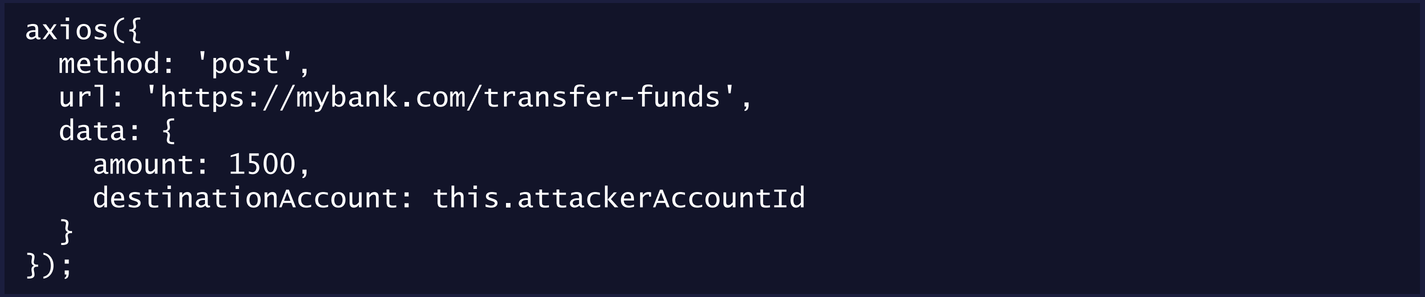
```
"dependencies": {
    "abbrev": "^1.0.5",
    "archy": "^1.0.0",
}
```

- Automatically check your dependencies

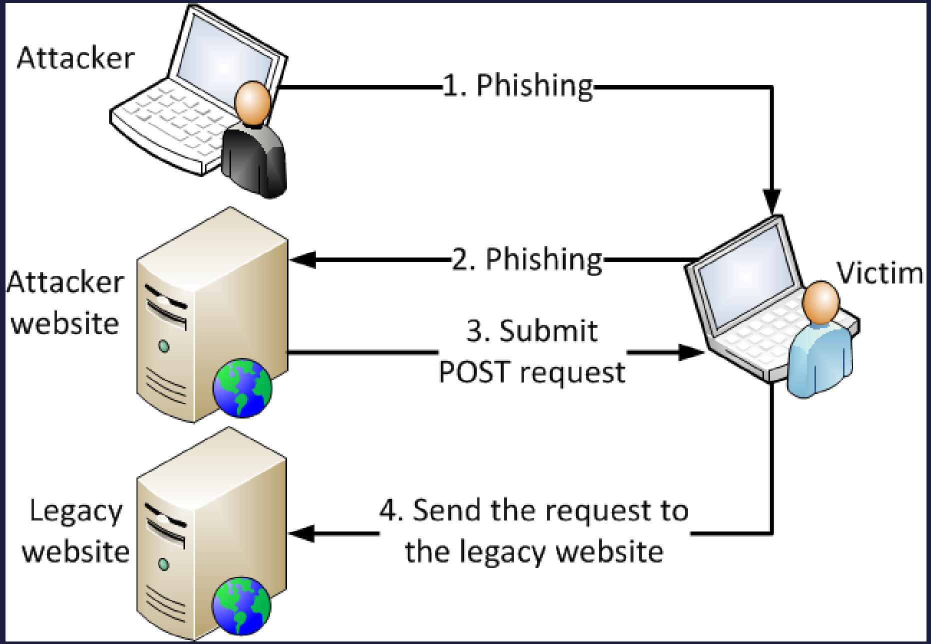
Top 10 2017 – A10 : Insufficient monitoring

- Log pretty much everything in your app
- Export your log into an other server (protect integrity)
- Use your logs for monitoring – make a SOC analyse them with a SIEM solution

Wild card – Top 10 2013 : A8

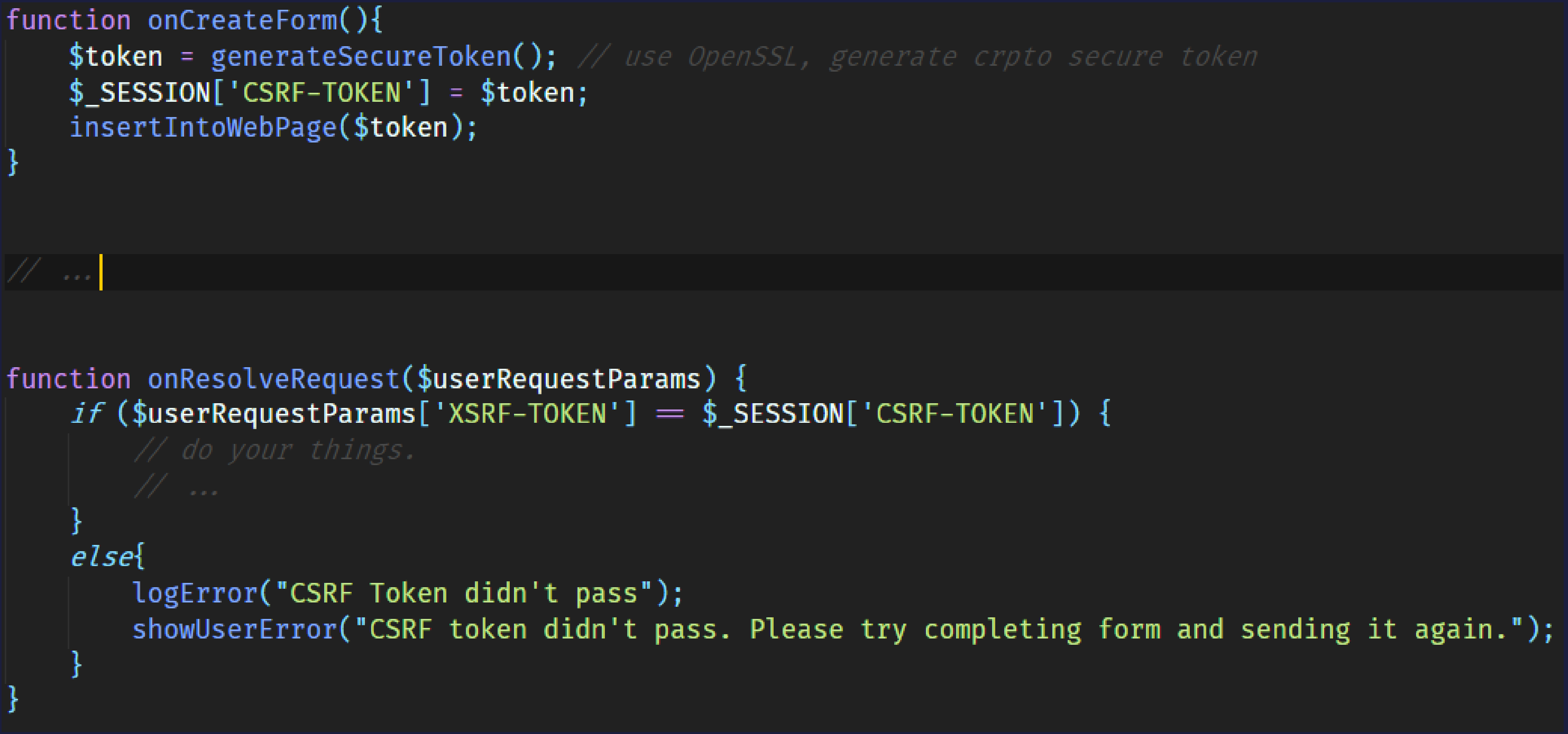


```
axios({
  method: 'post',
  url: 'https://mybank.com/transfer-funds',
  data: {
    amount: 1500,
    destinationAccount: this.attackerAccountId
  }
});
```



Wild card – CSRF (Top 10 2013 : A8)

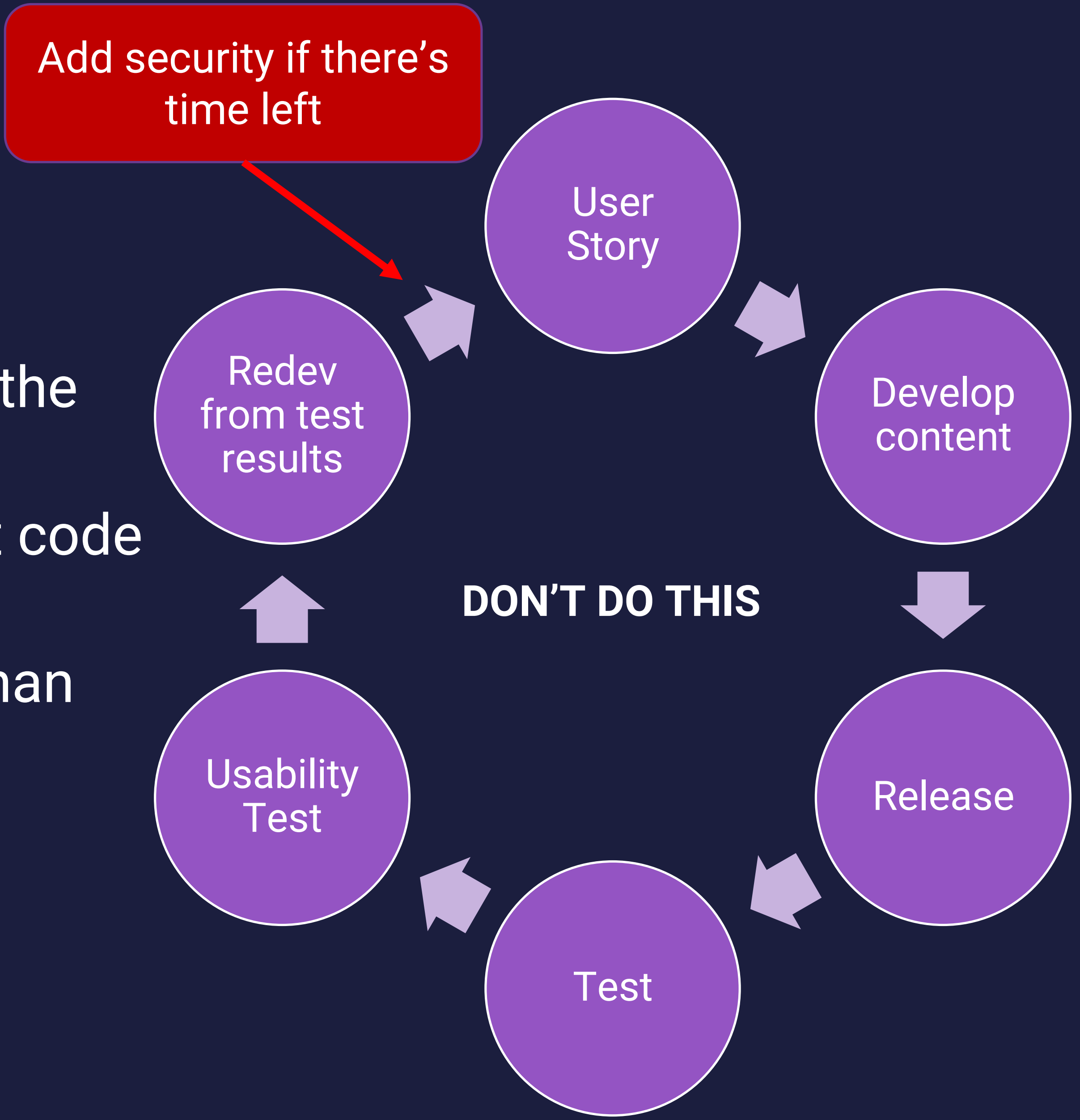
Best defense is synchronizer pattern. Included in all modern frameworks



```
function onCreateForm(){
    $token = generateSecureToken(); // use OpenSSL, generate crpto secure token
    $_SESSION['CSRF-TOKEN'] = $token;
    insertIntoWebPage($token);
}



// ...



function onResolveRequest($userRequestParams) {
    if ($userRequestParams['XSRF-TOKEN'] == $_SESSION['CSRF-TOKEN']) {
        // do your things.
        // ...
    }
    else{
        logError("CSRF Token didn't pass");
        showUserError("CSRF token didn't pass. Please try completing form and sending it again.");
    }
}
```

Conclusion

- Technical and non-technical
- Security must be present from the beginning to the end
- Devs must be concerned about code security
- Basic defense is often easier than expected



Ressources

- https://www.owasp.org
- https://www.owasp.org/images/7/72/OWASP_Top_10-2017_(en).pdf.pdf
- https://www.hacksplaining.com
- https://cwe.mitre.org/data/
- https://www.cert.ssi.gouv.fr/information/ (FR)

Thank you