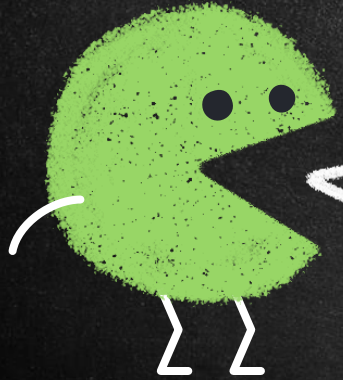
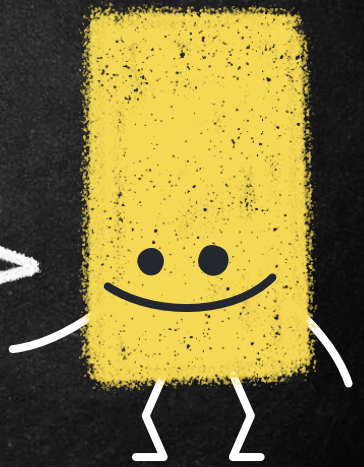
Sécurité des données

H2022

Vulnérabilités Web

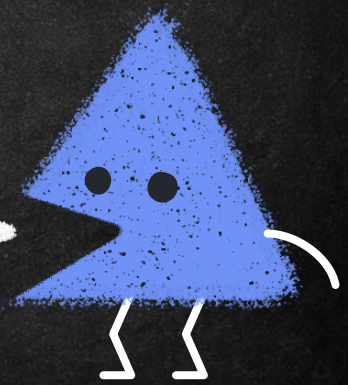




Yannick Charron

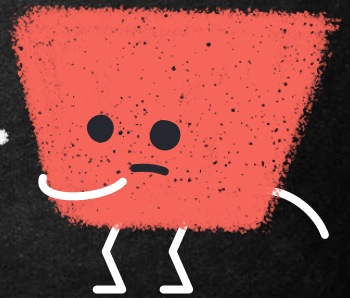
Plan de la séance

- Tour d'horizon
- Injection
  - Base de données
  - XSS (Cross-Site Scripting)

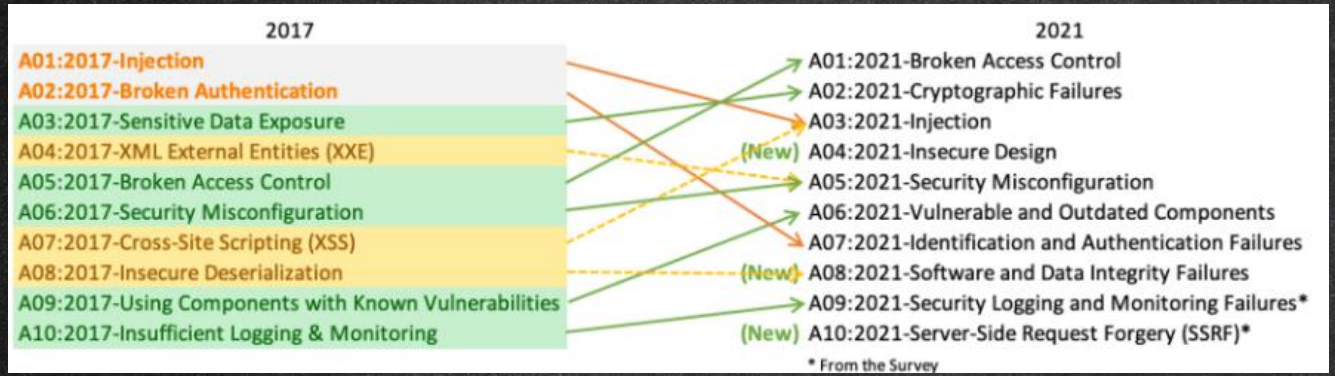


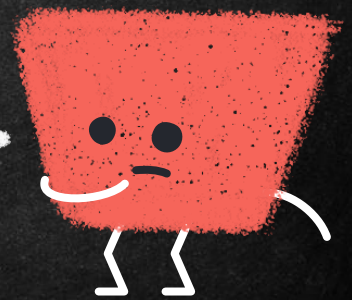
OWASP

- Open Web Application Security Project
- https://owasp.org/
- Ressources intéressantes
  - Top Ten
  - Juice Shop



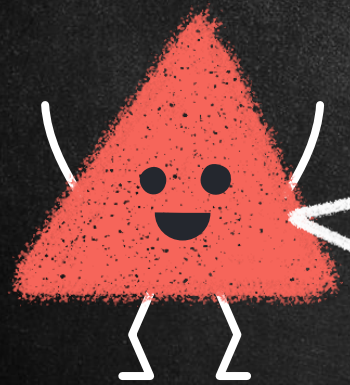
Top 10

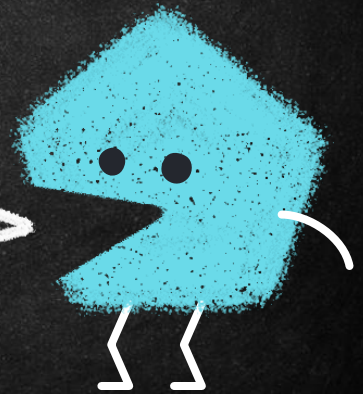

2017
A01:2017-Injection
A02:2017-Broken Authentication
A03:2017-Sensitive Data Exposure
A04:2017-XML External Entities (XXE)
A05:2017-Broken Access Control
A06:2017-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)
A08:2017-Insecure Deserialization
A09:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring
2021
A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
(New) A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
(New) A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
(New) A10:2021-Server-Side Request Forgery (SSRF)*
* From the Survey



Injection

SQL Injection





Injection SQL

- Accès complet à la base de données via une instruction malveillantes

Injection SQL

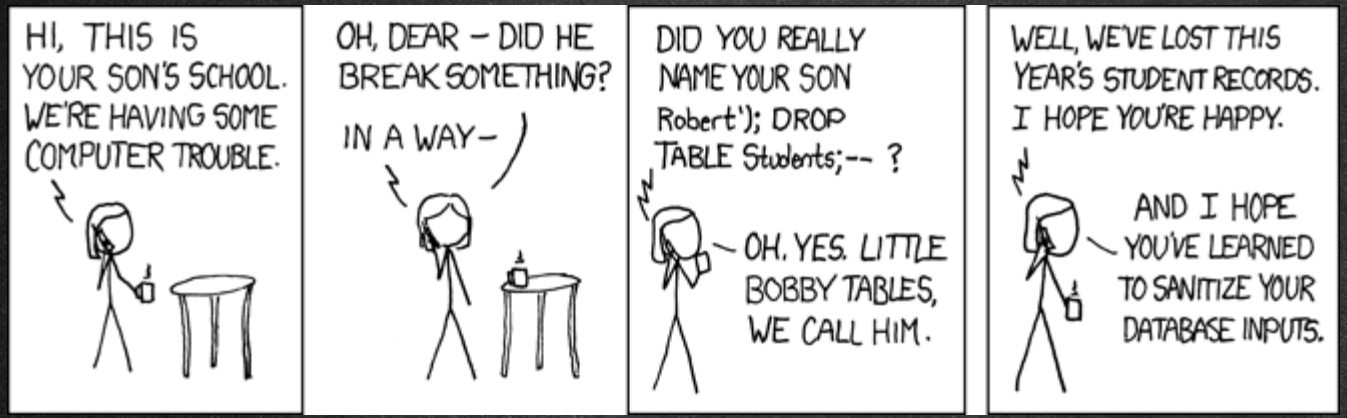
```
SELECT * FROM users
WHERE username = '".$username."'
AND password = '".$passwordHash."';

SELECT * FROM users
WHERE username = 'yannick'
AND password = 'monMotDePasseHashé';
```

Injection SQL

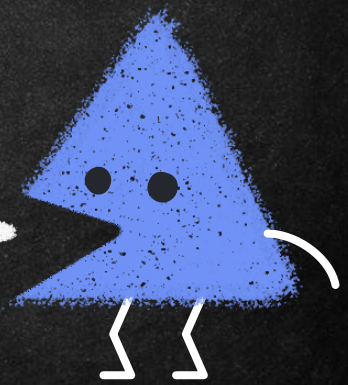
```
SELECT * FROM users
WHERE username = 'admin';--' AND password =
monMotDePasseHashé;

SELECT * FROM users
WHERE username = 'admin'
AND password = 'monMotDePasseHashé' OR 1=1--';
```


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

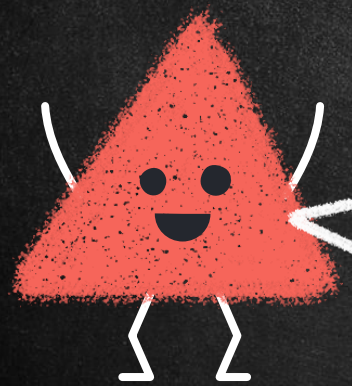
Comment se protéger

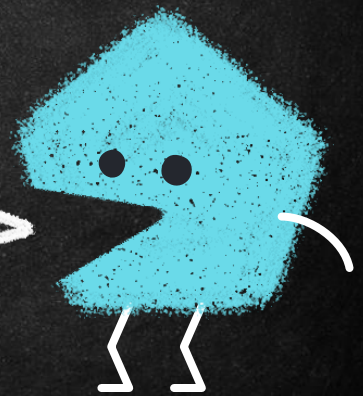
- Utilisateur avec privilèges limités
- OWASP Cheat Sheat



Injection

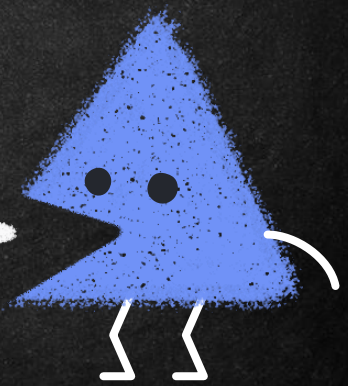
Cross-Site Scripting - XSS





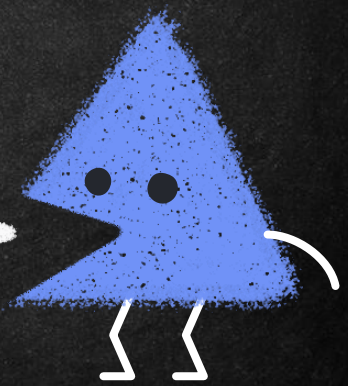
XSS

- Injection de code malveillant (javascript)
- Le code est généralement envoyé à un utilisateur via un lien généralement
- Le script peut accéder au cookie, session token ou d'autres informations sensible conservé sur le fureteur



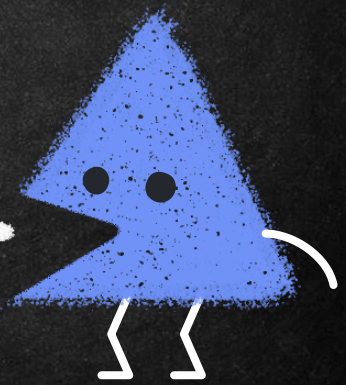
Différents types

- Stored
  - Script sauvegardé sur le serveur base de données
  - La victime reçoit l'attaque via la visite sur le site Web (forum, commentaire)



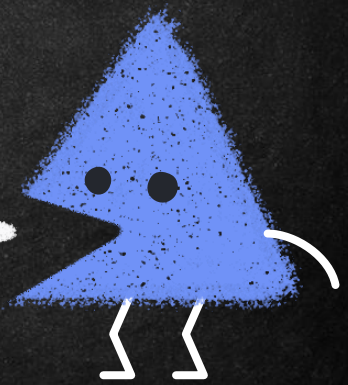
Différents types

- Reflected
  - Script injecté dans un champ de formulaire, dans le requête http
  - L’attaque est provoquée via un autre moyen : courriel, site Web, lien

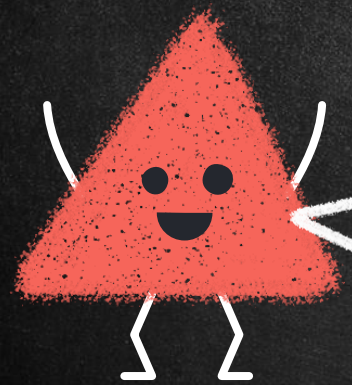


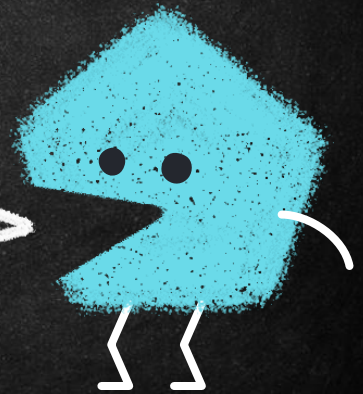
Comment se protéger

- OWASP Cheat Sheat



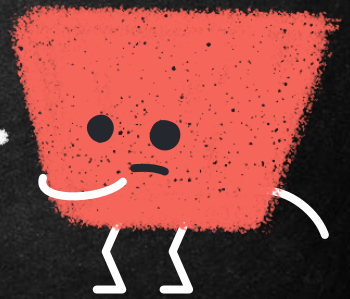
Broken Access Control





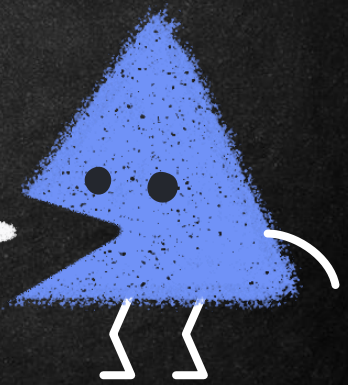
Broken Access Control

- Principe qu'un utilisateur doit agir avec les privilège qui lui sont octroyés.
- Lorsque ce principe n'est pas respecté, nous sommes devant un *Broken Access Control*.
- *Ce qui peut amener typiquement à des fuites modification ou destruction de données*



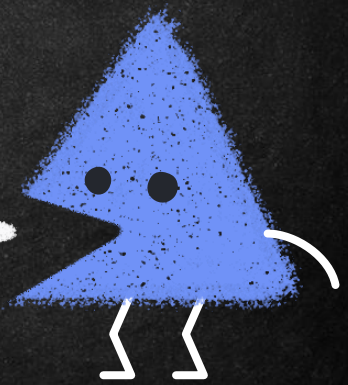
Broken Access Control

- Le principe du moins de privilège ou du zéro privilège n'est pas respecté.
- Outrepasser le contrôle en modifiant des URL, le DOM, ou les requête au API
- Voir ou éditer l'espace personnel d'un autre utilisateur via un identifiant unique
- Utilisation d'un API sans contrôle pour les méthode POST, PUT, PATCH, DELETE
- Mauvais configuration de CORS
- ...



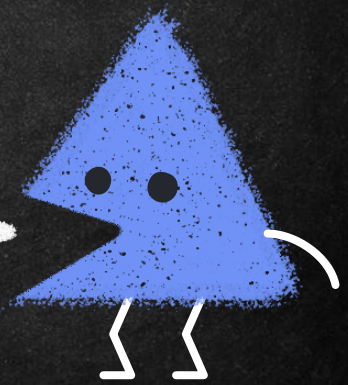
CSRF (Cross-Site Request Forgery)

- Force un utilisateur authentifié à envoyer une requête à une application Web pour laquelle il est authentifié.
- Cet type d'attaque exploite une vulnérabilité dans l'application Web qui n'est pas en mesure de s'assurer que la requête a générer avec le consentement de l'utilisateur.



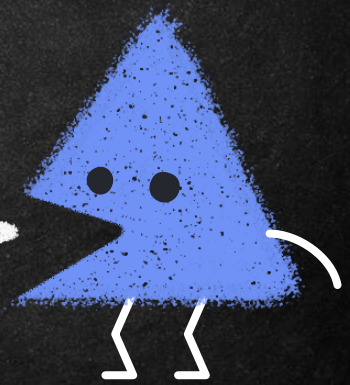
CSRF (Cross-Site Request Forgery)

- L'attaque pour être considéré CSRF doit avoir l'effet d'un changement d'état.

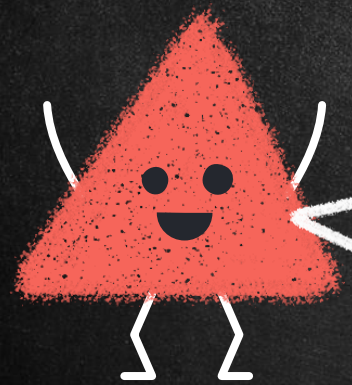


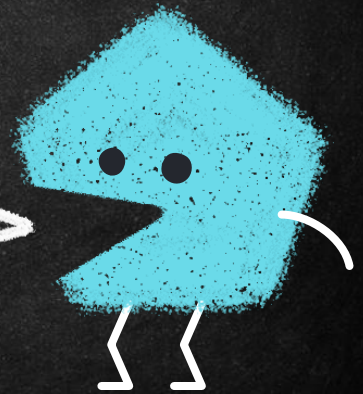
Comment se protéger

- OWASP Cheat Sheat



Broken Anti Automation





Broken Anti-Automation

- Automatisation
  - Mécanisme pour contrôler que des actions qui pourrait fait à multiple reprise par des bots.