Escàner de vulnerabilitats web

Alumne: Eric Bujeque Escorriola
Caps de projecte: Jordi Duch i Agusti Solanas

ÍNDEX

INTRODUCCIÓ

- Context i justificació del Treball

  - Hi ha una clara tendència a fer servir la **tecnologia web**.

  - Existeixen **aplicacions web** que guarden documents o informació confidencial.

vulnerables a intrusos

  - Cal assegurar-nos que cada aplicació web que pengem no siguin vulnerable.

  - **Idea**: Fer servir eines gratuïtes que passen diversos tests especialitzats a les webs, i desenvolupar alguns propis ajuntant-los en una única aplicació web, i que a més sigui capaç d'emmagatzemar els resultats per a la seva posterior comparació.

INTRODUCCIÓ

- Objectius:

- **Desenvolupar una aplicació** que permeti escanejar pàgines i analitzar els resultats.

- Que l'aplicació sigui **fàcil d'utilitzar.**

- Aconseguir un **disseny agradable i intuïtiu.**

- Que permeti **guardar els resultats** per la seva posterior comparació.

- *Aprendre a desenvolupar una aplicació web en Python i Django i aprofundir els coneixements en la seguretat web.*

INTRODUCCIÓ

- Enfocament i mètode seguit:

Possibles estratègies:

- Modificar una aplicació existent.
- Desenvolupar una aplicació nova.
- **Desenvolupar una aplicació nova que es recolzi de programes existents.**

¿Com fer la aplicació?:

- Aplicació d'escriptori.
- **Aplicació web.**

INTRODUCCIÓ

- Producte obtingut:

- Un **escàner de vulnerabilitats web** especialitzat en l'entrada de dades dels usuaris.

- Els usuaris registrats poden **escanejar** les web desitjades passant un seguit de tests de vulnerabilitats que poden escollir.

- L'aplicació mostra els resultats dels tests i els guarda en la **base de dades**.

- Alguns dels test de vulnerabilitats que es passen fan servir aplicacions de tercers. Altres estan programats en la pròpia aplicació fent peticions HTTP sobre la pàgina a escanejar.

SOFTWARE

- Software utilitzat i alternatives:

Software utilizat	Alternativa
Programació: **Python + Django**	- PHP - Java
Servidor: **Propi de Django**	- Apache
Base de dades: **SQLite**	- MySQL
Editors: **IntelliJ Idea**	- Sublime Text 3
Sistema operatiu: **Linux (Fedora)**	- Windows

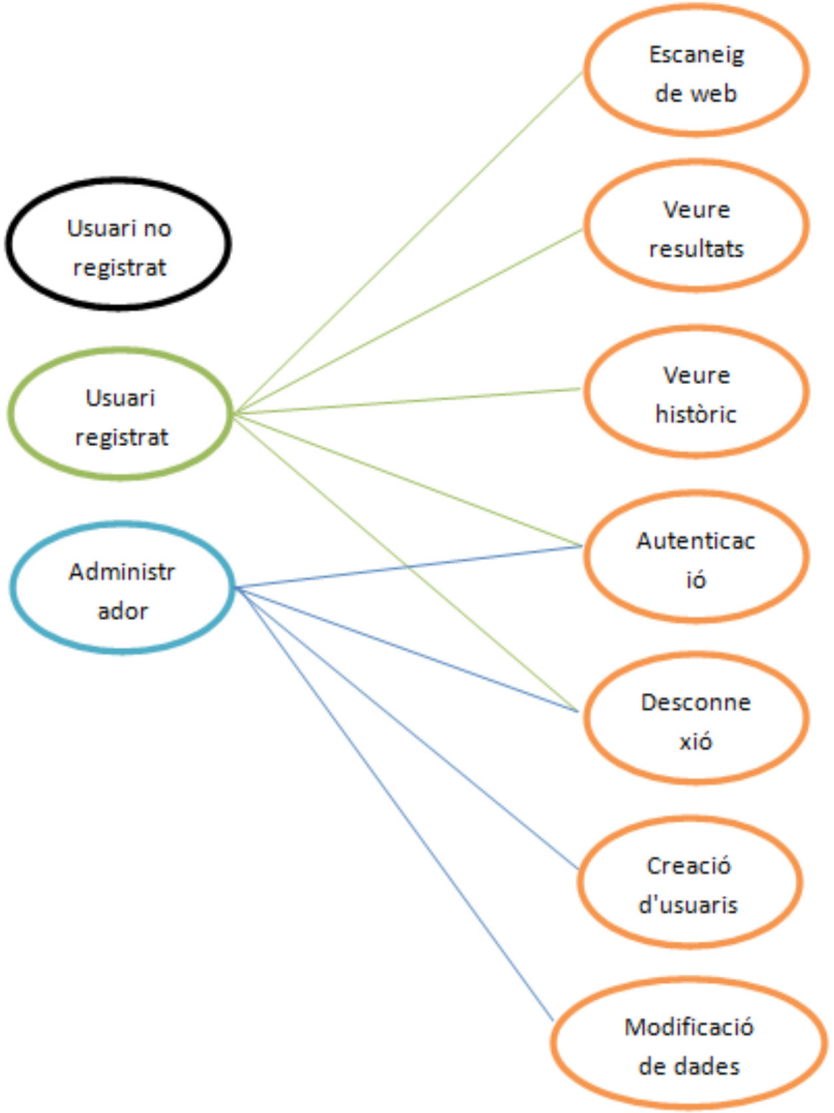
ANÀLISI

Requeriments funcionals:

- Sistema operatiu: **Linux (Fedora).**
- Llenguatge: **Python versió 3.**
- Servidor i Framework: **Django 1.8.2.**
- Base de dades: **SQLite 3.8.10.2.**

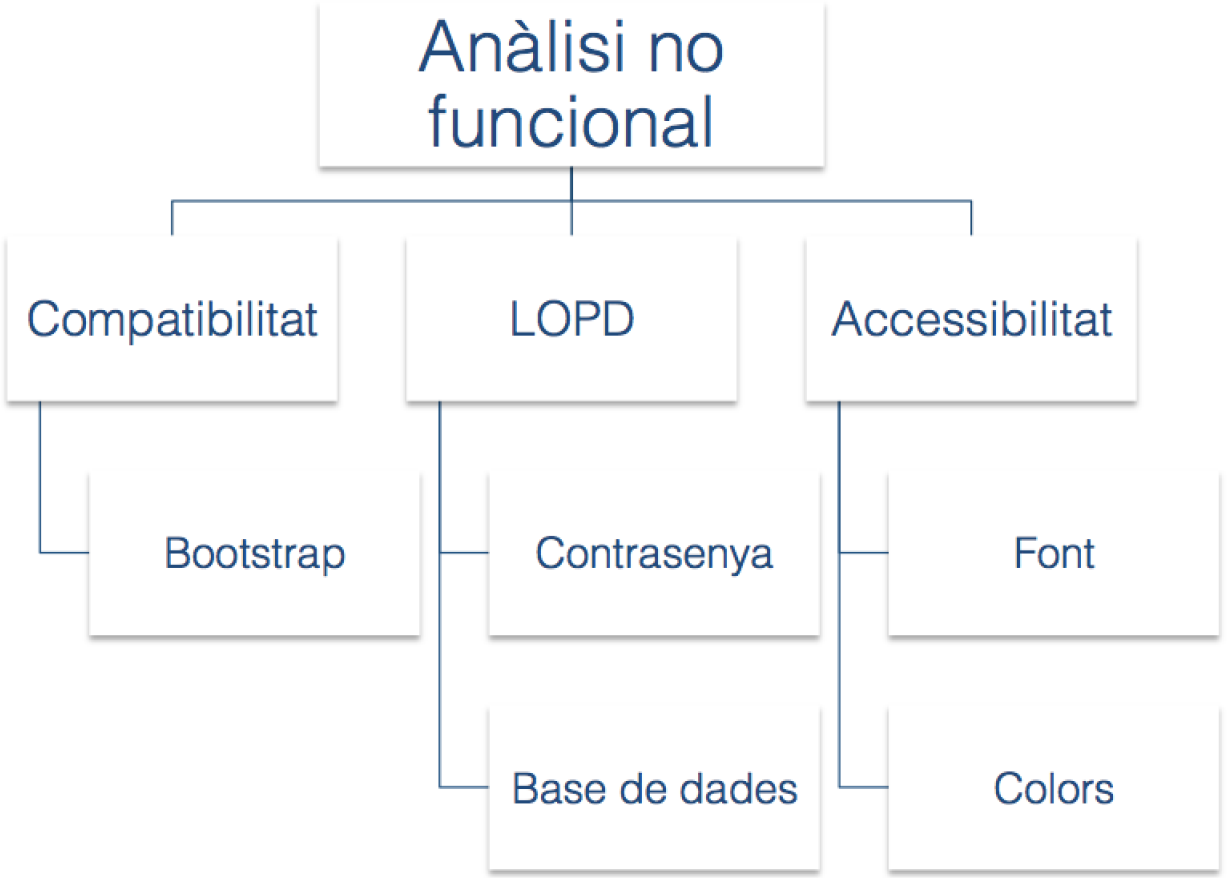
ANÀLISI

Diagrama de casos d'ús :



ANÀLISI

- Anàlisi no funcional:

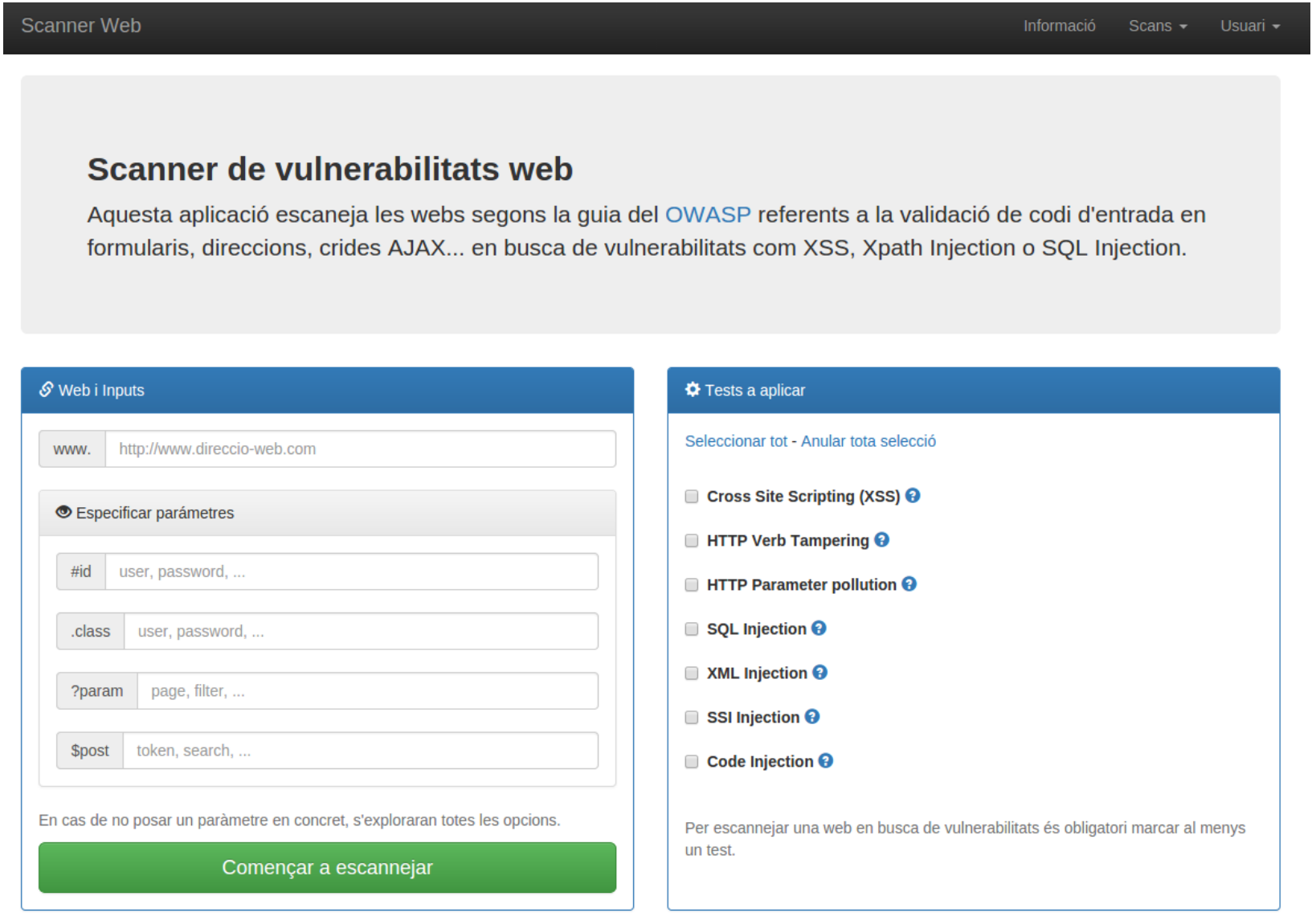


DISSENY

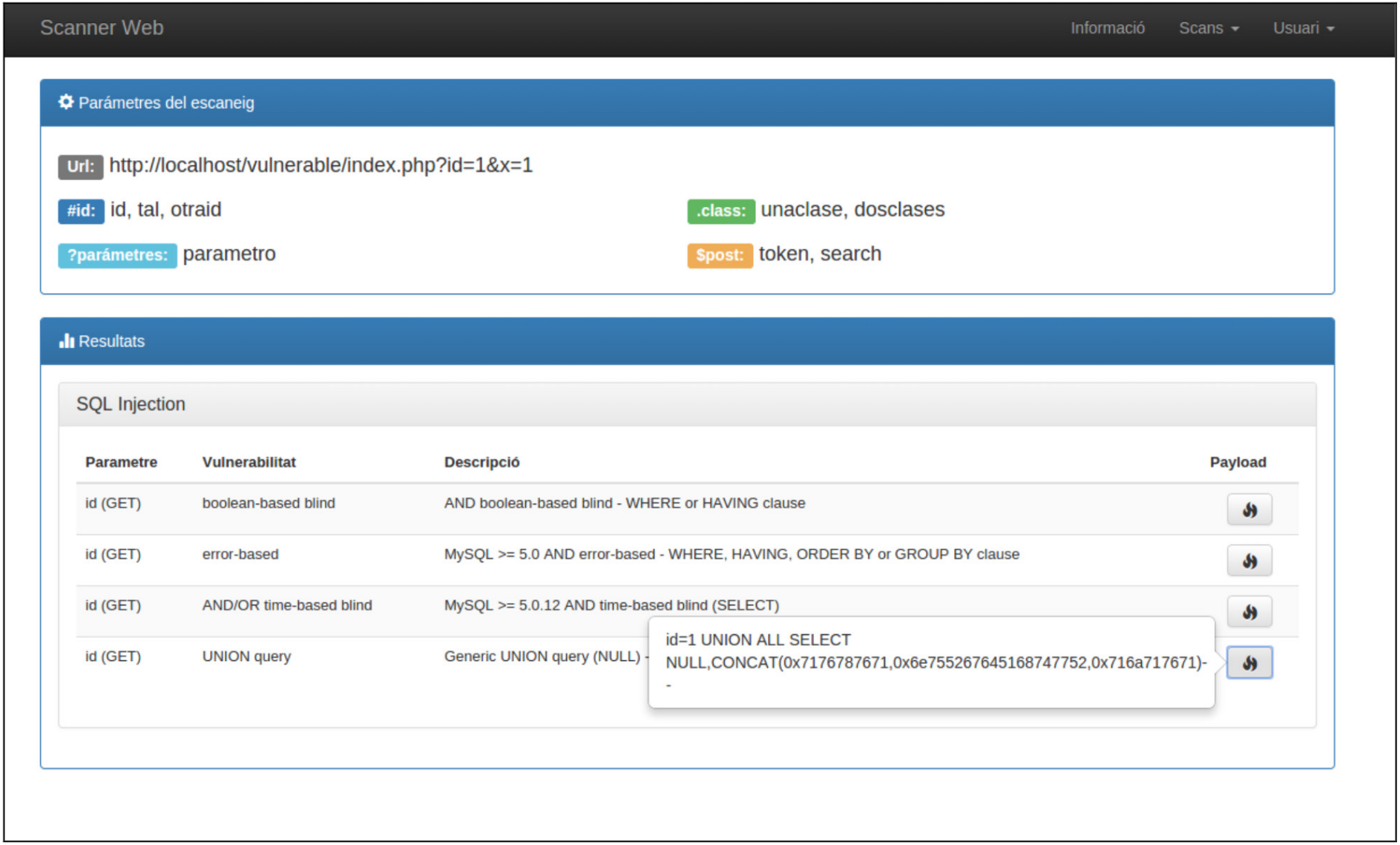
- Estructura de la web:
    - Simple e intuïtiu

Autenticació

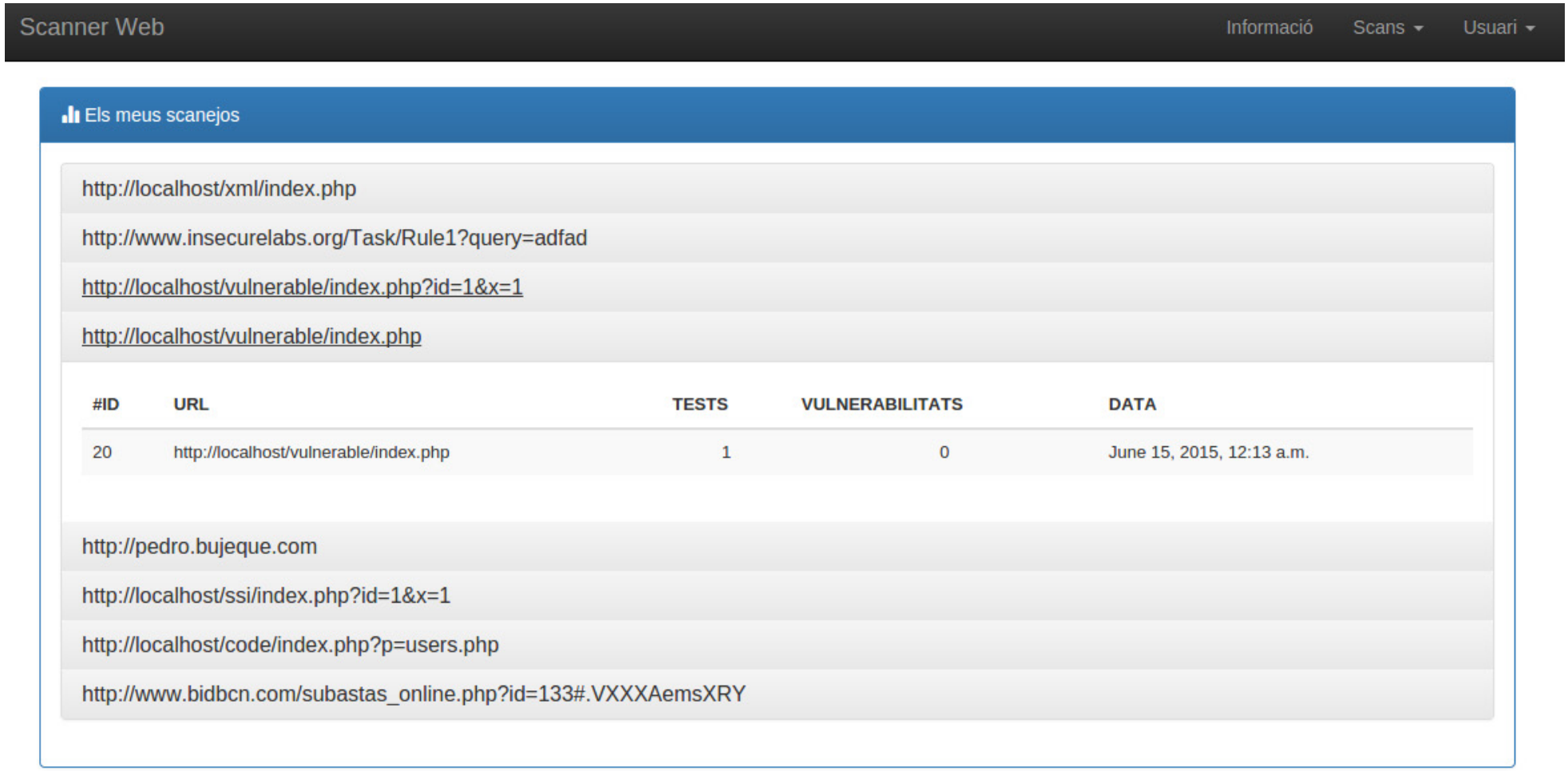
Pàgina d'escaneig



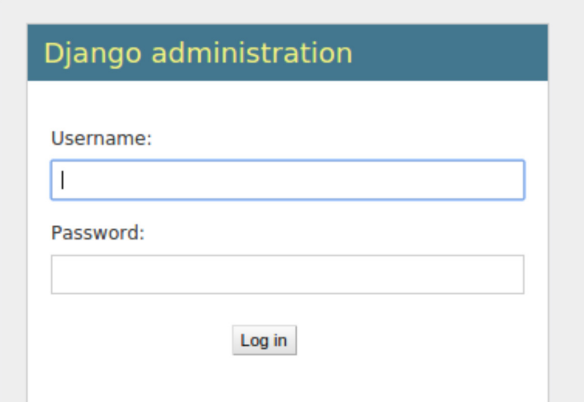
Pàgina de resultats



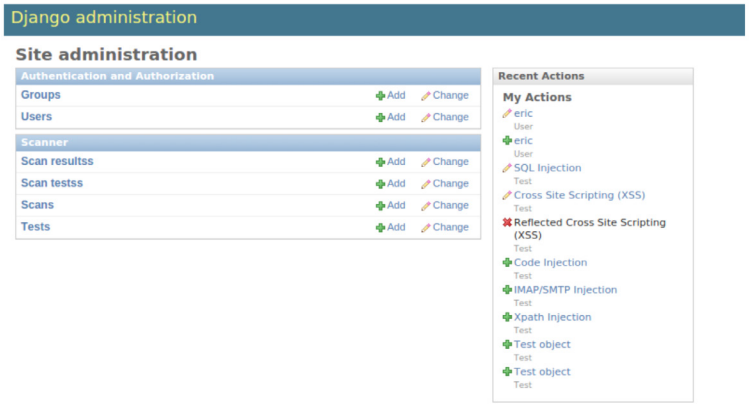
Pàgina d'escaneig



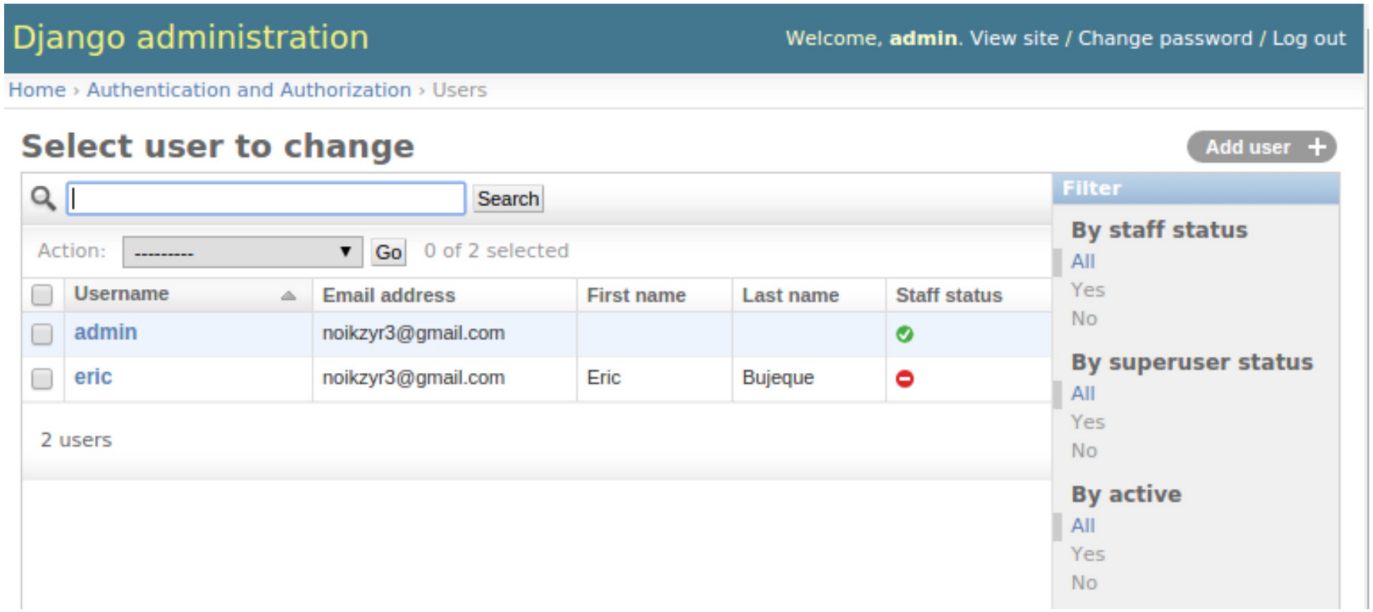
Administració



Pantalla d'autenticació de l'administrador

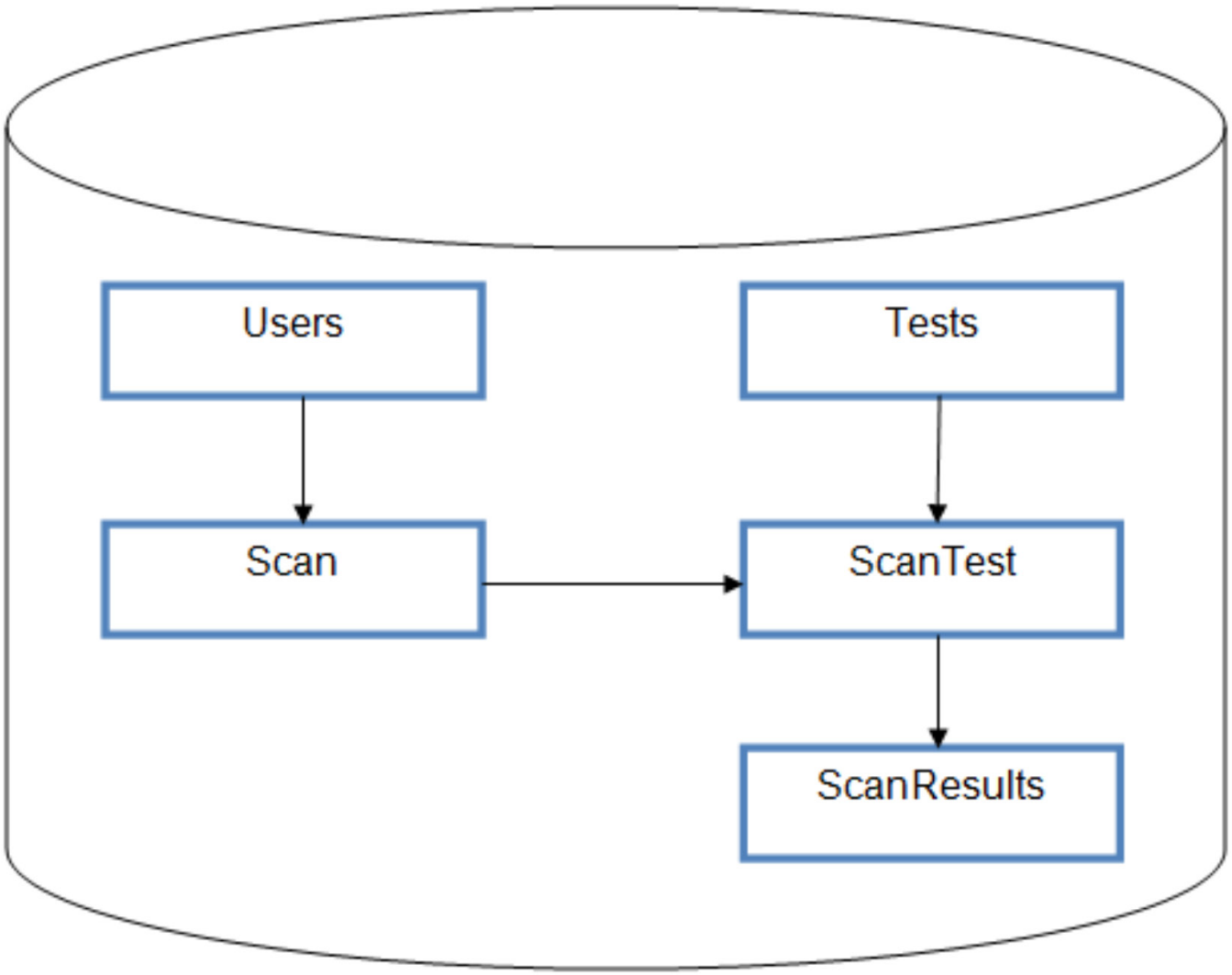


Pantalla de l'administrador



Pantalla d'administració d'usuaris

Model de dades



SEGURETAT

Base de dades

- SQLite utilitza un sistema de xifrat "*Caesar cipher*"

Control de sessions

- Per a cada vista que necessiti estar autenticat hi ha posat un "decorator" de Django anomenat ***@login_required()***.

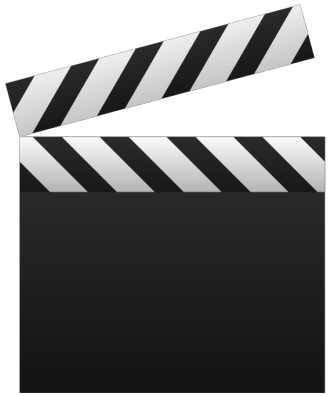
Xifratge de contrasenyes

- Django s'encarrega de xifrar les contrasenyes amb el sistema **PBKDF2.**

Protecció CSRF per formularis

- Protecció que se ha de implementar. S'ha d'enviar dintre dels formularis de les plantilles: **{% csrf_token %}**.

DEMOSTRACIÓ



CONCLUSIONS

- Treball molt enriquidor a nivell personal.

- S'han assolit els objectius principals.

- No s'ha seguit la planificació inicial, i s'ha concentrat el treball en les etapes inicials i finals del projecte.

- Les líneas futures són aspectes a millorar com:

    - Interacció de l'usuari amb els escanejos.
    - Eficiència dels tests.
    - Millores visuals.
    - Afegir més tests.