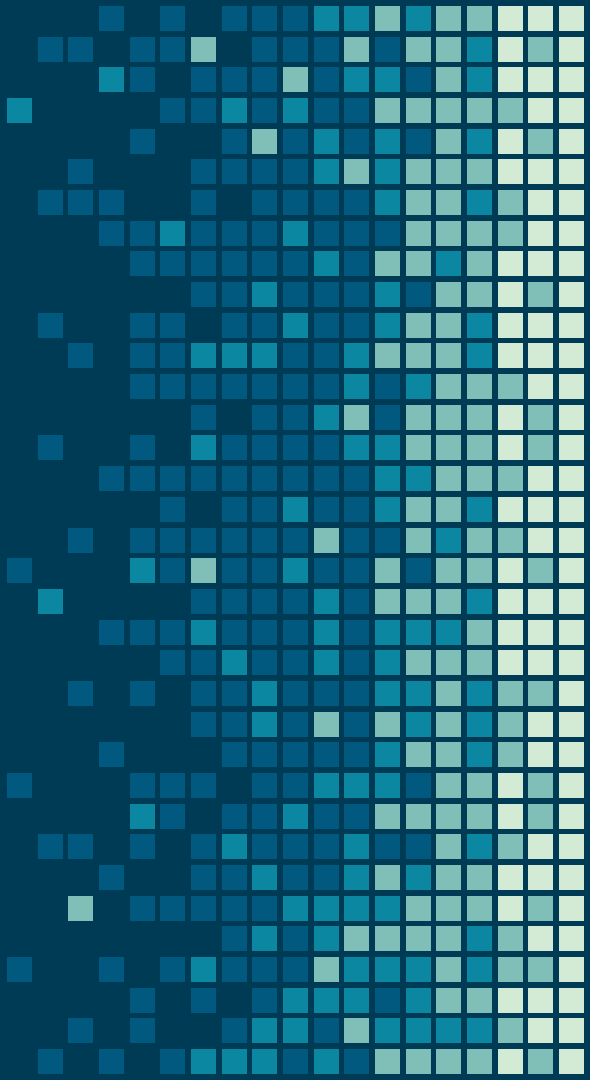
SQL injection

Autor: Bruno Chenoll



Qué es SQLi

Inyectar sentencias maliciosas a través de un campo de entrada para que se ejecuten en una base de datos

SELECT * FROM usuarios WHERE email='$mail' and password='$passwd'

Típica consulta en php para consultar la base de datos

A menos que el contenido de $mail o $passwd esté hardcodeado en el código, necesitaremos pedírselo al usuario

¿Puede meter cualquier cosa? Mientras pase nuestras comprobaciones de si es un email o si la contraseña tiene un mínimo de caracteres sí.

SELECT * FROM usuarios WHERE email='$mail' and password='$passwd'

¿Qué sucede si introducimos un email y de contraseña "' or 1=1--"?

Nada bueno(o sí).

SELECT * FROM usuarios WHERE email='admin@ugr.es' and password='' or 1=1--'

Lógica -> x or TRUE = TRUE

Podemos acceder a la cuenta de admin@ugr.es sin necesidad de saber su contraseña

Fallo descubierto en 1998

Tiene más de 20 años

T10 OWASP Top 10 Application Security Risks – 2017

A1:2017-Injection	Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2:2017-Broken Authentication	Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
A3:2017-Sensitive Data Exposure	Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
A4:2017-XML External Entities (XXE)	Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
A5:2017-Broken Access Control	Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

¿Cómo solucionarlo?

- Filtrando caracteres raros
- Escapando caracteres (\')
- Analizar entrada en busca de patrones de ataque
- Regular permisos

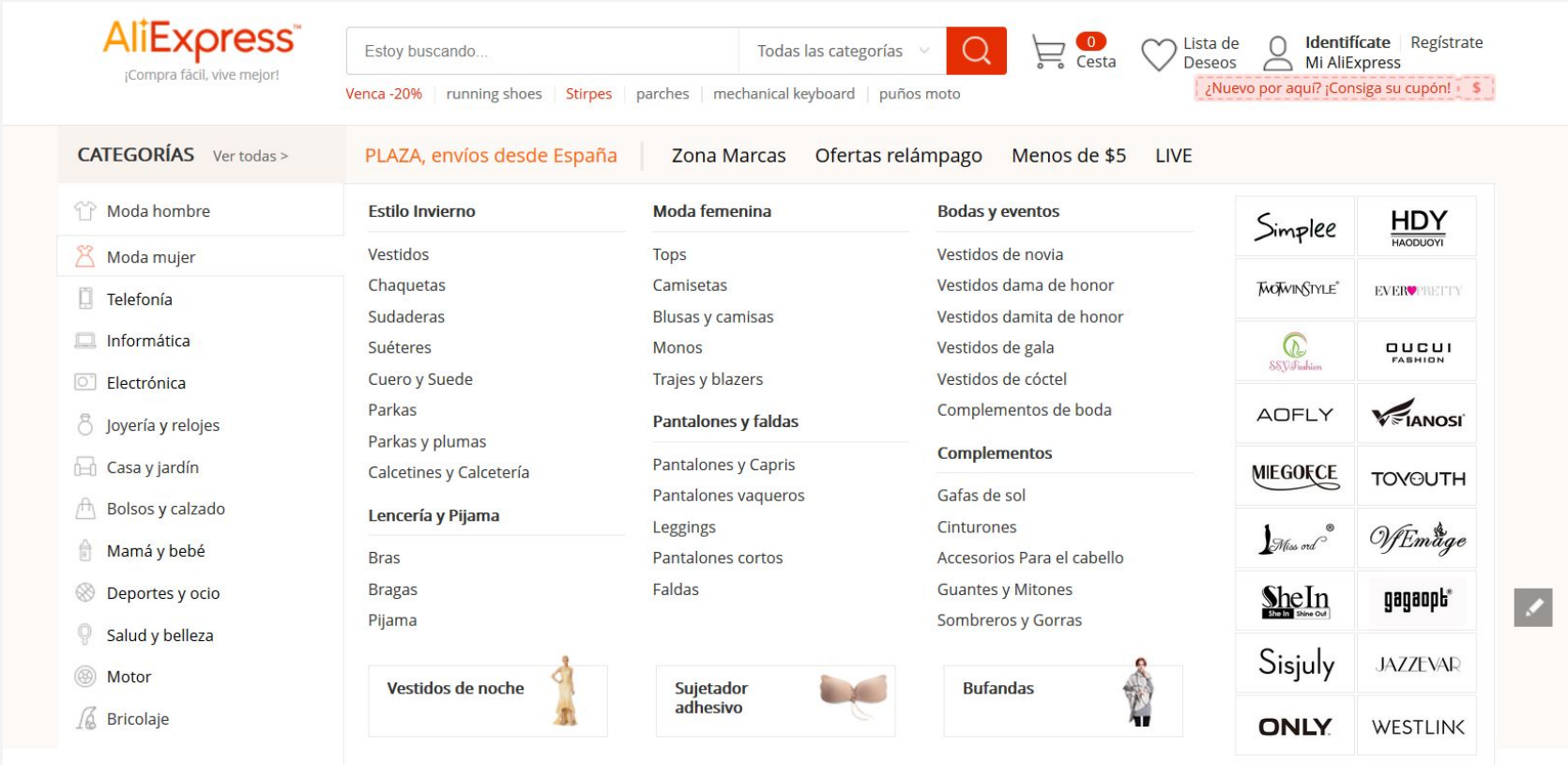
¿Es suficiente?


HE ESCAPADO CARACTERES, SOY INVENCIBLE
ANIMALITO

¿Qué problema puede haber entonces?

Imaginemos que tenemos una tienda en internet

Nos va bien y cada vez tenemos más productos y más categorías



32 queries diferentes sólo para acceder a una sección de mujer, hay 13 categorías por lo que podríamos estimar 416 accesos además del login

Mucho que filtrar

Hay que pensar que cualquier fallo en el filtrado puede desencadenar en una desgracia

Además, si utilizamos cookies u otros métodos donde su contenido termine en una query, hay que filtrar

Pero esto no acaba en el filtrado

En la vida real no trabajamos con una base de datos que hemos creado y tenemos acceso exclusivo, el fallo humano acecha en todos los rincones.

Para reducir los daños que nos pueden causar, tenemos que preocuparnos en administrar bien los privilegios

“¿Por qué debería preocuparme por la posteridad? ¿Qué ha hecho la posteridad por mí?

Groucho Marx


DEMO TIME