

uniR

UNIVERSIDAD
INTERNACIONAL
DE LA RIOJA

Seguridad en la Web

Práctica opcional

Proxy HTTP

La Universidad
en Internet

Javier Parra



Presentación

- Vamos a realizar una práctica utilizando varias herramientas para monitorizar la red y realizar secuestro de sesión.

- Para ello vamos a utilizar las herramientas:
 - Firefox Tamper Data
 - Jhijack

- Se va a poner en funcionamiento la práctica Blind String SQL Injection de WebGoat.

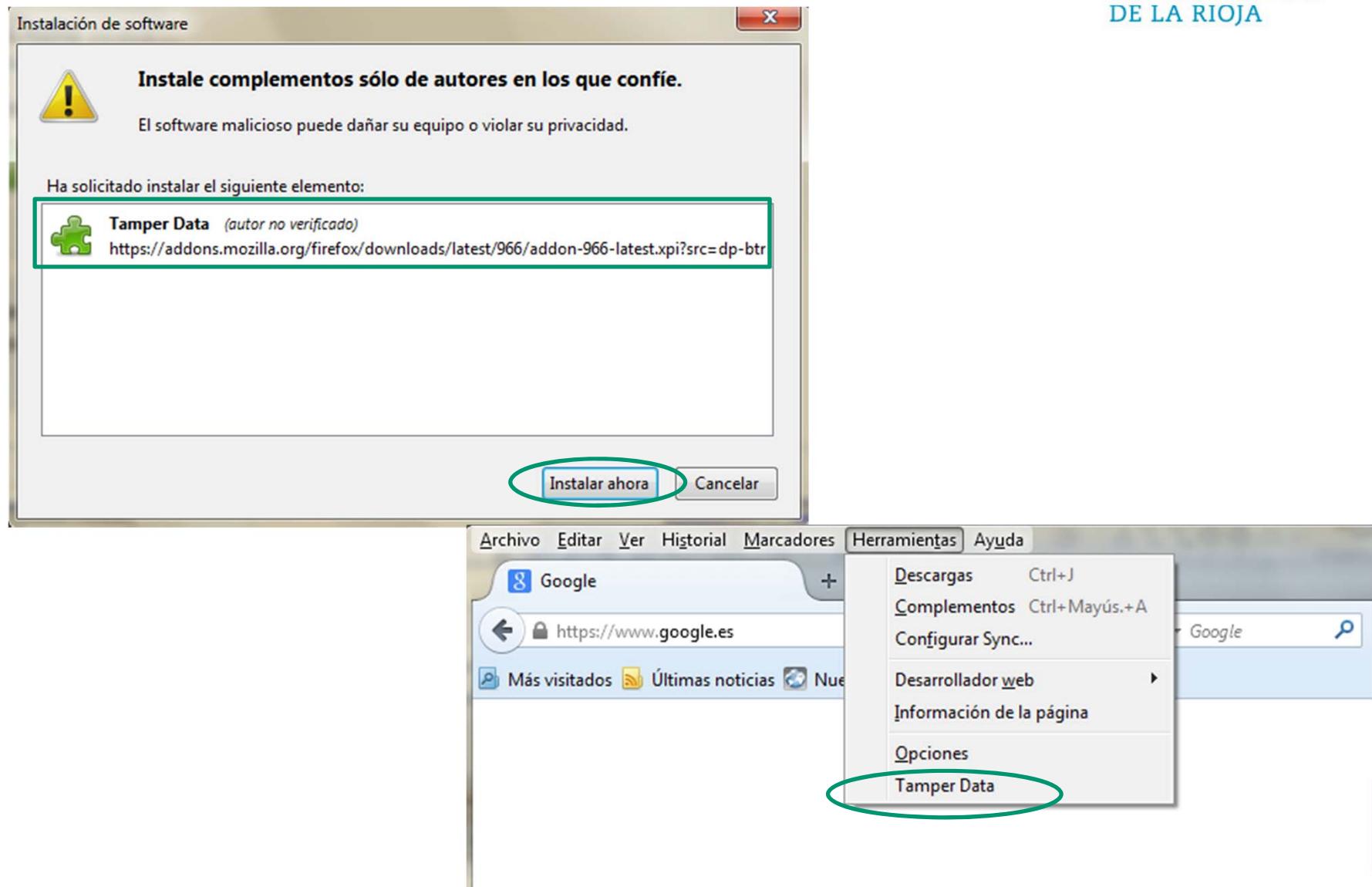
Firefox Tamper Data

Firefox Tamper Data - Descarga

- Extensión de Firefox que permite visualizar y modificar las cabeceras y los parámetros POST de las peticiones HTTP/HTTPS
- <https://addons.mozilla.org/es/firefox/addon/tamper-data/>

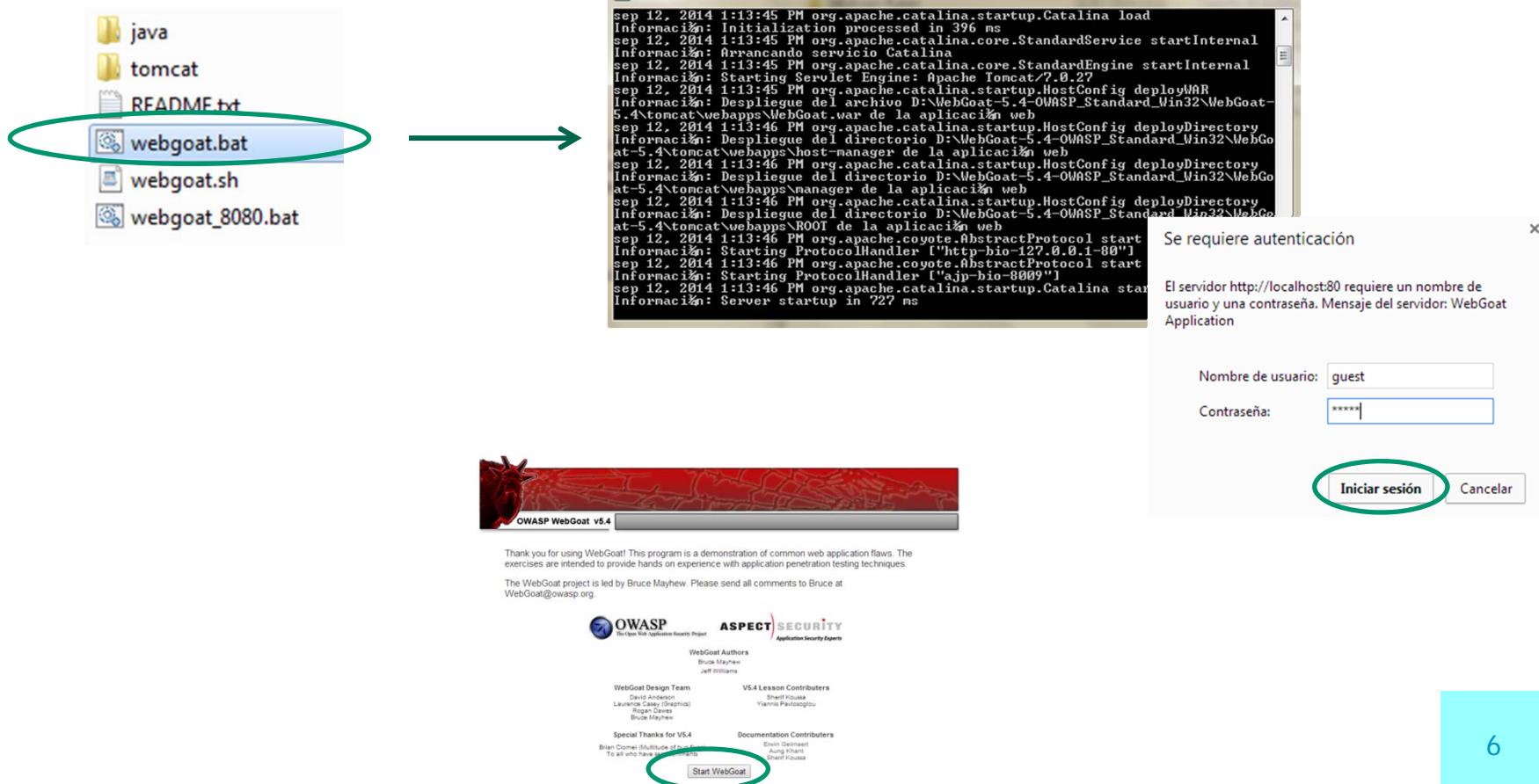
The screenshot shows the Mozilla Add-ons website interface. At the top, there's a navigation bar with links for 'Registrarse', 'Conectarse', 'Otras aplicaciones', and the 'mozilla' logo. Below the navigation is a search bar with the placeholder 'buscar complementos'. The main header says 'COMPLEMENTOS' with the Mozilla logo to its left. Underneath it, there are links for 'EXTENSIONES', 'TEMAS', 'COLECCIONES', and 'MÁS...'. The breadcrumb navigation shows the user is at 'Extensions > Tamper Data'. The main content area features the 'Tamper Data' extension by Adam Judson, version 11.0.1. It has a green puzzle piece icon. A description below the icon reads: 'Use tamperdata to view and modify HTTP/HTTPS headers and post parameters...'. A prominent green button with white text says '+ Agregar a Firefox', which is circled in green. To the right of the extension details, there's a rating section with five yellow stars and the text '117 valoraciones de los usuarios' and '127.327 usuarios'. At the bottom right of the extension card, there are links for 'Añadir a una colección' and 'Compartir este complemento'.

Firefox Tamper Data - Instalación



Firefox Tamper Data - Prueba

- Ejecutar aplicación (p.e. WebGoat)



Firefox Tamper Data - Prueba

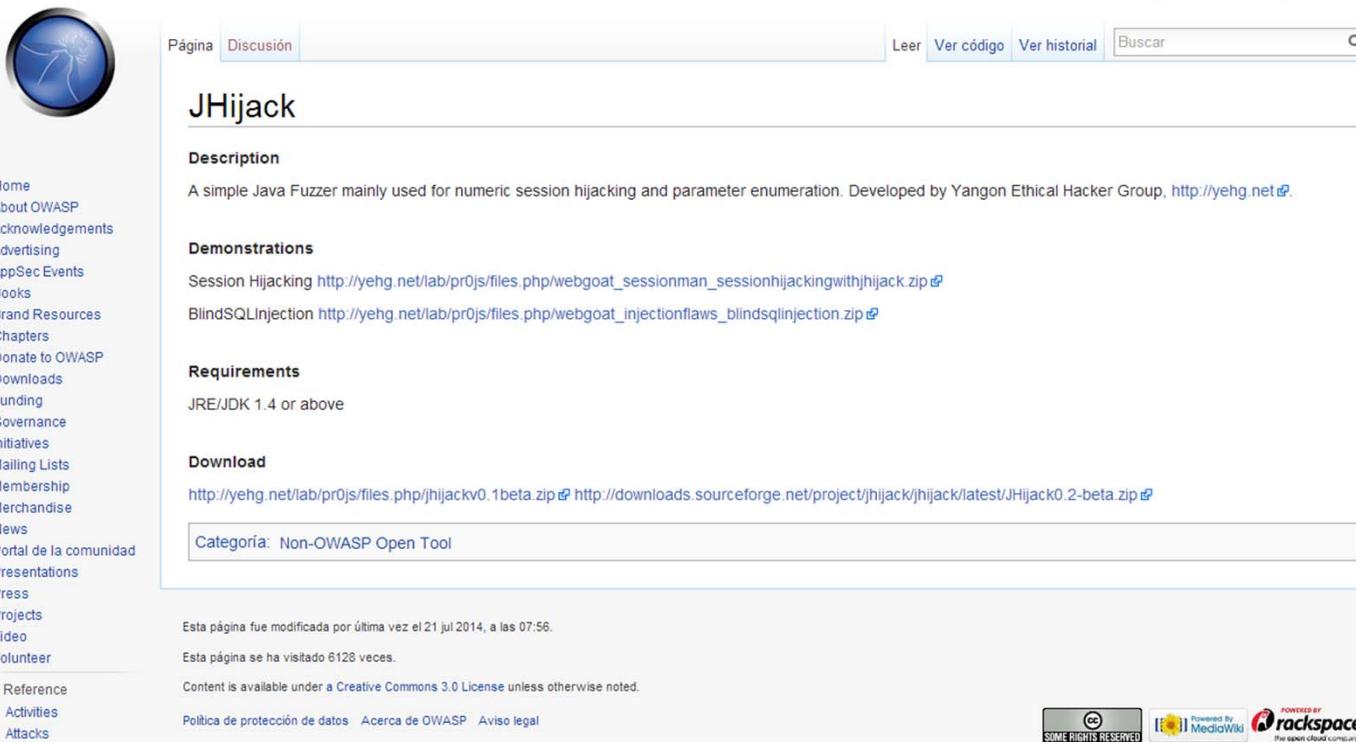
- Al interactuar con la aplicación, en Firefox Tamper Data aparecerá la información de las peticiones

The screenshot shows two windows side-by-side. On the left is the OWASP WebGoat v5.4 application. The main page displays a red background with a goat logo and the title "How to work with WebGoat". The navigation bar includes links for "Hints", "Show Params", "Show Cookies", "Lesson Plan", "Show Java", and "Solution". A sidebar on the left lists various security challenges, with "Blind String SQL Injection" circled in green at the bottom. The main content area shows a challenge titled "How To Work With WebGoat" with instructions and environment information. On the right is a Firefox browser window showing the "Tamper Data - Siguientes peticiones" extension. The extension interface has a header with "Comenzar modificación", "Parar modificación", and "Limpiar" buttons, and a "Opciones Ayuda" menu. Below is a table with columns: Hora, Duración, Duración total, Tamaño, Método, Estado, Tipo de contenido, URL, and Marcadores cargados. One row is highlighted in green, showing a GET request for "http://... LOAD_NORMAL" with a duration of 3 ms. At the bottom of the extension window, there are two tables for "Nombre de cabecera pedida" and "Nombre de cabecera recibida", both currently empty.

JHijack

JHijack – Sitio web

- Herramienta Java para simular la apropiación de sesiones
- <https://www.owasp.org/index.php/JHijack>

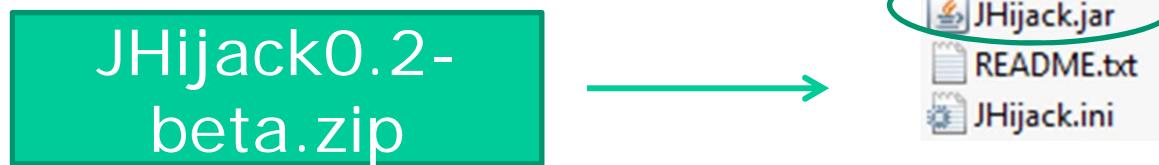


The screenshot shows a Wikipedia page for "JHijack". The page title is "JHijack". The content includes sections for "Description", "Demonstrations", "Requirements", and "Download". The "Description" section states: "A simple Java Fuzzer mainly used for numeric session hijacking and parameter enumeration. Developed by Yangon Ethical Hacker Group, <http://yehg.net>". The "Demonstrations" section links to "Session Hijacking" and "BlindSQLInjection". The "Requirements" section specifies "JRE/JDK 1.4 or above". The "Download" section provides links to "jHijackv0.1beta.zip" and "jHijack0.2-beta.zip". A sidebar on the left lists various OWASP links such as Home, About OWASP, Acknowledgements, Advertising, AppSec Events, Books, Brand Resources, Chapters, Donate to OWASP, Downloads, Funding, Governance, Initiatives, Mailing Lists, Membership, Merchandise, News, Portal de la comunidad, Presentations, Press, Projects, Video, Volunteer, Reference, Activities, and Attacks. At the bottom, there are footer links for "Política de protección de datos", "Acerca de OWASP", and "Aviso legal". Logos for Creative Commons, MediaWiki, and Rackspace are also present.

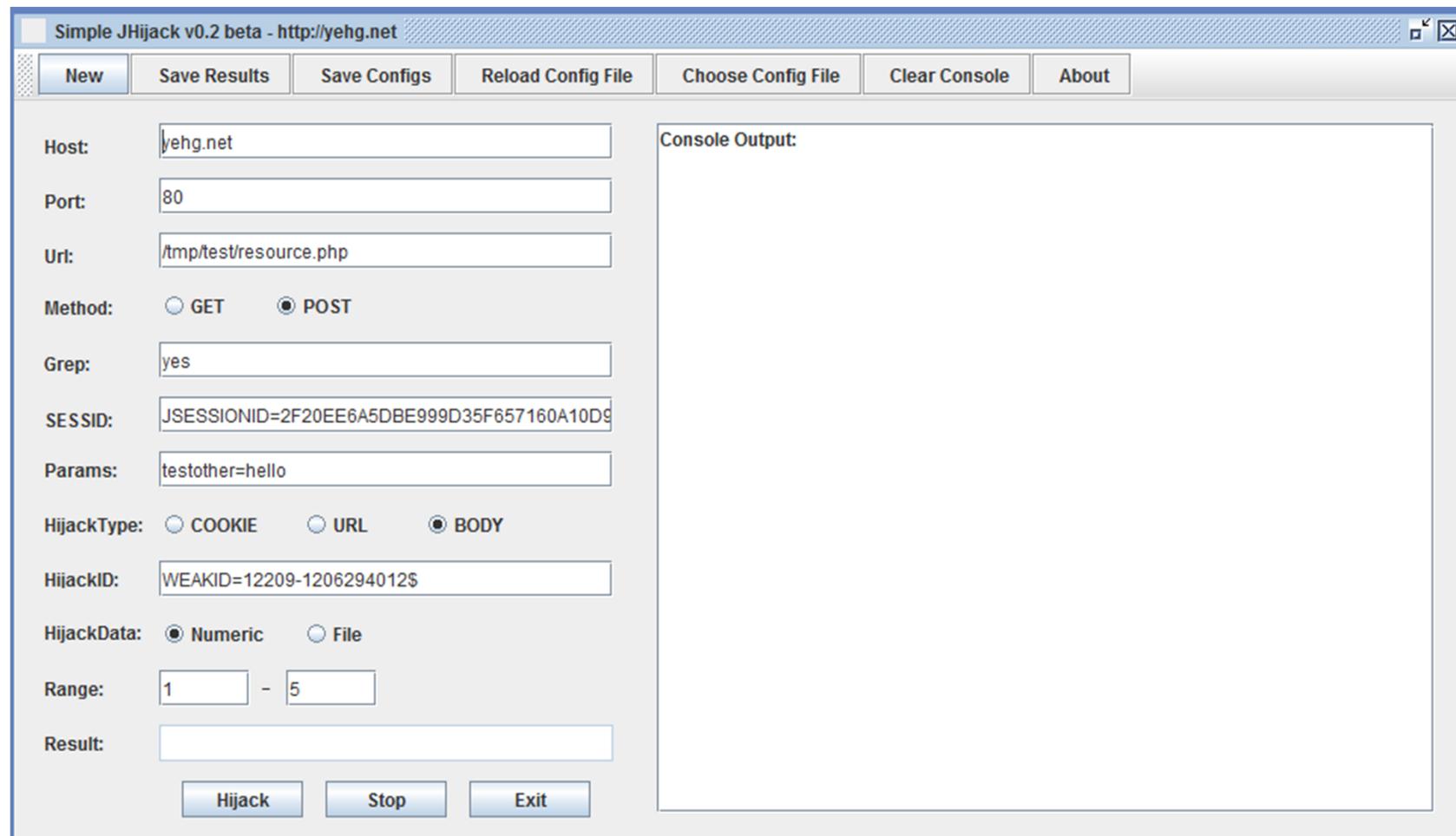
JHijack - Descarga

- Descarga:

<http://downloads.sourceforge.net/project/jhijack/jhijack/latest/JHijack0.2-beta.zip>



JHijack - Interfaz



Práctica WebGoat

Blind String SQL Injection

1. Abrir la lección en WebGoat con Firefox

- <http://localhost/WebGoat/attack>

The screenshot shows the OWASP WebGoat v5.4 interface. At the top, there's a banner with a red goat logo and the text "Internationalization is not available for this lesson". Below the banner, the title "Blind String SQL Injection" is displayed. On the left, a sidebar lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws (with sub-links: Command Injection, Numeric SQL Injection, Log Spoofing, XPATH Injection, String SQL Injection, LAB: SQL Injection, Stage 1: String SQL Injection, Stage 2: Parameterized Query #1, Stage 3: Numeric SQL Injection, Stage 4: Parameterized Query #2, Modify Data with SQL Injection, Add Data with SQL Injection, Database Backdoors, Blind Numeric SQL Injection, and Blind String SQL Injection). The "Blind String SQL Injection" link is circled in green. The main content area contains instructions for performing a blind SQL injection attack on a database table named "pins" where the "cc_number" is 432143214321. It shows a form where the user enters an account number (101) and a "Go!" button. Below the form, it says "Account number is valid". At the bottom, it credits "Created by Chuck Willis" and "MANDIANT INTELLIGENT INFORMATION SECURITY".

- Aplicación que devuelve si un número de cuenta es o no válido
- Se aprovechará las vulnerabilidades para obtener el nombre del titular de la cuenta
4321432143214321

2. Configuración

Configura JHijack para lanzar la consulta:

- Host: localhost
- Port: 80
- Url:

The image shows two windows side-by-side. On the left is the 'Simple JHijack v0.2 beta - http://yehg.net' application window. It has several input fields and buttons. The 'Host' field contains 'localhost', the 'Port' field contains '80', and the 'Url' field contains '/WebGoat/attack?Screen=165&menu=1100'. Arrows point from these three fields to their respective counterparts in the browser window on the right. The browser window shows a login page for 'OWASP WebGoat v5.4' with a red goat logo. The URL in the address bar is 'localhost/WebGoat/attack?Screen=165&menu=1100'. The page content includes links for 'Más visitados', 'Últimas noticias', 'Nuevo marcador', and 'Noticias'.

2. Configuración

Simple JHijack v0.2 beta - http://yehg.net

New Save Results Save Configs Reload Config File

Host: localhost
Port: 80
Url: /WebGoat/attack?Screen=165&menu=11
Method: GET POST
Grep: Account number is valid

SESSID:
Params:
HijackType: COOKIE URL BODY
HijackID:
HijackData: Numeric File
Range: -
Result:

Hijack Stop Exit

Logout

OWASP WebGoat v5.4

Blind String SQL Injection

< Hints > Show Params Show Cookies Lesson Plan Show Java Solution

Solution Videos

Restart this Lesson

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form below. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

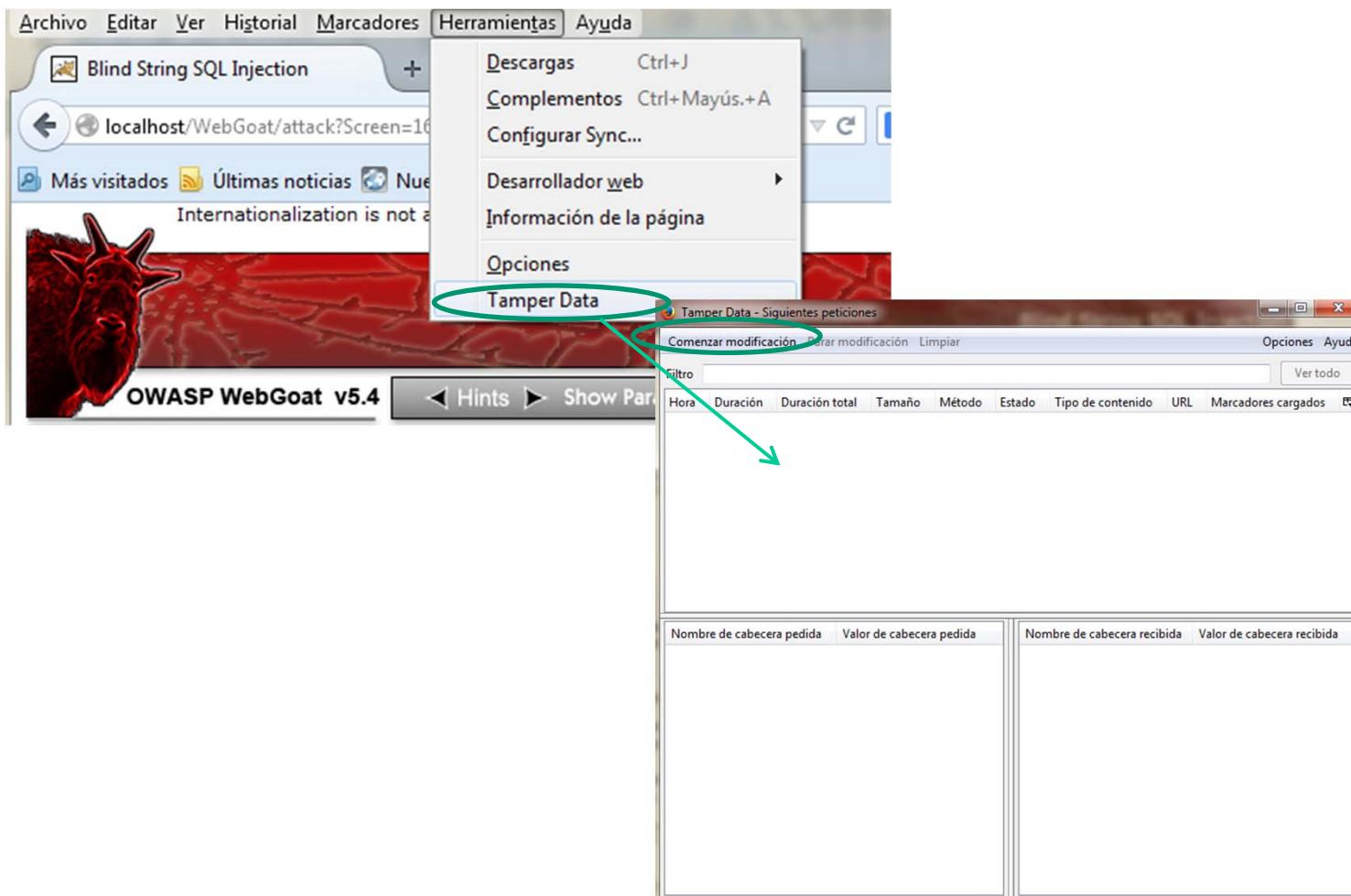
Enter your Account Number: Go!

Account number is valid

Created by Chuck Willis **MANDIANT**[®]
INTELLIGENT INFORMATION SECURITY

3. Obtención de la cookie

- Obtener cookie
 - Activar Firefox Tamper Data



3. Obtención de la cookie

Internationalization is not available for this lesson

Logout ?

OWASP WebGoat v5.4

Hints Show Params Show Cookies Lesson Plan Show Java Solution

Solution Videos

Restart this Lesson

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc_number** of **432143214321100**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number: 101

Account number is valid

Created by Chuck Willis **MANDIANT**
INTELLIGENT INFORMATION SECURITY

OWASP Foundation | Project WebGoat | Report Bug

Host: localhost

User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:29.0) Gecko/20100101

Accept: text/html,application/xhtml+xml,application/xml;q=0.9

Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3

Accept-Encoding: gzip, deflate

Referer: http://localhost/WebGoat/attack?Screen=165&menu=1100

Cookie: JSESSIONID=615A2E3162D1A93B8DDA7A67AD93BEB0

Authorization: Basic Z3Vlc3Q6Z3Vlc3Q=

Modificar petición?

http://localhost/WebGoat/attack?Screen=165&menu=1100

Continuar modificando?

Modificar Abortar petición

Nombre de parámetro p... Valor de parámetro...

account_number 101

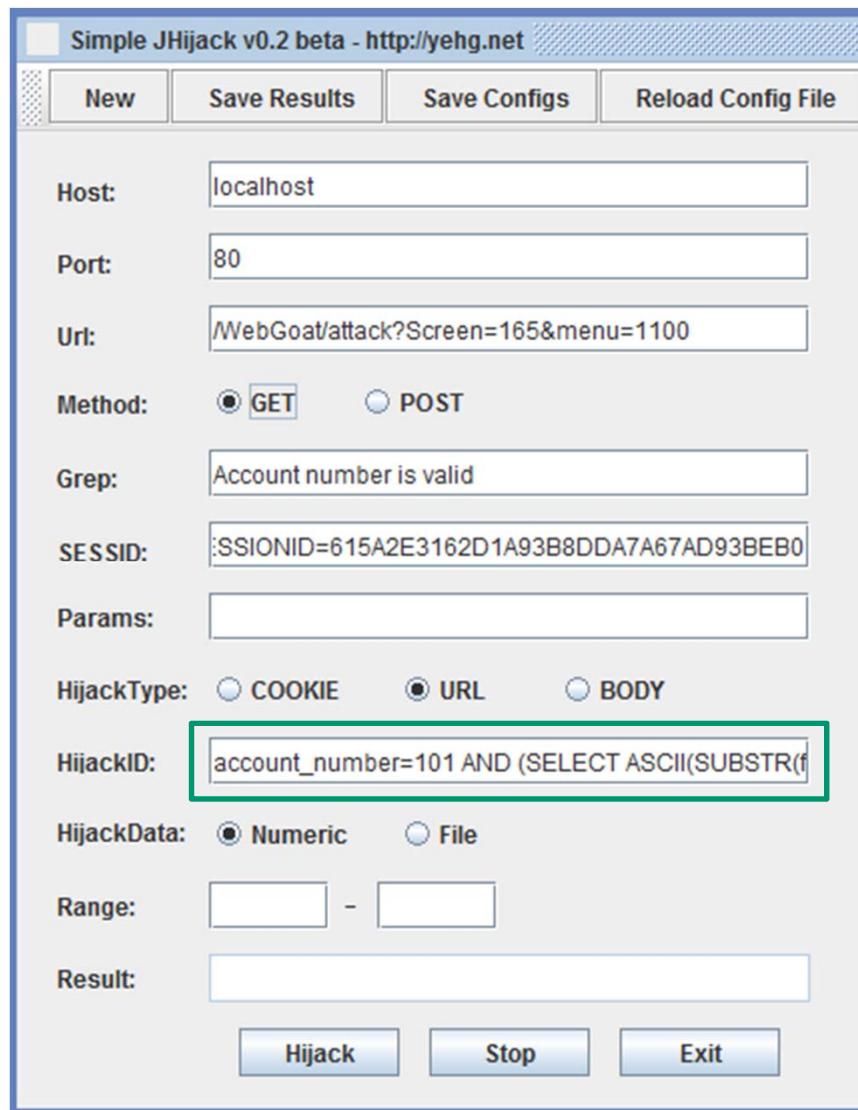
SUBMIT Go%21

Aceptar Cancelar

3. Obtención de la cookie

The image shows two windows side-by-side. On the left is the 'Simple JHijack v0.2 beta' interface. It has fields for Host (localhost), Port (80), Url (/WebGoat/attack?Screen=165&menu=1100), Method (GET), Grep (Account number is valid), SESSID (JSESSIONID=615A2E3162D1A93B8DDA7A67AD93BEB0), Params (empty), HijackType (URL), and HijackID (account_number=101). A green arrow points from the 'HijackID' field to the 'Cookie' field in the Tamper proxy window. On the right is the 'Ventana Tamper' window showing the request details. The 'Cookie' field contains the value JSESSIONID=615A2E3162D1A93B8DDA7A67AD93BEB0, which is highlighted with a green box. Another green arrow points from the 'HijackID' field in the JHijack window to this highlighted value. The Tamper window also shows other headers like Host, User-Agent, Accept, etc., and a parameter table with account_number: 101 and SUBMIT: Go%21.

4. Inyección de Código SQL



- Modificar la query para ir obteniendo las letras del nombre:
 - Actual: ...
account_number=101
 - Nuevo: ...
account_number=101 AND (SELECT ASCII(SUBSTR(name,1,1)) FROM pins WHERE cc_number=4321432143214321) =\\$ --

5. Rango de caracteres válidos

Simple JHijack v0.2 beta - http://yehg.net

New Save Results Save Configs Reload Config File

Host: localhost

Port: 80

Url: /WebGoat/attack?Screen=165&menu=1100

Method: GET POST

Grep: Account number is valid

SESSID: :SESSIONID=615A2E3162D1A93B8DDA7A67AD93BEB0

Params:

HijackType: COOKIE URL BODY

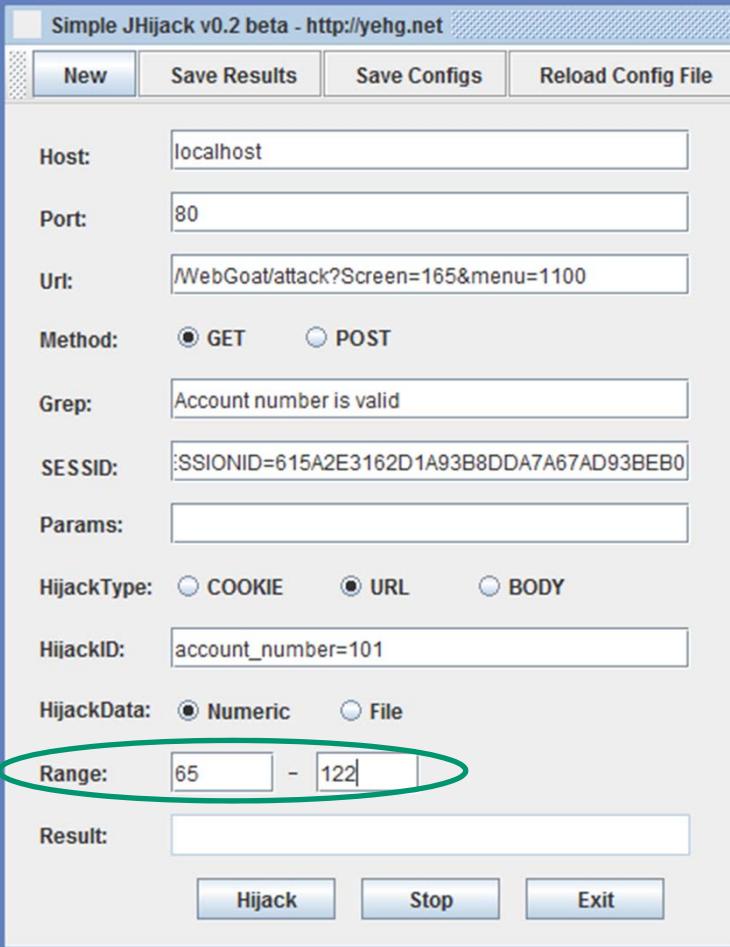
HijackID: account_number=101

HijackData: Numeric File

Range: 65 - 122

Result:

Hijack Stop Exit

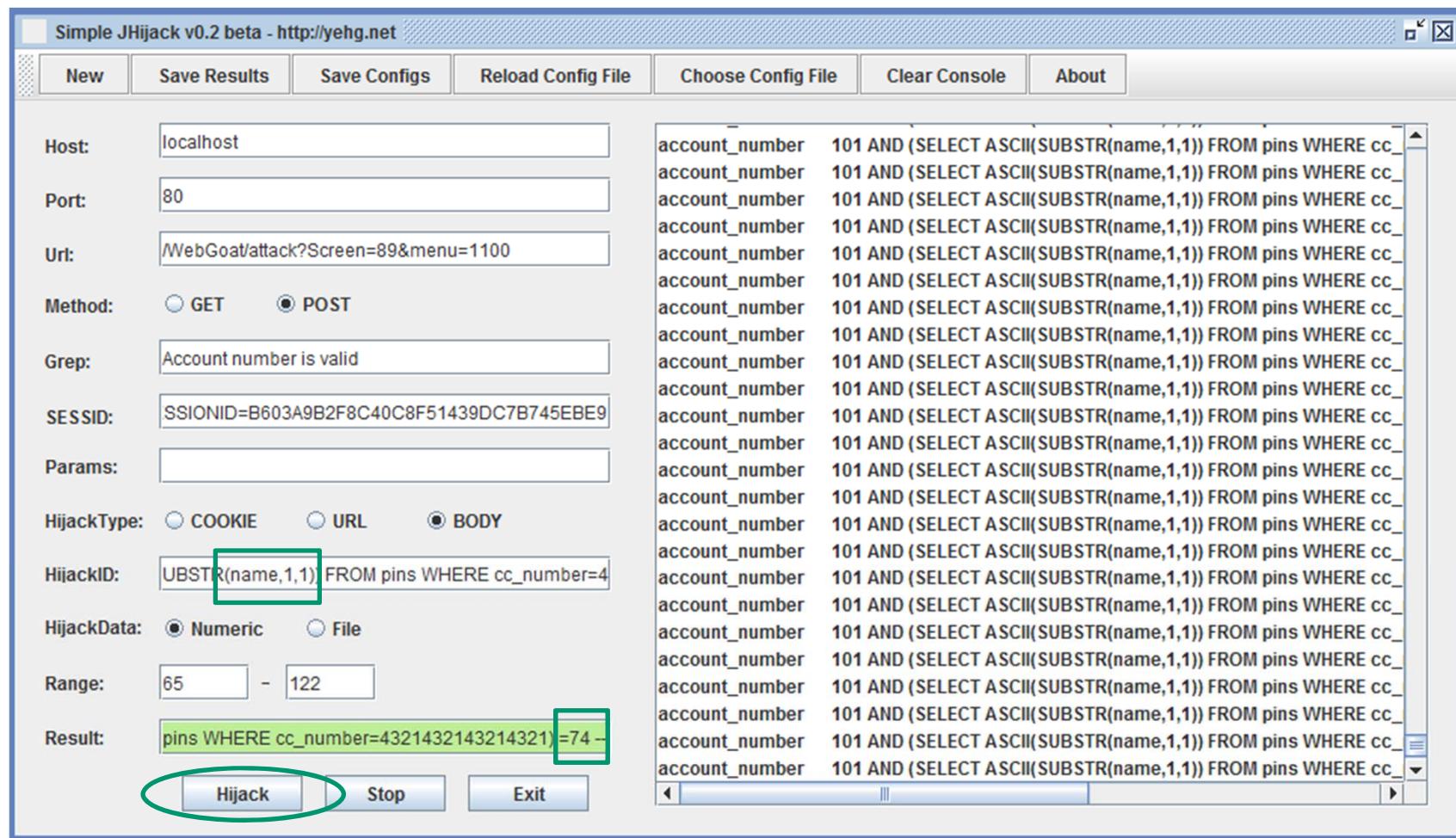


Codificación UTF-8

- A: 65
- ...
- Z: 90
- a: 97
- ...
- z: 122

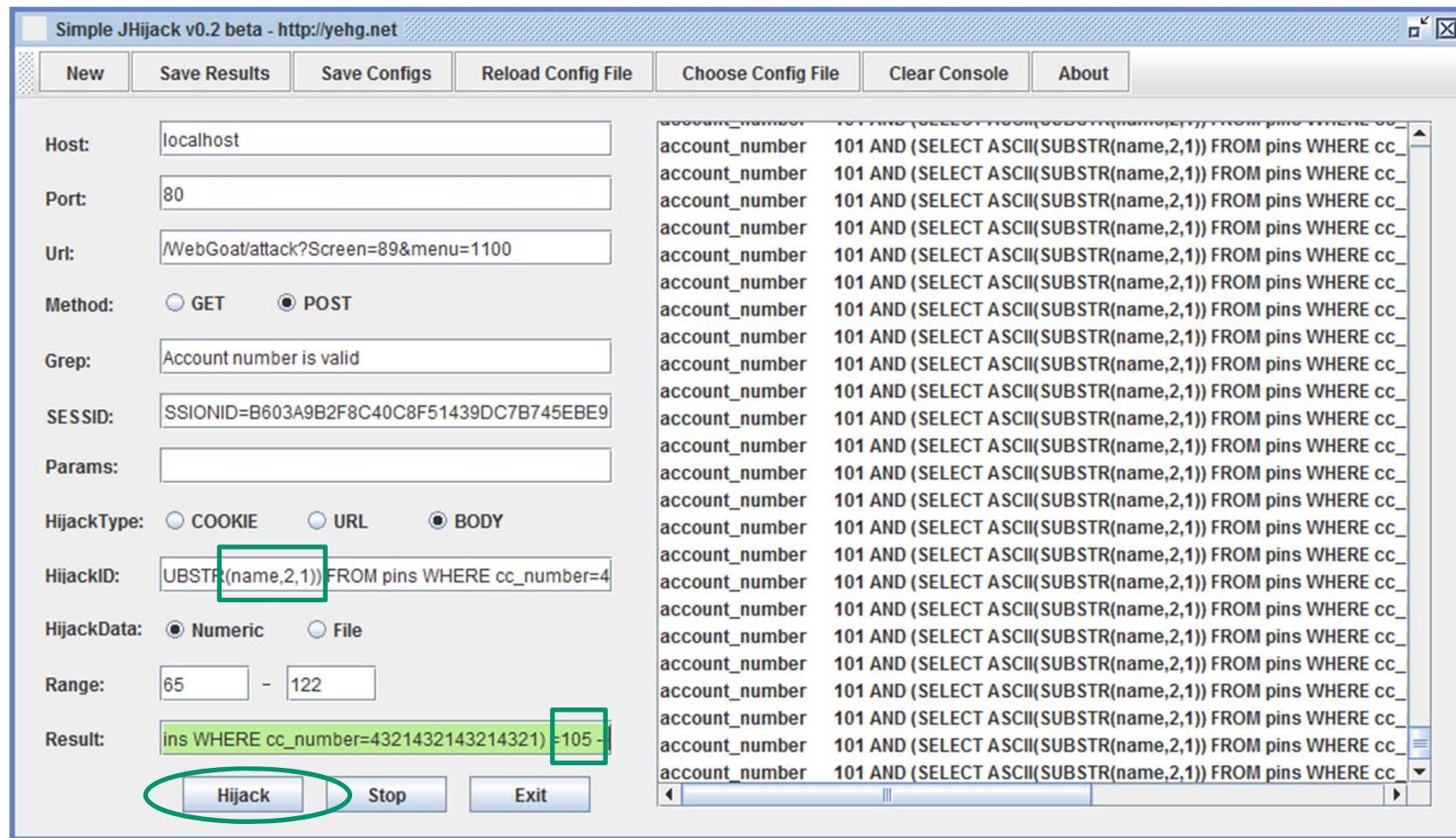
6. Lanzar ataque

- Paso 1: 1^a letra → 74



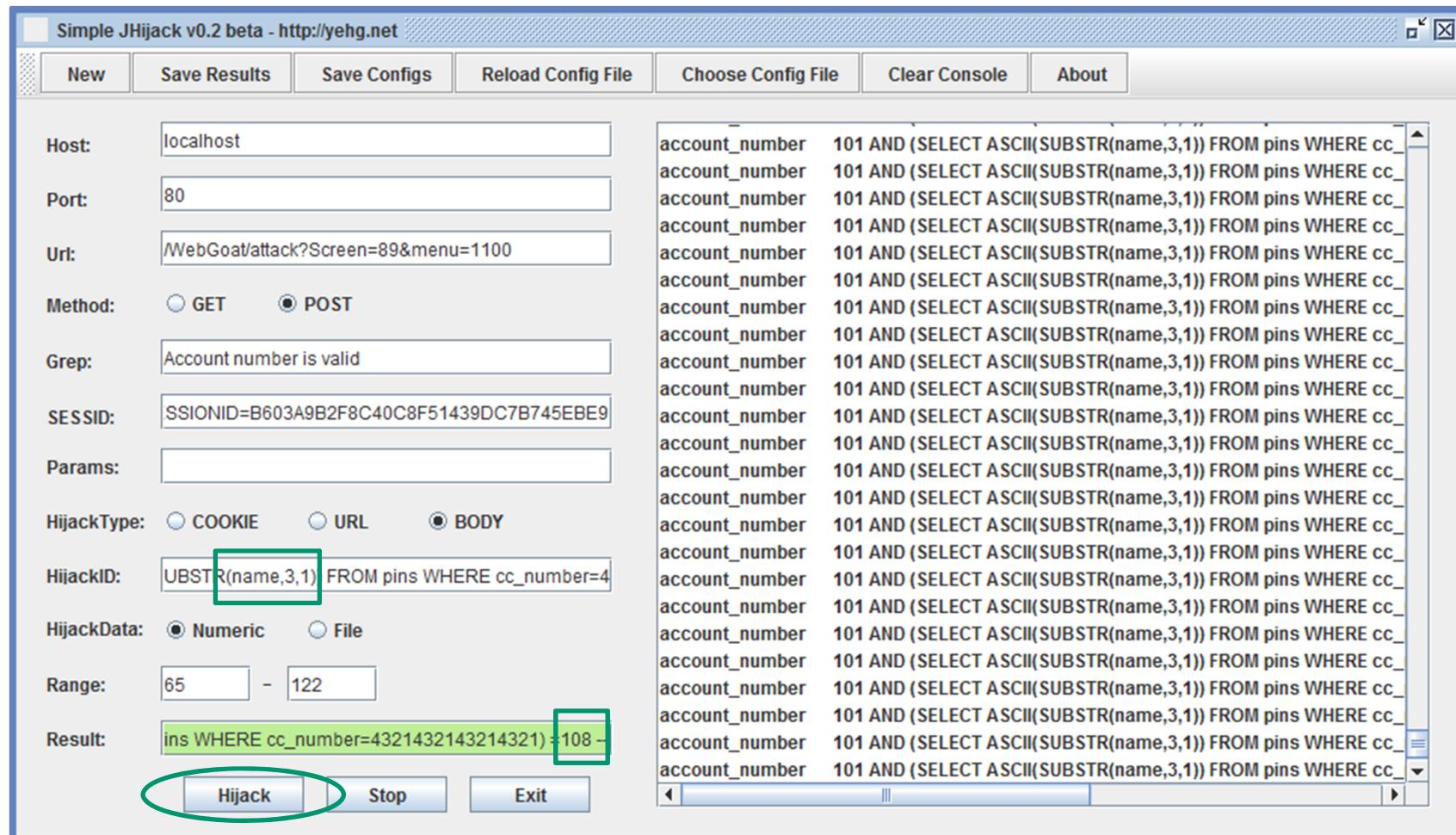
6. Lanzar ataque

- Paso 2: 2^a letra → 105



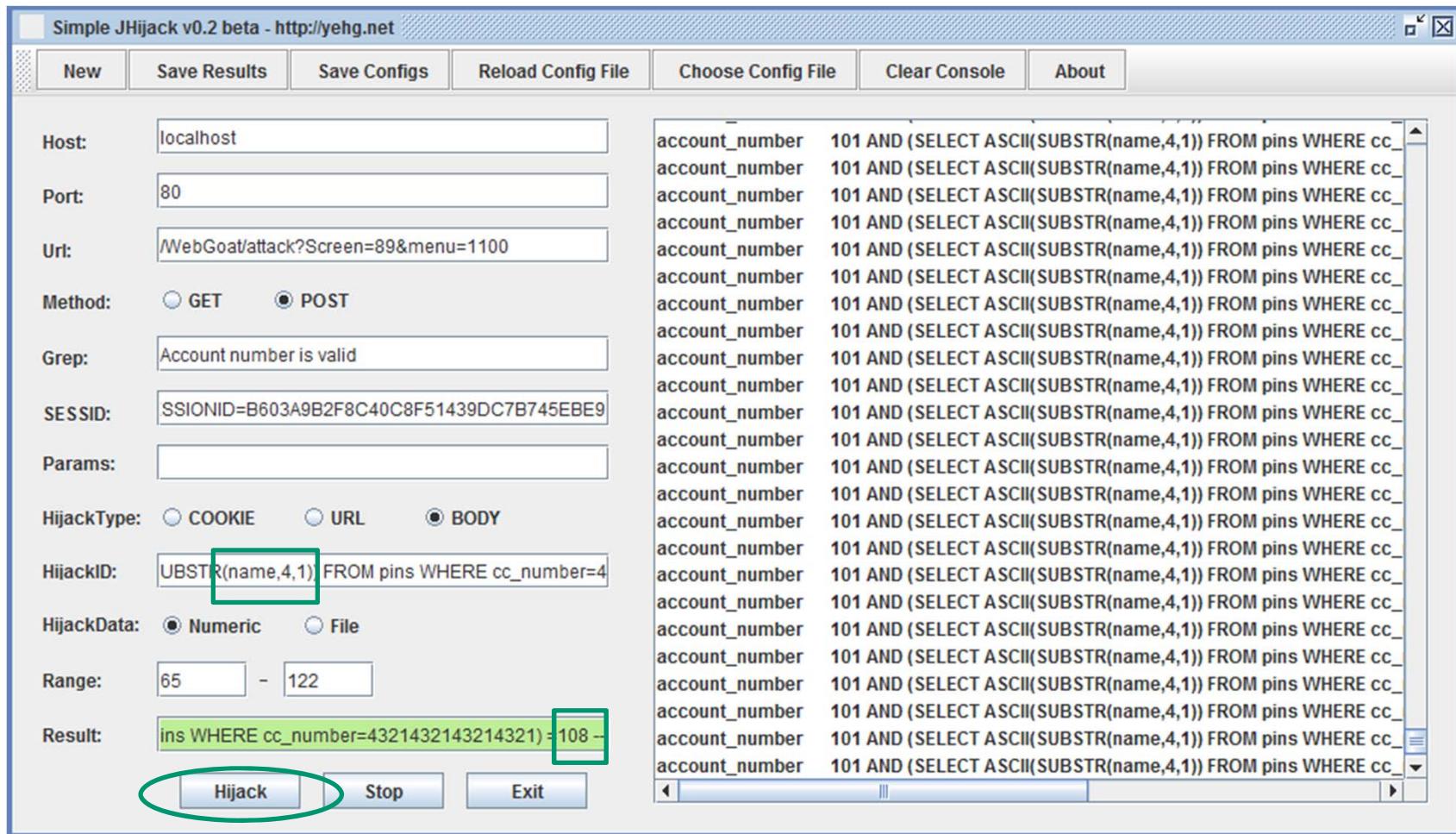
6. Lanzar ataque

- Paso 3: 3^a letra → 108



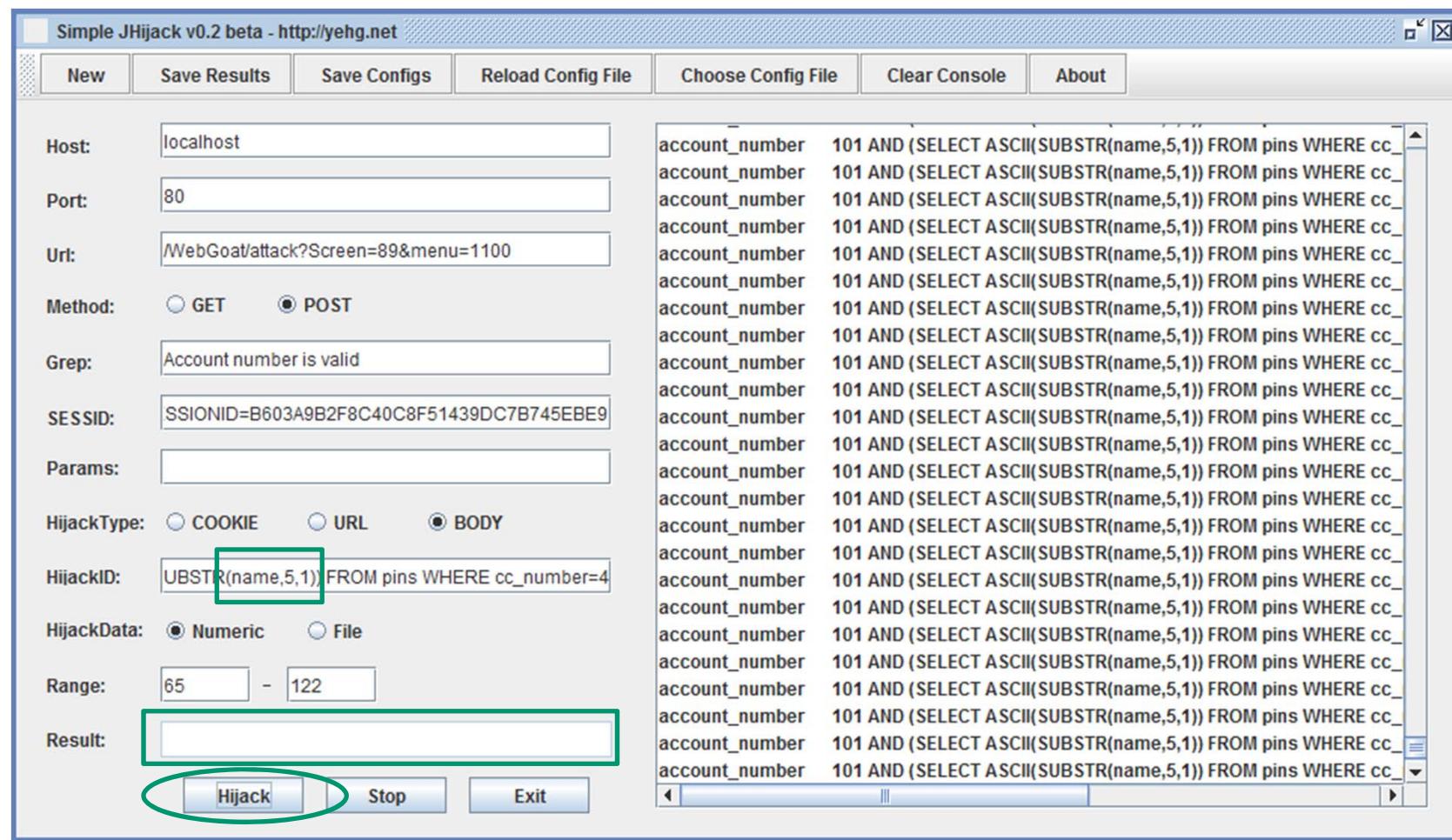
6. Lanzar ataque

- Paso 4: 4^a letra → 108



6. Lanzar ataque

- Paso 5: 5^a letra → No hay



7. Obtención de resultados

- Letras:

- 74: J
- 105: i
- 108: l
- 108: l

- Nombre: Jill

Caracteres ASCII imprimibles			
32	espacio	64	@
33	!	65	A
34	"	66	B
35	#	67	C
36	\$	68	D
37	%	69	E
38	&	70	F
39	'	71	G
40	(72	H
41)	73	I
42	*	74	J
43	+	75	K
44	,	76	L
45	-	77	M
46	.	78	N
47	/	79	O
48	0	80	P
49	1	81	Q
50	2	82	R
51	3	83	S
52	4	84	T
53	5	85	U
54	6	86	V
55	7	87	W
56	8	88	X
57	9	89	Y
58	:	90	Z
59	;	91	[
60	<	92	\
61	=	93]
62	>	94	^
63	?	95	_

7. Obtención de resultados

Internationalization is not available for this lesson

Logout ?

Blind String SQL Injection

OWASP WebGoat v5.4

< Hints ▶ Show Params Show Cookies Lesson Plan Show Java Solution

Introduction General Access Control Flaws AJAX Security Authentication Flaws Buffer Overflows Code Quality Concurrency Cross-Site Scripting (XSS) Improper Error Handling Injection Flaws

[Command Injection](#)
[Numeric SQL Injection](#)
[Log Spoofing](#)
[XPath Injection](#)
[String SQL Injection](#)
[LAB: SQL Injection](#)

[Stage 1: String SQL Injection](#)
[Stage 2: Parameterized Query #1](#)
[Stage 3: Numeric SQL Injection](#)
[Stage 4: Parameterized Query #2](#)

Modify Data with SQL Injection

Solution Videos

Restart this Lesson

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

* Congratulations. You have successfully completed this lesson.

Enter your Account Number: Go!

Created by Chuck Willis **MANDIANT®**
INTELLIGENT INFORMATION SECURITY

OWASP Foundation | Project WebGoat | Report Bug

uniR

UNIVERSIDAD
INTERNACIONAL
DE LA RIOJA

Seguridad en la Web

Práctica opcional

Proxy HTTP

La Universidad
en Internet

Javier Parra

