

How to scan web sites with Faraday IDE on Kali Linux

Introduction

The motivation for this paper is to show the user how to quickly get Kali Linux up and running, along with Damn Vulnerable Web Application (DVWA) to use the Faraday IDE in order to scan a web site [in this case, DVWA]. Faraday is a bad-ass tool to aggregate results from numerous other tools in one place and show the total findings of vulnerabilities. Basically: “Faraday introduces a new concept – IPE (Integrated Penetration-Test Environment) a multiuser Penetration test IDE. Designed for distribution, indexation and analysis of the data generated during a security audit. The main purpose of Faraday is to re-use the available tools in the community to take advantage of them in a multiuser way.” source:

<https://tools.kali.org/information-gathering/faraday>

Requirements

If you see the following \$ symbol on a command line to execute, what that means is that the command is executed as a regular user, i.e. the Ubuntu user. Ignore the leading \$ and execute the rest of the command.

```
$ command to execute as a regular user
```

If you see a command line lead with the # symbol, then that means that the command is executed as the root user. This implies you need to elevate to the root user before running the command, e.g. with: `sudo`

```
su - root.
```

```
# command to execute as the root user
```

VirtualBox

Go to: <https://www.virtualbox.org/wiki/Downloads> and download VirtualBox.

The author is running on Ubuntu 17.04, so following to this URL:

https://www.virtualbox.org/wiki/Linux_Downloads

For Ubuntu, double click on the .deb file, i.e. `virtualbox-5.2_5.2.0-118431-Ubuntu-zesty_amd64.deb`, and install VirtualBox on your local workstation.

Clean VirtualBox Networking

Run these two commands from a Terminal:

```
VBoxManage list natnetworks
```

```
VBoxManage list dhcpservers
```

Output:

```
NetworkName: 192.168.139-NAT
IP:          192.168.139.1
Network:     192.168.139.0/24
```

```

IPv6 Enabled:    No
IPv6 Prefix:    fd17:625c:f037:a88b::/64
DHCP Enabled:   Yes
Enabled:        Yes
loopback mappings (ipv4)
    127.0.0.1=2

NetworkName:    192.168.139-NAT
IP:             192.168.139.3
NetworkMask:    255.255.255.0
lowerIPAddress: 192.168.139.101
upperIPAddress: 192.168.139.254
Enabled:        Yes

NetworkName:    HostInterfaceNetworking-vboxnet0
IP:             172.20.0.3
NetworkMask:    255.255.255.0
lowerIPAddress: 172.20.0.101
upperIPAddress: 172.20.0.254
Enabled:        Yes

NetworkName:    HostInterfaceNetworking-vboxnet1
IP:             0.0.0.0
NetworkMask:    0.0.0.0
lowerIPAddress: 0.0.0.0
upperIPAddress: 0.0.0.0
Enabled:        No

```

Now, delete ALL of the pre-installed VirtualBox networks (one at a time following the syntax below):

```

VBoxManage natnetwork remove --netname <NetworkName_from_above>
VBoxManage natnetwork remove --netname 192.168.139-NAT
# repeat as many times as necessary to delete all of them.

```

```

VBoxManage dhcpserver remove --netname <DHCP_Server_NetworkName_from_above>
VBoxManage dhcpserver remove --netname 192.168.139-NAT
# repeat as many times as necessary to delete all of them.

```

Add VirtualBox Networking

Now, add the new VirtualBox networks so the Kali Linux guides work.

```

VBoxManage natnetwork add \
  --netname 192.168.139-NAT \
  --network "192.168.139.0/24" \
  --enable --dhcp on

```

```

VBoxManage dhcpserver add \
  --netname 192.168.139-NAT \
  --ip 192.168.139.3 \
  --lowerip 192.168.139.101 \
  --upperip 192.168.139.254 \
  --netmask 255.255.255.0 \
  --enable

```

```

VBoxManage hostonlyif create

```

```

VBoxManage hostonlyif ipconfig vboxnet0 \
  --ip 172.20.0.1 \
  --netmask 255.255.255.0

```

```

VBoxManage dhcpserver add \
  --ifname vboxnet0 \
  --ip 172.20.0.3 \
  --lowerip 172.20.0.101 \
  --upperip 172.20.0.254 \
  --netmask 255.255.255.0

```

```
VBoxManage dhcpserver modify \  
--ifname vboxnet0 \  
--enable
```

Vagrant

Go to: <https://www.vagrantup.com/downloads.html>, follow the appropriate link to your OS and 32 or 64 bit version representing your local workstation. Download.

For Ubuntu, double click on the .deb file, i.e. vagrant_2.0.1_x86_64.deb, and install Vagrant on your local system.

Kali Linux

The author highly recommends to create a directory structure that is easy to navigate and find your code. As an example, you could use something similar to:

```
$(HOME)/Source_Code/Education/vagrant-machines/kali-linux-vm/
```

Go ahead and make this structure with the following command (inside a Terminal):

```
$ mkdir -p $(HOME)/Source_Code/Education/vagrant-machines/kali-linux-vm/
```

Inside of the kali-linux-vm directory, populate a new file with the exact name, "Vagrantfile". Case matters, uppercase the "V".

Vagrantfile:

```
# -*- mode: ruby -*-  
# vi: set ft=ruby :  
  
# Vagrantfile API/syntax version.  
VAGRANTFILE_API_VERSION = "2"  
  
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|  
  config.vm.box = "Sliim/kali-2017.2-amd64"  
  config.vm.box_version = "1"  
  
  # For Linux systems with the Wireless network, uncomment the line:  
  config.vm.network "public_network", bridge: "wlo1", auto_config: true  
  
  # For macbook/OSx systems, uncomment the line:  
  #config.vm.network "public_network", bridge: "en0: Wi-Fi (AirPort)", auto_config: true  
  
  config.vm.hostname = "kali-linux-vagrant"  
  
  config.vm.provider "virtualbox" do |vb|  
    vb.memory = "4096"  
    vb.cpus = "3"  
    vb.gui = true  
    vb.customize ["modifyvm", :id, "--cpuexecutioncap", "95"]  
    vb.customize ["modifyvm", :id, "--vram", "32"]  
    vb.customize ["modifyvm", :id, "--accelerate3d", "on"]  
    vb.customize ["modifyvm", :id, "--ostype", "Debian_64"]  
    vb.customize ["modifyvm", :id, "--boot1", "dvd"]  
    vb.customize ["modifyvm", :id, "--boot2", "disk"]  
    vb.customize ["modifyvm", :id, "--audio", "none"]  
    vb.customize ["modifyvm", :id, "--clipboard", "hosttoguest"]  
  end  
end
```

```
vb.customize ["modifyvm", :id, "--draganddrop", "hosttoguest"]
vb.customize ["modifyvm", :id, "--paravirtprovider", "kvm"]
end
end
```

Save and write this file.

From a Terminal, change directory to:

```
$ cd ${HOME}/Source_Code/Education/vagrant-machines/kali-linux-vm/
```

Then run (inside the directory kali-linux-vm):

```
$ vagrant up
```

This will download the appropriate image and start the virtual machine.

Once running, through the VirtuaBox GUI, login as root. Password is “toor”, root backwards. Edit the following file:

```
/etc/ssh/sshd_config
```

And change the line:

```
#PermitRootLogin prothibit-password
```

To:

```
PermitRootLogin yes
```

Then restart the ssh daemon:

```
# kill -HUP $(pgrep sshd)
```

Notice, you are on a Bridged adapter, this will open the instance to allow root to ssh in with the most unsecure password in the world. Only make this change (allowing root to login via SSH) if you require root SSH access. You can change the root user’s password, which is highly recommended.

Damn Vulnerable Web Application (DVWA)

Go ahead and make this structure with the following command (inside a Terminal):

```
$ mkdir -p ${HOME}/Source_Code/Education/vagrant-machines/dvwa-linux-vm/
```

Inside of the dvwa-linux-vm directory, populate a new file with the exact name, “Vagrantfile”. Case matters, uppercase the “V”.

Vagrantfile:

```
#
# setup local instance of Damn Vulnerable Web Application (DVWA):
#
# Vagrantfile API/syntax version. Don't touch unless you know what you're doing!
VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|

  # For Linux systems with the Wireless network, uncomment the line:
  config.vm.network "public_network", bridge: "wlo1", auto_config: true
```

```

# For macbook/OSx systems, uncomment the line:
#config.vm.network "public_network", bridge: "en0: Wi-Fi (AirPort)", auto_config: true

# uncomment the next line for Macbook/OSx systems, wireless :
# config.vm.network "public_network", bridge: "en0: Wi-Fi (AirPort)", auto_config: true

config.vm.provision :shell, path: "bootstrap.sh"
config.vm.hostname = "dvwa"

config.vm.provider "virtualbox" do |vb|
  vb.memory = "1024"
  vb.cpus = "2"
  vb.gui = false
  vb.customize ["modifyvm", :id, "--cpuexecutioncap", "95"]
  vb.customize ["modifyvm", :id, "--vram", "32"]
  vb.customize ["modifyvm", :id, "--accelerate3d", "on"]
  vb.customize ["modifyvm", :id, "--ostype", "Ubuntu_64"]
  vb.customize ["modifyvm", :id, "--boot1", "dvd"]
  vb.customize ["modifyvm", :id, "--boot2", "disk"]
  vb.customize ["modifyvm", :id, "--audio", "none"]
  vb.customize ["modifyvm", :id, "--clipboard", "hosttoguest"]
  vb.customize ["modifyvm", :id, "--draganddrop", "hosttoguest"]
  vb.customize ["modifyvm", :id, "--paravirtprovider", "kvm"]
end
end

```

Save and write this file.

Inside of the dvwa-linux-vm directory, populate a new file with the exact name, "bootstrap.sh". Case matters, all lowercase.

bootstrap.sh (include the shebang in your file, the `#!/usr/bin/env bash`):

```

#!/usr/bin/env bash
PHP_FPM_PATH_INI='/etc/php/7.0/fpm/php.ini'
PHP_FPM_POOL_CONF='/etc/php/7.0/fpm/pool.d/www.conf'
MYSQL_ROOT_PW='Assword12345'
MYSQL_dvwa_user='dvwa_root'
MYSQL_dvwa_password='sunshine'
DVWA_admin_password='admin'
recaptcha_public_key='u8392ihj32k18hujalkshuil32'
recaptcha_private_key='89ry8932873832lih32ilj32'

install_base() {
  add-apt-repository -y ppa:nginx/stable
  sudo apt-get update
  sudo apt-get dist-upgrade -y
  sudo apt-get install -y nginx mariadb-server mariadb-client php php-common php-cgi php-fpm
  php-gd php-cli php-pear php-mcrypt php-mysql php-gd git vim
}

config_mysql(){
  mysqladmin -u root password "${MYSQL_ROOT_PW}"
  # Config the mysql config file for root so it doesn't prompt for password.
  # Also sets pw in plain text for easy access.
  # Don't forget to change the password here!!
}

cat <<EOF > /root/.my.cnf
[client]
user="root"
password="${MYSQL_ROOT_PW}"
EOF
mysql -BNe "drop database if exists dvwa;"
mysql -BNe "CREATE DATABASE dvwa;"

```

```

mysql -Bne "GRANT ALL ON *.* TO '"${MYSQL_dwva_user}"'@'localhost' IDENTIFIED BY
'"${MYSQL_dwva_password}";"

service mysql restart
}

config_php(){
  ##Config PHP FPM INI to disable some security settings

  sed -i 's/^;cgi.fix_pathinfo.*$/cgi.fix_pathinfo = 0/g' ${PHP_FPM_PATH_INI}
  sed -i 's/allow_url_include = Off/allow_url_include = On/g' ${PHP_FPM_PATH_INI}
  sed -i 's/allow_url_fopen = Off/allow_url_fopen = On/g' ${PHP_FPM_PATH_INI}
  sed -i 's/safe_mode = On/safe_mode = Off/g' ${PHP_FPM_PATH_INI}
  echo "magic_quotes_gpc = Off" >> ${PHP_FPM_PATH_INI}
  sed -i 's/display_errors = Off/display_errors = On/g' ${PHP_FPM_PATH_INI}

  ##explicitly set pool options (these are defaults in ubuntu 16.04 so i'm commenting them out.
  If they are not defaults for you try uncommenting these
  #sed -i 's/^;security.limit_extensions.*$/security.limit_extensions
= .php .php3 .php4 .php5 .php7/g' /etc/php/7.0/fpm/pool.d/www.conf
  #sed -i 's/^listen.owner.*$/listen.owner = www-data/g' /etc/php/7.0/fpm/pool.d/www.conf
  #sed -i 's/^listen.group.*$/listen.group = www-data/g' /etc/php/7.0/fpm/pool.d/www.conf
  #sed -i 's/^;listen.mode.*$/listen.mode = 0660/g' /etc/php/7.0/fpm/pool.d/www.conf

  systemctl restart php7.0-fpm
}

config_nginx(){
cat << 'EOF' > /etc/nginx/sites-enabled/default
server
{
  listen 80;
  root /var/www/html;
  index index.php index.html index.htm;
  #server_name localhost
  location "/"
  {
    index index.php index.html index.htm;
    #try_files $uri $uri/ =404;
  }

  location ~ /\.php$
  {
    include /etc/nginx/fastcgi_params;
    fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $request_filename;
  }
}
EOF

systemctl restart nginx
}

install_dwva(){
  if [[ ! -d "/var/www/html" ]];
  then
    mkdir -p /var/www;
    ln -s /usr/share/nginx/html /var/www/html;
    chown -R www-data. /var/www/html;
  fi

  cd /var/www/html
  rm -rf /var/www/html/.[*]
  rm -rf /var/www/html/*
  git clone https://github.com/ethicalhack3r/DVWA.git ./

```

```

chown -R www-data. ./
cp config/config.inc.php.dist config/config.inc.php

### chmod uploads and log file to be writable by nobody
chmod 777 ./hackable/uploads/
chmod 777 ./external/phpids/0.6/lib/IDS/tmp/phpids_log.txt

## change the values in the config to match our setup (these are what you need to update!
sed -i '/db_user/ s/root/'${MYSQL_dvwa_user}'/' /var/www/html/config/config.inc.php
sed -i '/db_password/ s/p@ssw0rd/'${MYSQL_dvwa_password}'/'
/var/www/html/config/config.inc.php
sed -i "/recaptcha_public_key/ s/'/'${recaptcha_public_key}'"/'
/var/www/html/config/config.inc.php
sed -i "/recaptcha_private_key/ s/'/'${recaptcha_private_key}'"/'
/var/www/html/config/config.inc.php
}

update_mysql_user_pws(){
## The mysql passwords are set via /usr/share/nginx/html/dvwa/includes/DBMS/MySQL.php.
# If you edit this every time they are reset it will reset to those.
# Otherwise you can do a sql update statement to update them all (they are just md5's of the
string.
# The issue is the users table doesn't get created until you click that button T_T to init.

#mysql -BNe "UPDATE dvwa.users SET password = md5('YOUR_MYSQL_PW_HERE') WHERE user = 'admin';"
#mysql -BNe "UPDATE dvwa.users SET password = md5('YOUR_MYSQL_PW_HERE') WHERE user = 'gordonb';"
#mysql -BNe "UPDATE dvwa.users SET password = md5('YOUR_MYSQL_PW_HERE') WHERE user = '1337';"
#mysql -BNe "UPDATE dvwa.users SET password = md5('YOUR_MYSQL_PW_HERE') WHERE user = 'pablo';"
#mysql -BNe "UPDATE dvwa.users SET password = md5('YOUR_MYSQL_PW_HERE') WHERE user = 'smithy';"

sed -i '/admin/ s/password/'${DVWA_admin_password}'/g'
/var/www/html/dvwa/includes/DBMS/MySQL.php
sed -i '/gordonb/ s/abc123/'${DVWA_admin_password}'/g'
/var/www/html/dvwa/includes/DBMS/MySQL.php
sed -i '/1337/ s/charley/'${DVWA_admin_password}'/g' /var/www/html/dvwa/includes/DBMS/MySQL.php
sed -i '/pablo/ s/letmein/'${DVWA_admin_password}'/g' /var/www/html/dvwa/includes/DBMS/MySQL.php
sed -i '/smithy/ s/password/'${DVWA_admin_password}'/g'
/var/www/html/dvwa/includes/DBMS/MySQL.php
}

install_base
config_mysql
install_dvwa
update_mysql_user_pws
config_php
config_nginx
Save and write this file.

```

From a Terminal, change directory to:

```
$ cd ${HOME}/Source_Code/Education/vagrant-machines/dvwa-linux-vm/
```

Then run (inside the directory dvwa-linux-vm):

```
$ vagrant up
```

You will need the IP address from the new DVWA virtual machine.

Login with:

```
$ vagrant ssh
```

Then run:

```
$ ip a
```

Choose the second network adapter, it should look like:

```
ubuntu@dvwa:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 02:53:17:3c:de:80 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::53:17ff:fe3c:de80/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 08:00:27:f0:77:2d brd ff:ff:ff:ff:ff:ff
    inet 172.20.156.76/24 brd 172.20.156.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fef0:772d/64 scope link
        valid_lft forever preferred_lft forever
```

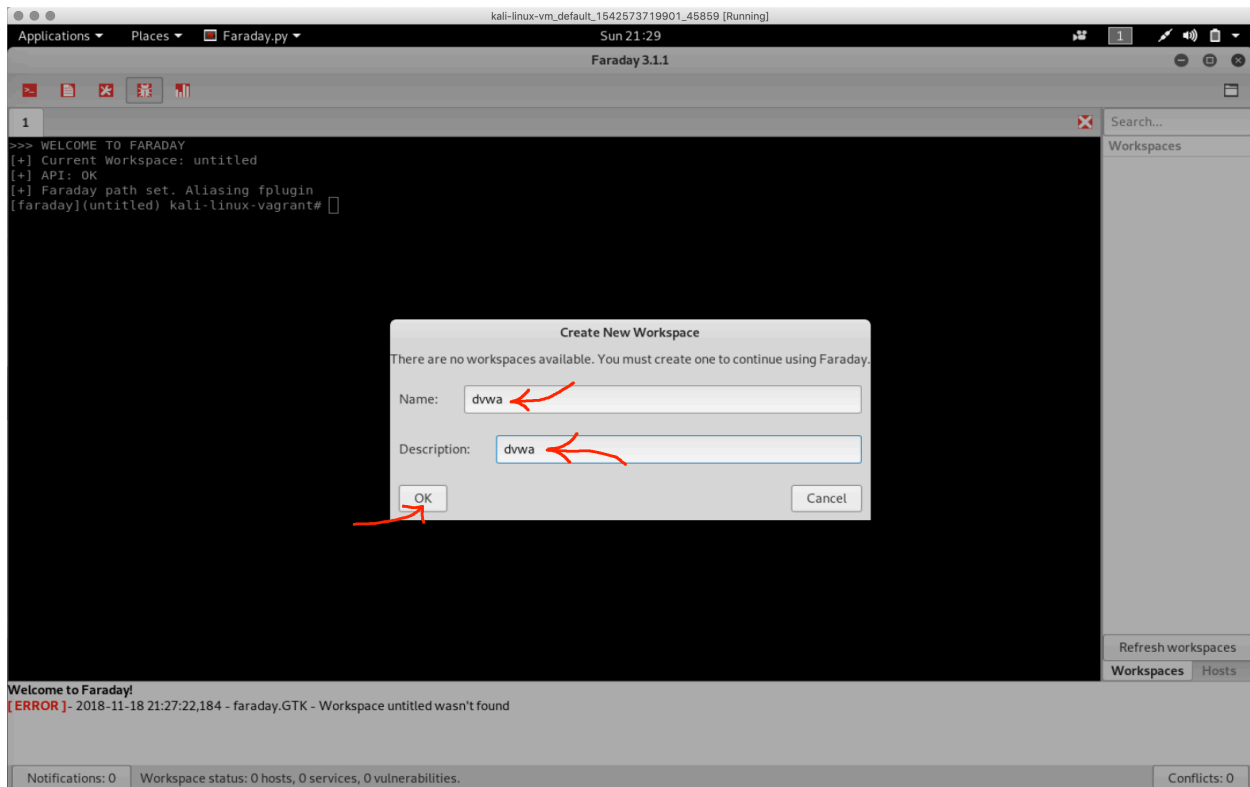
The author's home wireless network uses 172.20.156.0/24 as the network range. Therefore, the adapter, enp0s8 is what he is looking for. The IP to use as a target is 172.20.156.76. Write down your value.

Faraday IDE (Kali Linux version)

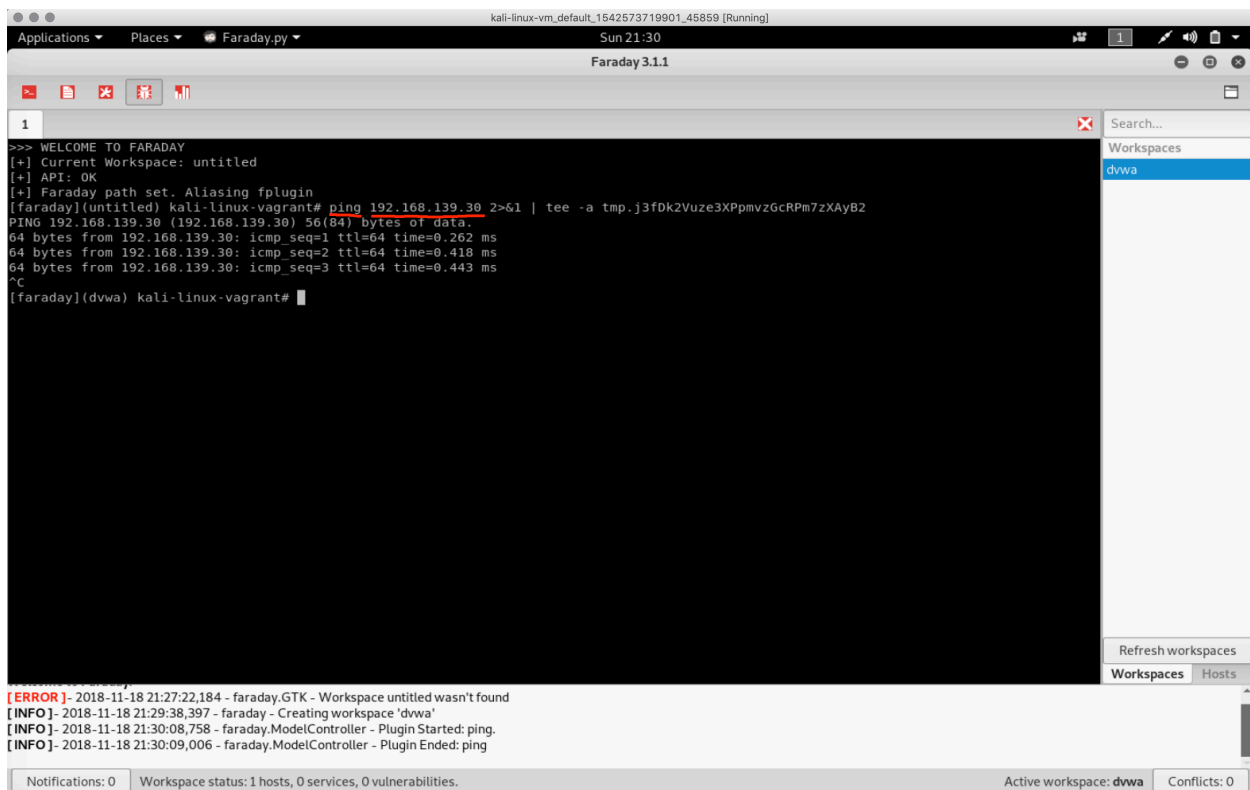
First launch both Vagrant boxes for Kali-Linux and DVWA.

For attack sequence, you can simple copy and paste out of the Appendix section in the right order. Change the ip 192.168.139.30 to your IP for your DVWA instance.

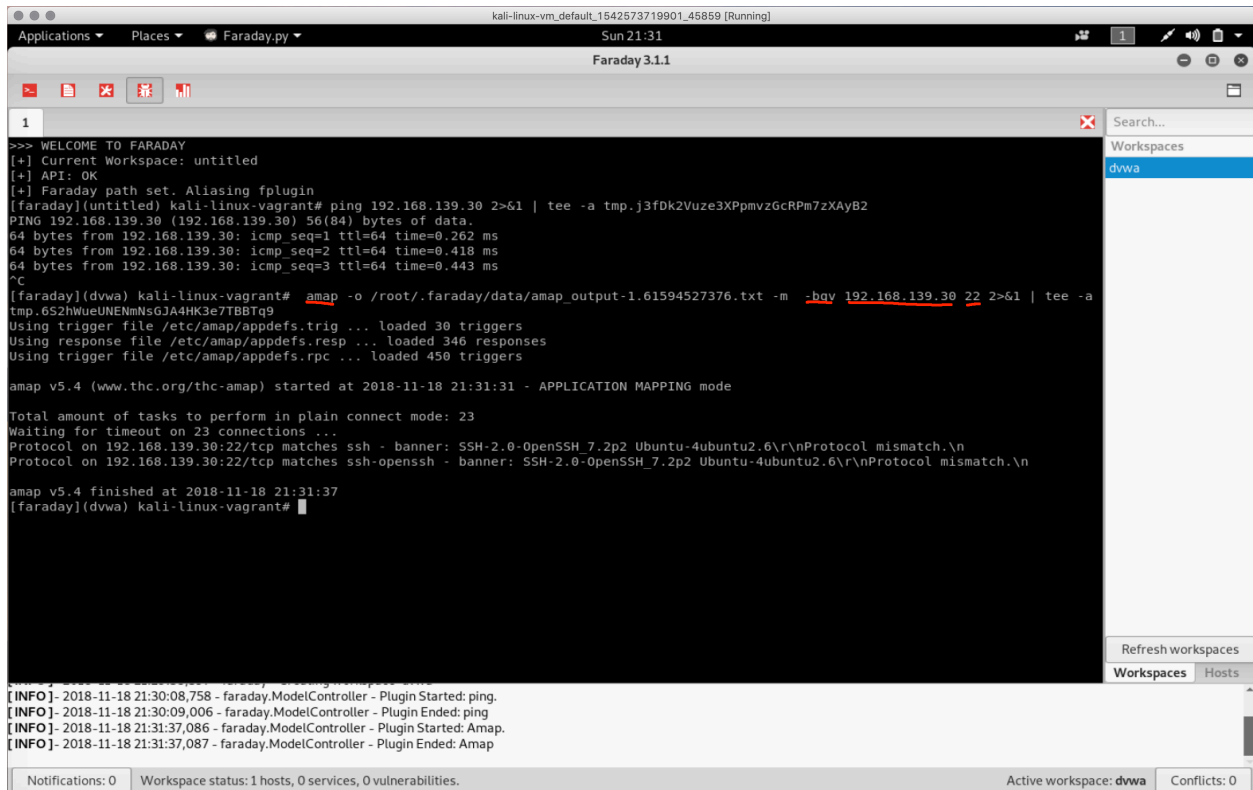
Then log into Kali-Linux with username: root and password: toor.



In the console/terminal, type in: `ping 192.168.139.30`



Type in: `amap -bqv 192.168.139.30 22` then `amap -bqv 192.168.139.30 80`.



The screenshot shows the Faraday 3.1.1 application window. The terminal displays the following output:

```
>>> WELCOME TO FARADAY
[+] Current Workspace: untitled
[+] API: OK
[+] Faraday path set. Aliasing fplugin
[faraday](untitled) kali-linux-vagrant# ping 192.168.139.30 2>&1 | tee -a tmp.j3fdk2Vuze3XPpmvzGcRPm7zXAY82
PING 192.168.139.30 (192.168.139.30) 56(84) bytes of data:
64 bytes from 192.168.139.30: icmp_seq=1 ttl=64 time=0.262 ms
64 bytes from 192.168.139.30: icmp_seq=2 ttl=64 time=0.418 ms
64 bytes from 192.168.139.30: icmp_seq=3 ttl=64 time=0.443 ms
^C
[faraday](dwwa) kali-linux-vagrant# amap -o /root/.faraday/data/amap_output-1.61594527376.txt -m -bqv 192.168.139.30 22 2>&1 | tee -a
tmp.652hwueUNENmNsGJA4HK3e7TBBTq9
Using trigger file /etc/amap/appdefs.trig ... loaded 30 triggers
Using response file /etc/amap/appdefs.resp ... loaded 346 responses
Using trigger file /etc/amap/appdefs.rpc ... loaded 450 triggers

amap v5.4 (www.thc.org/thc-amap) started at 2018-11-18 21:31:31 - APPLICATION MAPPING mode

Total amount of tasks to perform in plain connect mode: 23
Waiting for timeout on 23 connections ...
Protocol on 192.168.139.30:22/tcp matches ssh - banner: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.6\r\nProtocol mismatch.\n
Protocol on 192.168.139.30:22/tcp matches ssh-openssh - banner: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.6\r\nProtocol mismatch.\n

amap v5.4 finished at 2018-11-18 21:31:37
[faraday](dwwa) kali-linux-vagrant#
```

At the bottom of the window, there is a status bar with the following information:

- Notifications: 0
- Workspace status: 1 hosts, 0 services, 0 vulnerabilities.
- Active workspace: dwwa
- Conflicts: 0

Next, type in: `dirb http://192.168.139.30/ /usr/share/wordlists/dirb/common.txt -u admin:admin`

Then: `dirb http://192.168.139.30/ /usr/share/dirb/wordlists/vulns/apache.txt`

```
kali-linux-vm_default_1542573719901_46859 [Running]
Sun 21:33
Faraday 3.1.1
1
By The Dark Raver
-----
START TIME: Sun Nov 18 21:33:27 2018
URL_BASE: http://192.168.139.30/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt
OPTION: Silent Mode
AUTHORIZATION: admin:admin
OPTION: Not Stopping on warning messages
-----
GENERATED WORDS: 4612
---- Scanning URL: http://192.168.139.30/ ----
+ http://192.168.139.30/.git/HEAD (CODE:200|SIZE:23)
+ http://192.168.139.30/.htaccess (CODE:200|SIZE:500)
==> DIRECTORY: http://192.168.139.30/config/
==> DIRECTORY: http://192.168.139.30/docs/
==> DIRECTORY: http://192.168.139.30/external/
+ http://192.168.139.30/favicon.ico (CODE:200|SIZE:1406)
+ http://192.168.139.30/index.php (CODE:302|SIZE:0)
+ http://192.168.139.30/php.ini (CODE:200|SIZE:148)
+ http://192.168.139.30/phpinfo.php (CODE:302|SIZE:0)
+ http://192.168.139.30/robots.txt (CODE:200|SIZE:26)
---- Entering directory: http://192.168.139.30/config/ ----
---- Entering directory: http://192.168.139.30/docs/ ----
---- Entering directory: http://192.168.139.30/external/ ----
-----
END TIME: Sun Nov 18 21:33:34 2018
DOWNLOADED: 18448 FOUND: 7
[Faraday](dwa) kali-linux-vagrant#

[INFO] - 2018-11-18 21:31:37,086 - faraday.ModelController - Plugin Started: Amap.
[INFO] - 2018-11-18 21:31:37,087 - faraday.ModelController - Plugin Ended: Amap.
[INFO] - 2018-11-18 21:33:34,412 - faraday.ModelController - Plugin Started: dirb.
[INFO] - 2018-11-18 21:33:34,667 - faraday.ModelController - Plugin Ended: dirb

Notifications: 0 Workspace status: 1 hosts, 1 services, 0 vulnerabilities. Active workspace: dwa Conflicts: 0
```

Now run: `sqlmap -u "http://192.168.139.30/?p=1&forumaction=search" --dbs`

```
kali-linux-vm_default_1542573719901_46859 [Running]
Sun 21:35
Faraday 3.1.1
1
[21:35:15] [INFO] testing 'Oracle stacked queries (DBMS PIPE.RECEIVE_MESSAGE - comment)'
[21:35:15] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[21:35:16] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[21:35:16] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[21:35:16] [INFO] testing 'Oracle AND time-based blind'
[21:35:16] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[21:35:17] [WARNING] GET parameter 'p' does not seem to be injectable
[21:35:17] [WARNING] GET parameter 'forumaction' does not appear to be dynamic
[21:35:17] [WARNING] heuristic (basic) test shows that GET parameter 'forumaction' might not be injectable
[21:35:17] [INFO] testing for SQL injection on GET parameter 'forumaction'
[21:35:17] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[21:35:17] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[21:35:17] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[21:35:17] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[21:35:17] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[21:35:17] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[21:35:17] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[21:35:17] [INFO] testing 'MySQL inline queries'
[21:35:17] [INFO] testing 'PostgreSQL inline queries'
[21:35:17] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[21:35:17] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[21:35:18] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[21:35:18] [INFO] testing 'Oracle stacked queries (DBMS PIPE.RECEIVE_MESSAGE - comment)'
[21:35:18] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[21:35:18] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[21:35:18] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[21:35:18] [INFO] testing 'Oracle AND time-based blind'
[21:35:18] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[21:35:19] [WARNING] GET parameter 'forumaction' does not seem to be injectable
[21:35:19] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment')

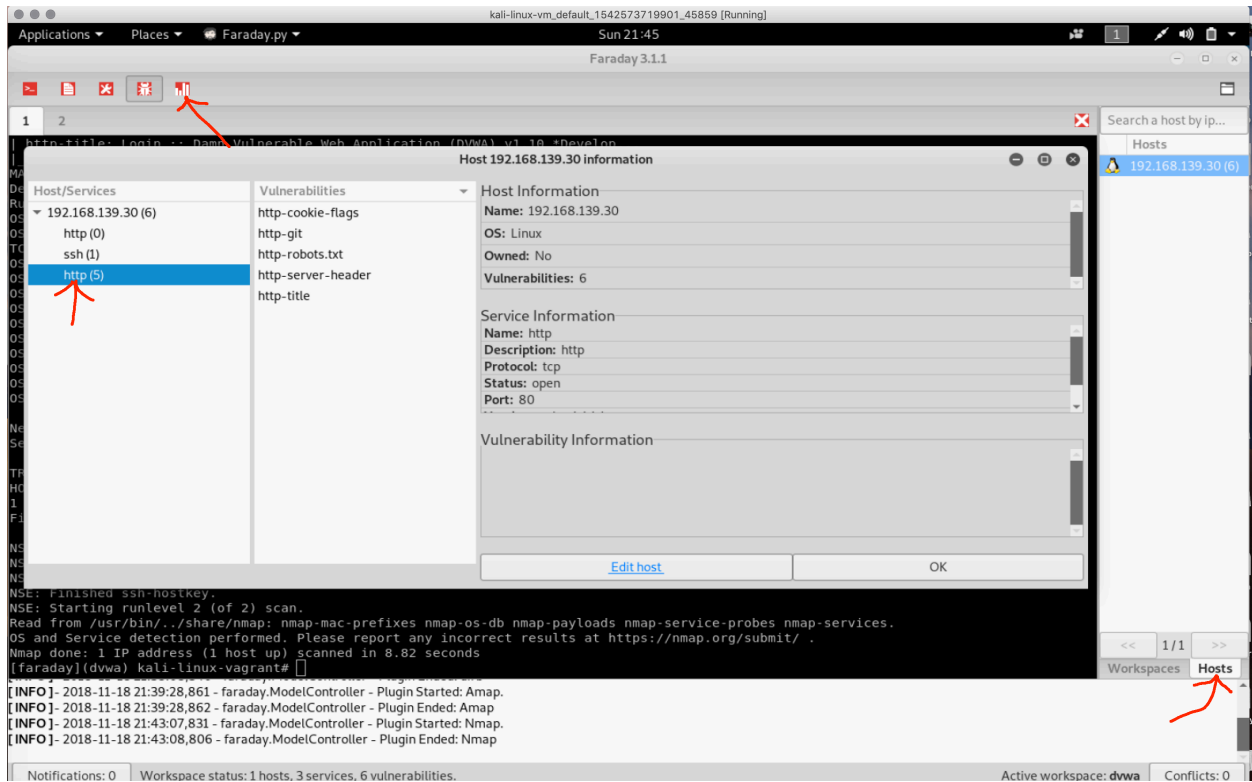
[*] shutting down at 21:35:19

[Faraday](dwa) kali-linux-vagrant#

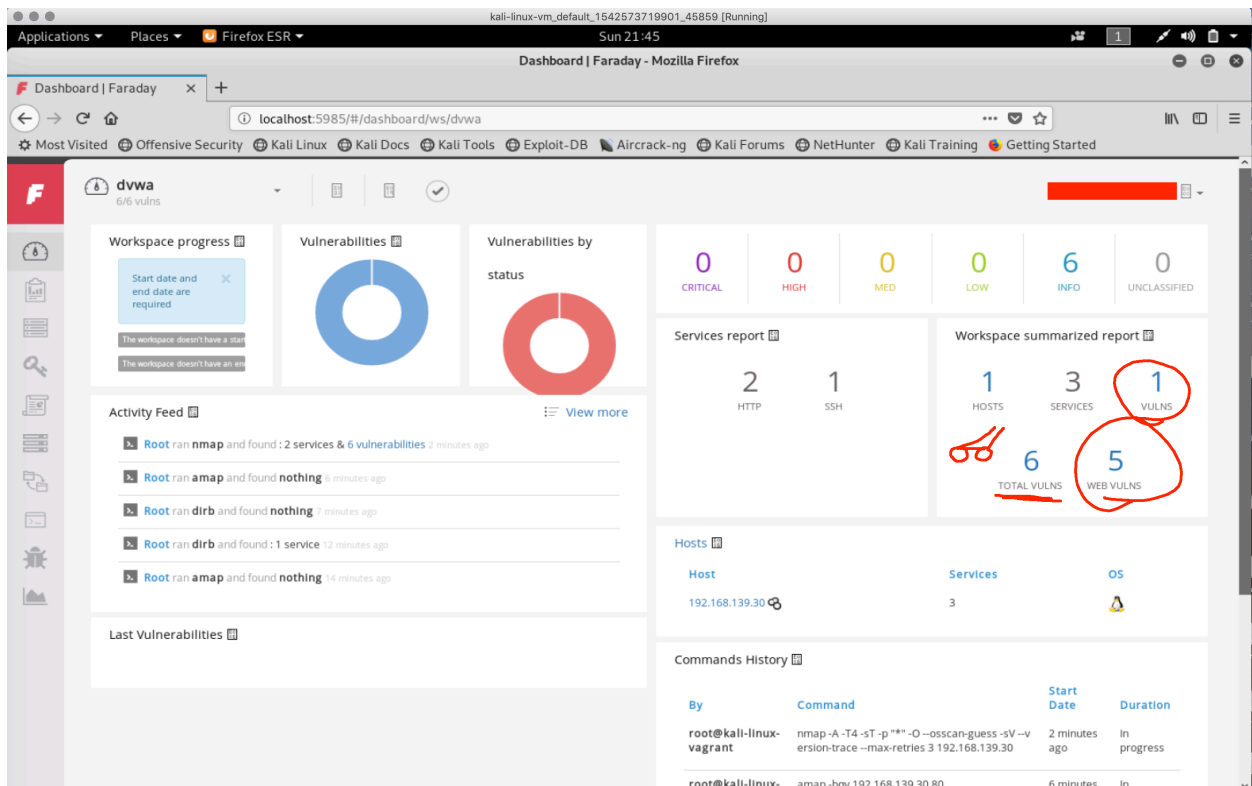
[INFO] - 2018-11-18 21:31:37,087 - faraday.ModelController - Plugin Ended: Amap
[INFO] - 2018-11-18 21:33:34,412 - faraday.ModelController - Plugin Started: dirb.
[INFO] - 2018-11-18 21:33:34,667 - faraday.ModelController - Plugin Ended: dirb
[ERROR] - 2018-11-18 21:35:02,094 - tornado.access - 500 POST /cmd/input(?!1) 6.23ms

Notifications: 0 Workspace status: 1 hosts, 1 services, 0 vulnerabilities. Active workspace: dwa Conflicts: 0
```

Now, click on the upper left red button that looks like graphs.



This opens a web page with some good information.



Click on the Chart icon on the left. Here we see more about the vulnerabilities.

The screenshot shows a web application security tool interface. The browser address bar shows the URL `localhost:5985/#/status/ws/dvwa/search/type=VulnerabilityWeb`. The page title is "Status Report | Faraday - Mozilla Firefox". The interface includes a sidebar on the left with various icons, including a chart icon. The main content area displays a table of search results for "type:VulnerabilityWeb".

SEV	NAME	SERVICE	HOSTNAMES	TARGET	DESC	ID	DATE	STA
INFO	http-title	(80/tcp) http (n...		192.168.139.30	Login :: Damn Vulnerable Web Application (DVWA) v1.10 *Develop... Requested resource was login...	6	4 minutes ago	
INFO	http-server-hea...	(80/tcp) http (n...		192.168.139.30	nginx/1.14.1 Output: None: nginx/1.14.1	5	4 minutes ago	
INFO	http-robots.txt	(80/tcp) http (n...		192.168.139.30	1 disallowed entry /	4	4 minutes ago	
INFO	http-git	(80/tcp) http (n...		192.168.139.30	192.168.139.30:80/ git/ Git repository found! Repository description: Unnamed repository; ed...	3	4 minutes ago	
INFO	http-cookie-flags	(80/tcp) http (n...		192.168.139.30	/: PHPSESSID: httponly flag not set	2	4 minutes ago	

At the bottom right, there is a summary box: Total 5, Viewing 5, Selected 0. At the bottom left, there is a pagination control showing "100 Items per page" and "1 - 5 of 5 items".

Conclusion

By following this guide, the user has installed VirtualBox, Vagrant, Kali-Linux, DVWA and ran multiple vulnerability tests/checks against the DVWA instance. All isolated on the local system. The older version of Faraday had financials tied in with the report so that a person could quantify the cost of having 'n' number of vulnerabilities within an organization. Somehow that went away with the new free version. My guess is the company wants you to pay for a license.

If you are in a job that requires Pen Testing, the author highly recommends testing out Faraday IDE. If your team needs to have multiple people sync up their findings and share outcomes, then pay for the license at: <https://www.faradaysec.com/>

The author does not get paid nor any endorsements from any site. These recommendations are from his 20 years of experience (and bias) in the IT field.

The author truly hopes you enjoyed this guide. He had a blast writing it and learning more about Faraday IDE in depth.

Appendix

Commands reference:

Get latest vulnerabilities:

```
cd /root/.faraday/data/  
wget http://cve.mitre.org/data/downloads/allitems.csv.gz  
gunzip allitems.csv.gz  
mv allitem.csv cwe.csv
```

Import those vulnerabilities:

```
cd /usr/share/python-faraday/  
python2 ./bin/fplugin -username masterf \  
  --password faraday -w dvwa import_csv \  
  --csv /root/.faraday/data/cwe.csv
```

Ping target:

```
ping 192.168.139.30
```

Run amap against SSH and HTTPD services:

```
amap -bqv 192.168.139.30 22  
amap -bqv 192.168.139.30 80
```

Run nmap against everything:

```
nmap -A -T4 -sT -p "*" -O --osscan-guess -sV --version-trace \  
  --max-retries 3 192.168.139.30
```

I'm totally loving dirb. Two popular checks:

```
dirb http://192.168.139.30/ \  
  /usr/share/wordlists/dirb/common.txt -u admin:admin
```

```
dirb http://192.168.139.30/ \  
  /usr/share/dirb/wordlists/vulns/apache.txt
```

Test the site with sqlmap. TODO, run a better check than forumaction=search.

```
sqlmap -u "http://192.168.139.30/?p=1&forumaction=search" --dbs
```