How to scan DVWA with the Free edition of Burp Suite

Introduction

The motivation behind this paper is to have a working reference model for scanning any site with the free edition of Burp Suite. I am using DVWA (Damn Vulnerable Web Application) because I can control the route to said, and the host; thereby avoiding any legal predicaments. This boils down to that both hosts are running on the same laptop. Be exceedingly cautious with Burp Suite to limit the scope of what it can scan. The author of this paper will not be responsible for dummies scanning the wrong site and their to-be legal problems.

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"]
     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='u8392ihj32kl8hujalkshuil32'
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_dvwa_user}"'@'localhost' IDENTIFIED BY
'"${MYSQL_dvwa_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_dvwa(){

    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
```

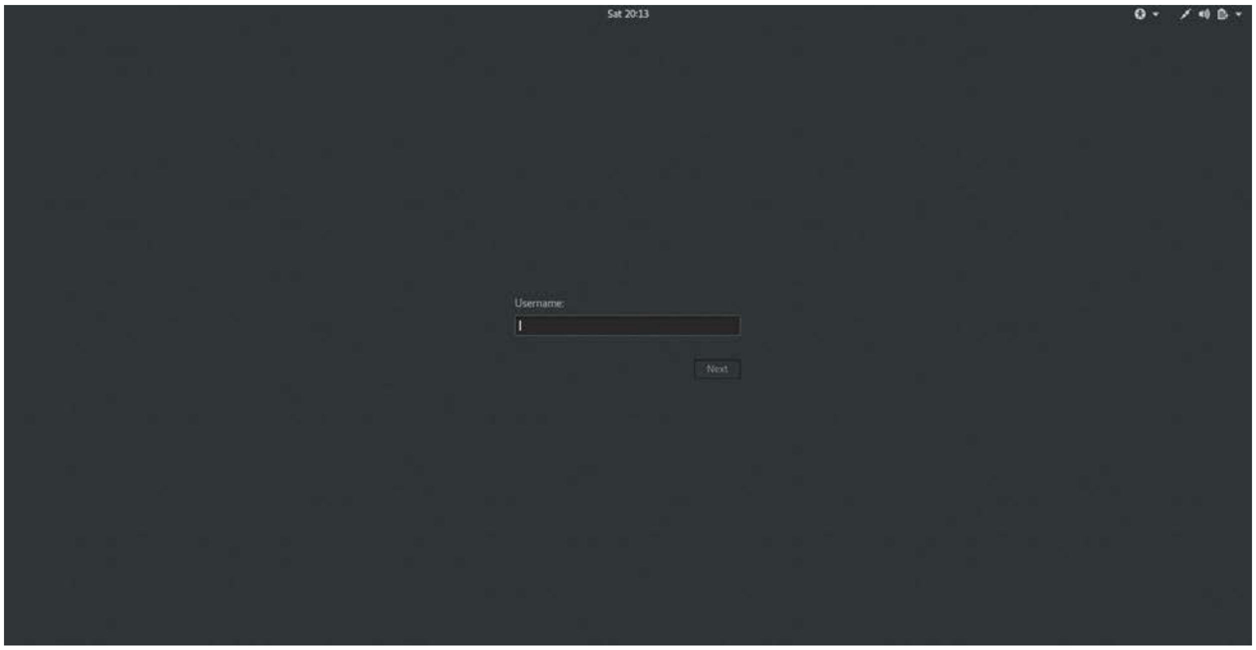
The author’s home wireless network uses 172.20.156.0/24 as the network range. Therefore, the adapter, enp0s8 is what he is looking for. The IP to use as a target is 172.20.156.76. Write down your value.

Burp Suite

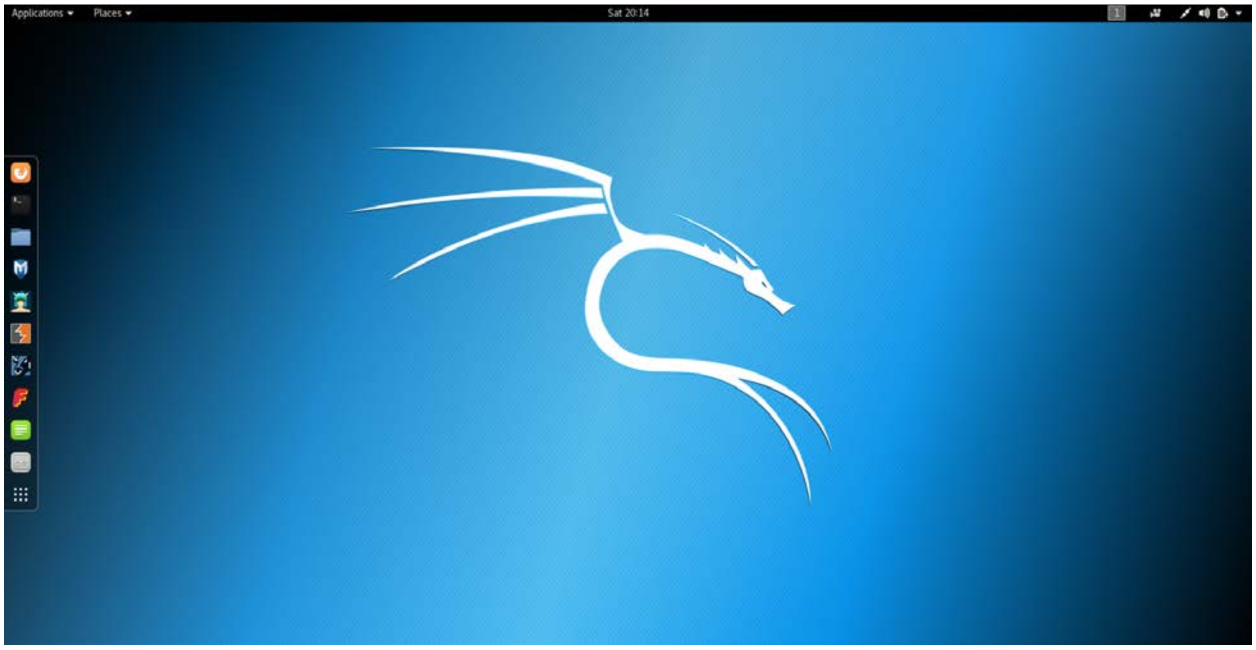
Start both Kali-Linux and DVWA vagrant systems up.

Via the GUI console for Kali-Linux, log in with:
username: root

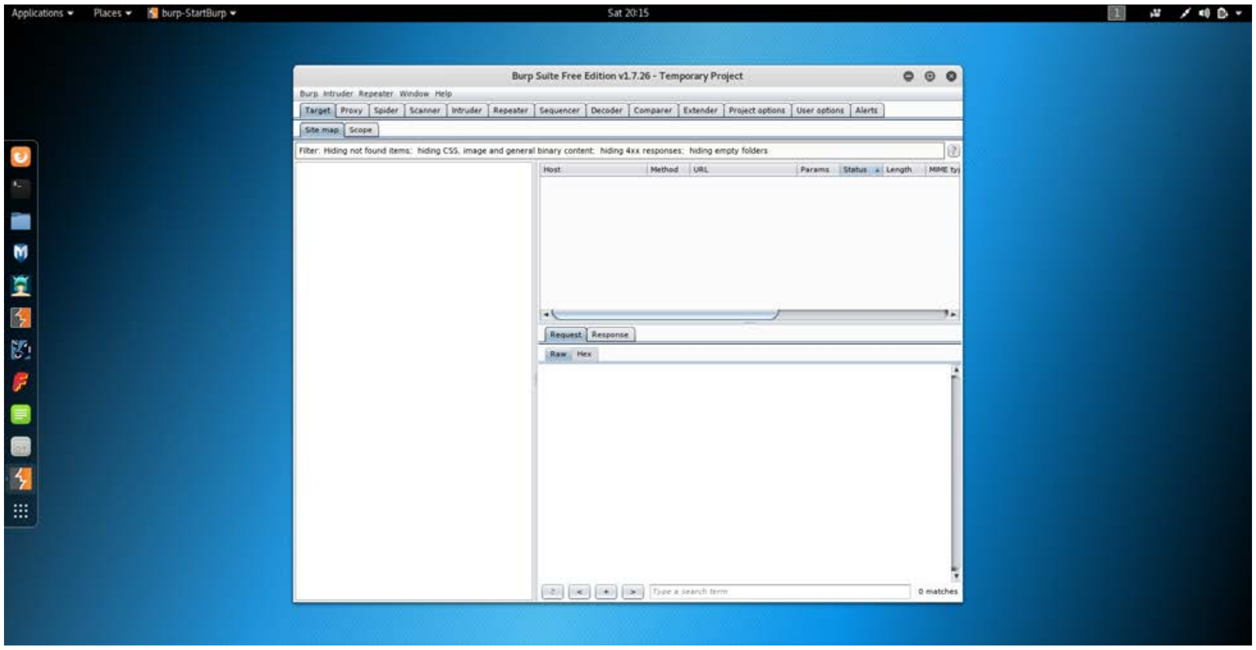
password: toor



The following screen shows the desktop.

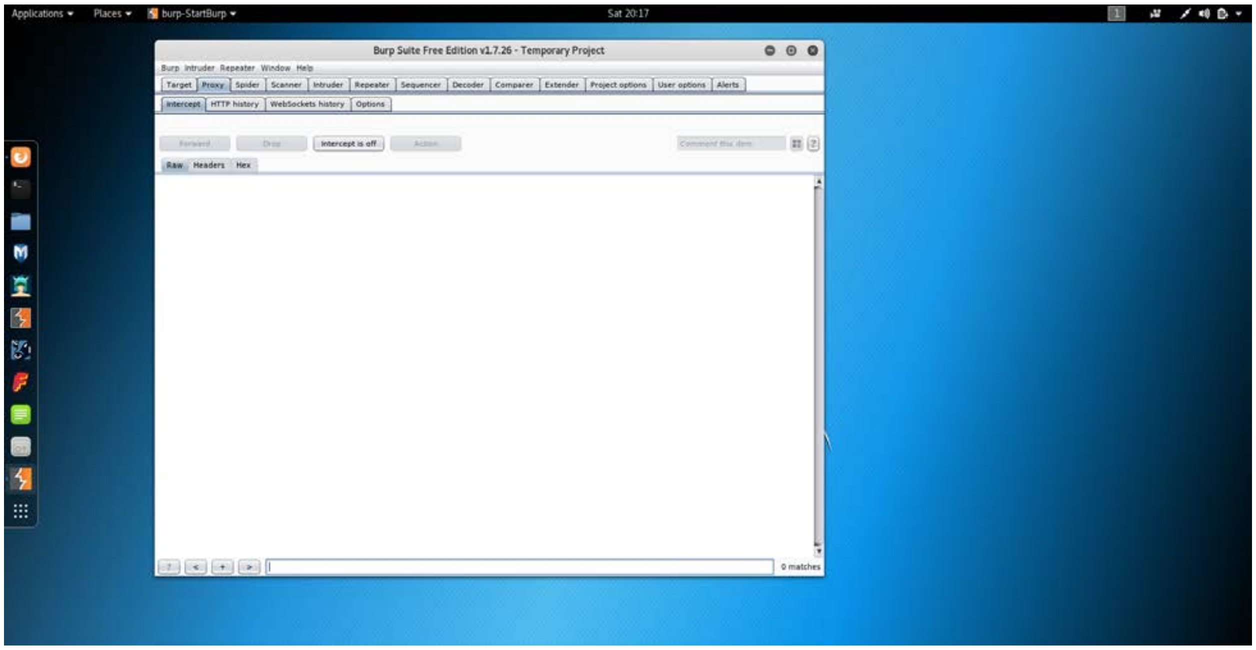


Go ahead and open Burp Suite, in the picture above, the sixth icon down on the sidebar.

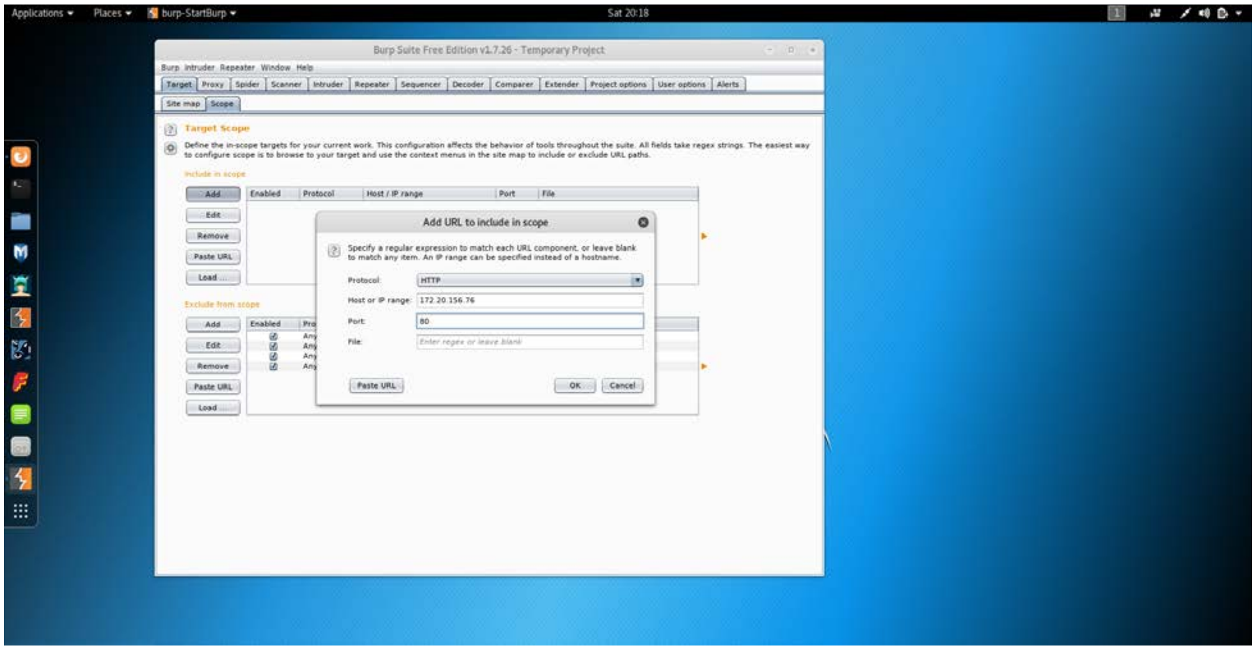

Applications
Places
burp-StartBurp
Sat 20:15
Burp Suite Free Edition v1.7.26 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Site map Scope
Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
Host Method URL Params Status Length MIME type
Request Response
Raw Hex
0 matches

Above we see the first page once activated.

Click on the Proxy tab, then Intercept tab, to see if the Intercept is on or off.


Applications
Places
burp-StartBurp
Sat 20:17
Burp Suite Free Edition v1.7.26 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is off Action
Comment this item
Raw Headers Hex
0 matches

Next, open the Target, then Scope tabs. In Target Scope, click on Add, then give it the IP address of your target. In this case, it is DVWA.


Applications
Places
burp-StartBurp
Sat 20:18
Burp Suite Free Edition v1.7.26 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Site map Scope
Add
Edit
Remove
Paste URL
Load ...
Paste URL
OK
Cancel

Target Scope

Define the in-scope targets for your current work. This configuration affects the behavior of tools throughout the suite. All fields take regex strings. The easiest way to configure scope is to browse to your target and use the context menus in the site map to include or exclude URL paths.

Include in scope

Enabled	Protocol	Host / IP range	Port	File

Exclude from scope

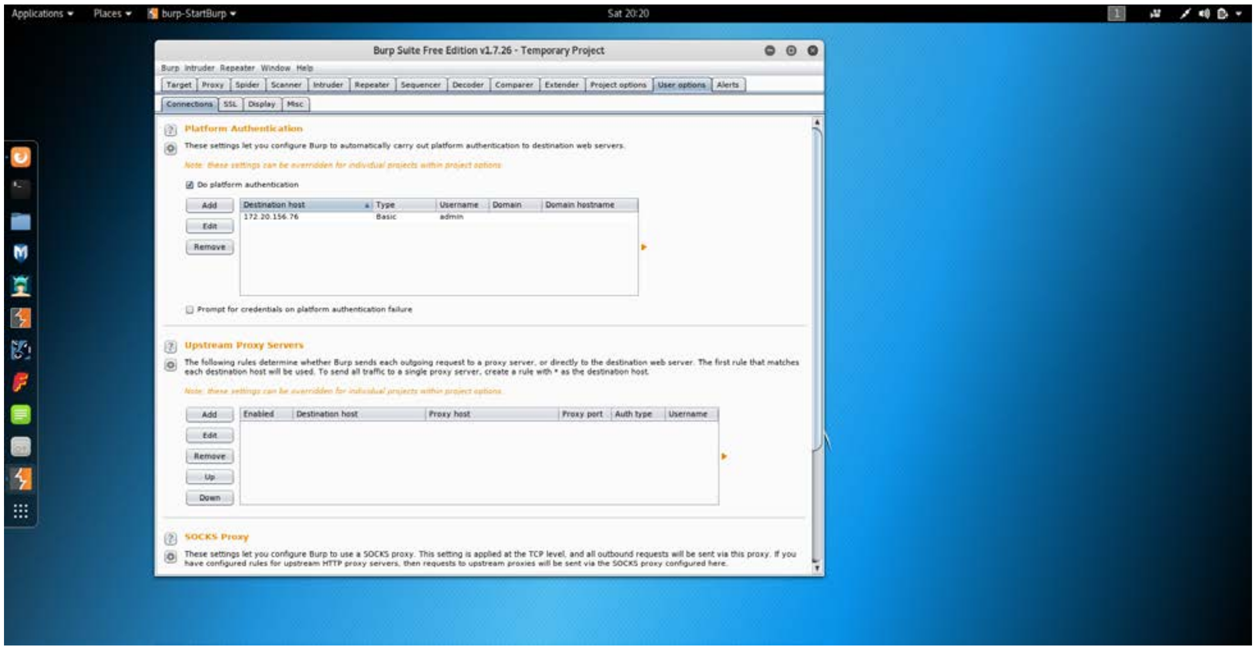
Enabled	Pro
✓	Any
✓	Any
✓	Any
✓	Any

Add URL to include in scope

Specify a regular expression to match each URL component, or leave blank to match any item. An IP range can be specified instead of a hostname.

Protocol: HTTP
Host or IP range: 172.20.156.76
Port: 80
File: Enter regex or leave blank

Under User options, you can add credentials, as needed.


Applications
Places
burp-StartBurp
Sat 20:20
Burp Suite Free Edition v1.7.26 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Connections SSL Display Misc
Add
Edit
Remove
Up
Down

Platform Authentication

These settings let you configure Burp to automatically carry out platform authentication to destination web servers.

Note: these settings can be overridden for individual projects within project options

☑ Do platform authentication

Destination host	Type	Username	Domain	Domain hostname
172.20.156.76	Basic	admin		

☐ Prompt for credentials on platform authentication failure

Upstream Proxy Servers

The following rules determine whether Burp sends each outgoing request to a proxy server, or directly to the destination web server. The first rule that matches each destination host will be used. To send all traffic to a single proxy server, create a rule with * as the destination host.

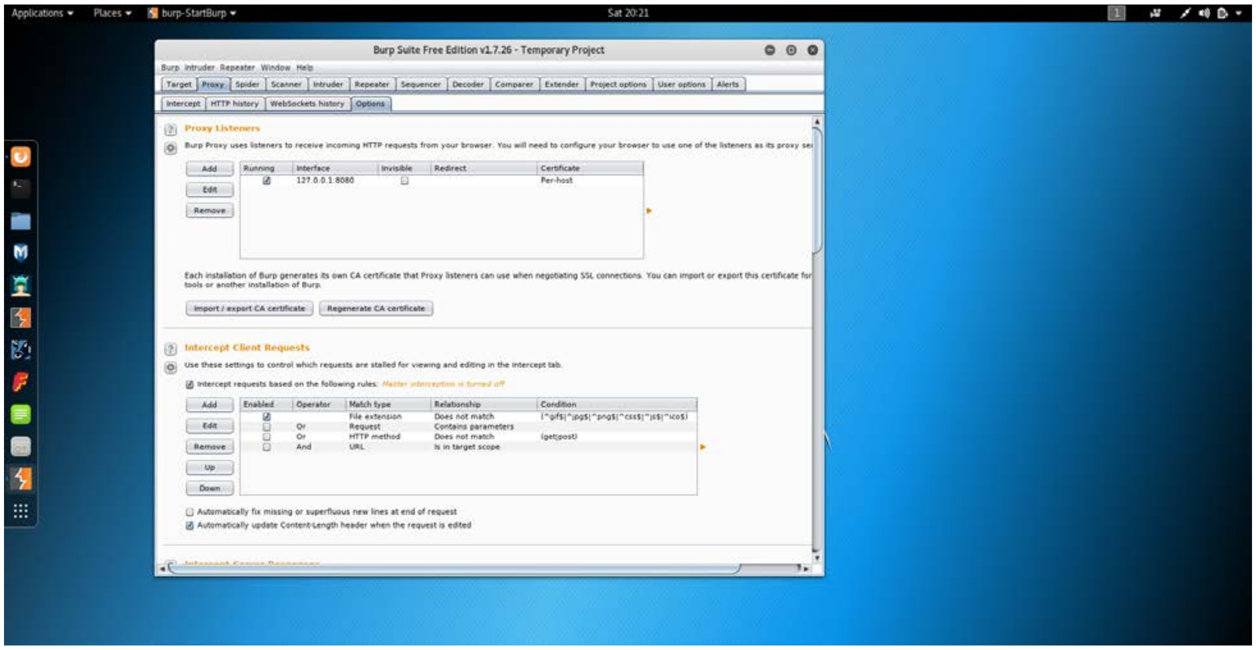
Note: these settings can be overridden for individual projects within project options

Enabled	Destination host	Proxy host	Proxy port	Auth type	Username

SOCKS Proxy

These settings let you configure Burp to use a SOCKS proxy. This setting is applied at the TCP level, and all outbound requests will be sent via this proxy. If you have configured rules for upstream HTTP proxy servers, then requests to upstream proxies will be sent via the SOCKS proxy configured here.

Next, on Proxy, then Options, we see that this is the page to set your local proxy that you will then configure your Firefox browser to proxy to 127.0.0.1:8080 as an HTTP proxy, and then check the box for all protocols (inside the Firefox browser). Go ahead and launch Firefox if you haven't and set the proxy. Then open a new tab and navigate to your target web page (DVWA which for me is on 172.20.156.76).


Applications
Places
burp-StartBurp
Sat 20:21
Burp Suite Free Edition v1.7.26 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Add
Edit
Remove
Import / export CA certificate
Regenerate CA certificate
Up
Down

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy se

Running	Interface	Invisible	Redirect	Certificate
✓	127.0.0.1:8080	☐		Per-host

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating SSL connections. You can import or export this certificate for tools or another installation of Burp.

Intercept Client Requests

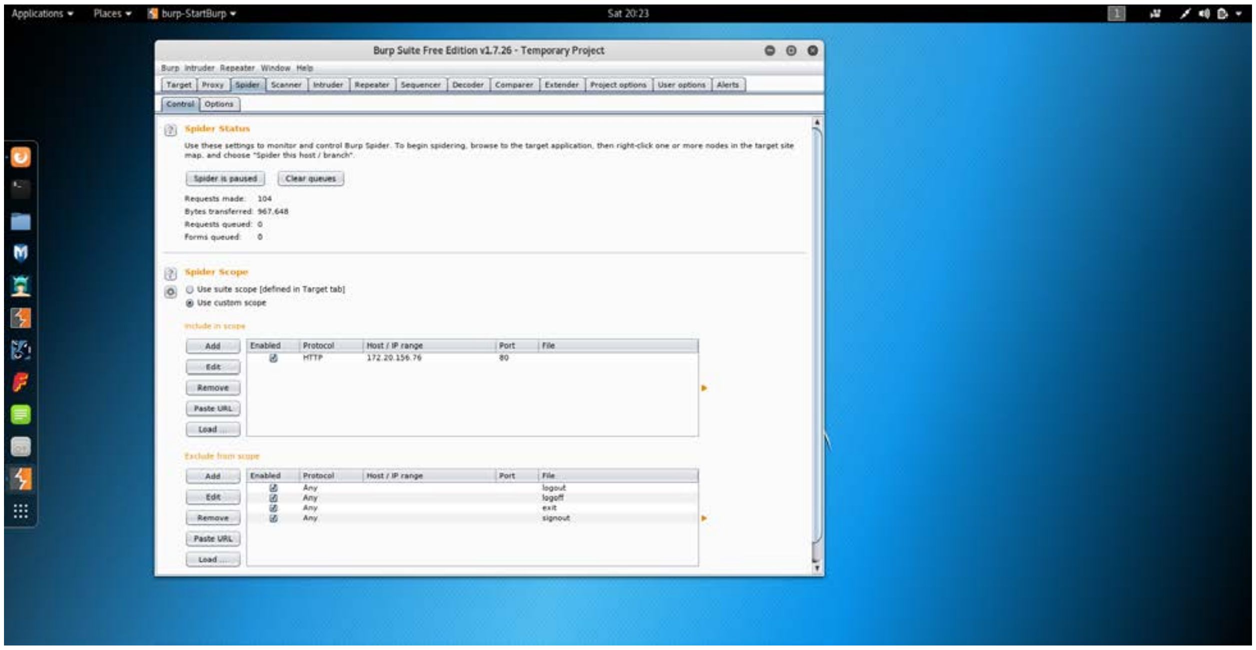
Use these settings to control which requests are stalled for viewing and editing in the intercept tab.

✓ Intercept requests based on the following rules: Master interception is turned off

Enabled	Operator	Match type	Relationship	Condition
✓		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$\|^ico$)
☐	Or	Request	Contains parameters	
☐	Or	HTTP method	Does not match	(get|post)
☐	And	URL	Is in target scope	

☐ Automatically fix missing or superfluous new lines at end of request
✓ Automatically update Content-Length header when the request is edited

Verify on the Target tab that your scope is set to your IP address of the target. Then on the Spider tab, change the Spider Scope to the second radio button down and define the custom scope, in this case, HTTP, 172.20.156.76, Port 80. Click on the Spider's paused button to un-pause the Spider. This will crawl your site.


Applications
Places
burp-StartBurp
Sat 20:23
Burp Suite Free Edition v1.7.26 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Control Options
Spider is paused
Clear queues
Add
Edit
Remove
Paste URL
Load ...

Spider Status

Use these settings to monitor and control Burp Spider. To begin spidering, browse to the target application, then right-click one or more nodes in the target site map, and choose "Spider this host / branch".

Requests made: 104
Bytes transferred: 967,648
Requests queued: 0
Forms queued: 0

Spider Scope

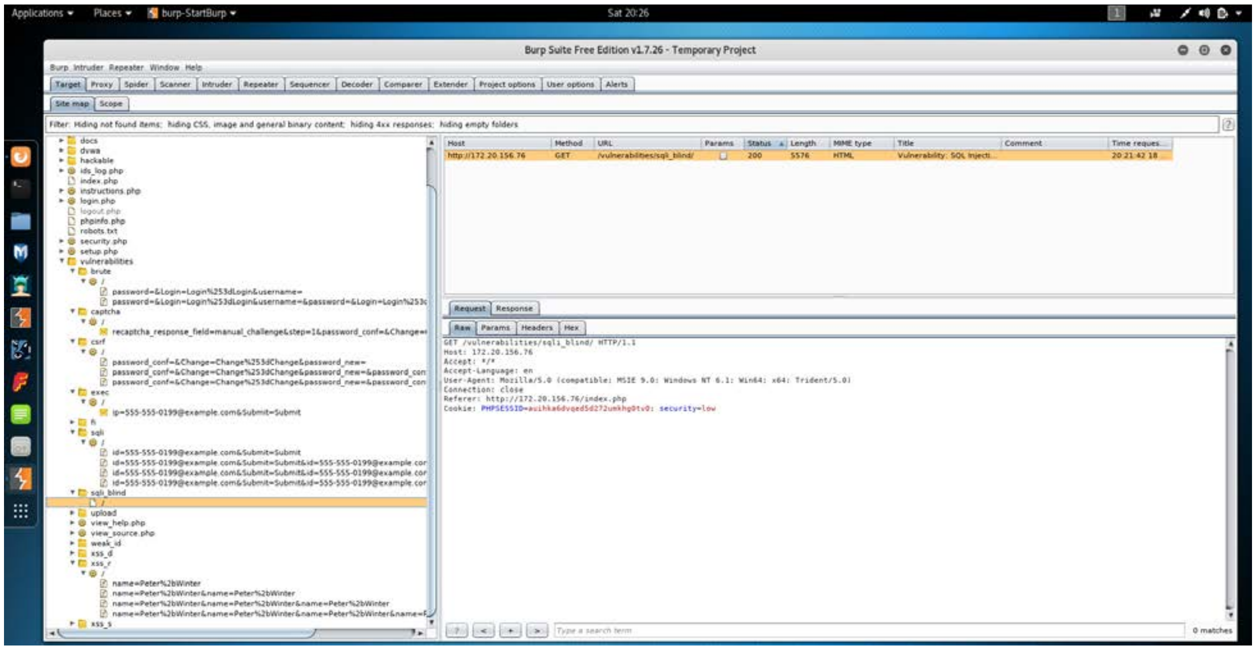
○ Use suite scope [defined in Target tab]
● Use custom scope

Include in scope

Enabled	Protocol	Host / IP range	Port	File
✓	HTTP	172.20.156.76	80	

Exclude from scope

Enabled	Protocol	Host / IP range	Port	File
✓	Any			logout
✓	Any			logoff
✓	Any			exit
✓	Any			signout



Host	Method	URL	Params	Status	Length	MIME type	Title	Comment	Time reques...
http://172.20.156.76	GET	/vulnerabilities/sqli_blind/	☐	200	5576	HTML	Vulnerability: SQL Injecti...		20:21:42 18

```
GET /vulnerabilities/sqli_blind/ HTTP/1.1
Host: 172.20.156.76
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Referer: http://172.20.156.76/index.php
Cookie: PHPSESSID=auihka6dvqed5d272umkhg0tv0; security=low
```

After the Spider is finished, very quick, you will have results back in your main target tab. Here we see the many vulnerabilities that of course exist in DVWA.

Conclusion

By following this guide, you will have setup both Kali-Linux and DVWA, and launched Burp Suite with a successful attack of DVWA (on your local system). This is just the beginning. Kind of like being in kindergarten. Burp Suite is so powerful. If you are heavily into penetration testing and/or vulnerability assessments, the author highly recommends purchasing a licensed copy of Burp Suite Professional. There are a lot more baked in attacks and it does more right out of the box. This is not saying you cannot add more tests to the free edition, but it comes down to how much you know. If you have a masters or Ph.D in Computer Science, then this is a worthy challenge. If you don't, then the chore of finding and injecting worthwhile tests is not for the faint of heart. This stuff is hard; Burp Suite does a decent job of making testing less painful for the average person.

The author is not getting paid any money for these papers, nor does he receive any compensation from external companies. In two different jobs, he was worked as a pen tester and used the professional version of Burp Suite to provide solid information back to customers on vulnerabilities within their respective infrastructures. What is shown in this paper is less than 20% of what the capabilities that the Professional version provides.