# How to perform a security assessment with Paros Proxy using Kali Linux

# Introduction

Paros Proxy is a security and vulnerability testing tool. Paros can be used to spider/crawl an entire site (URL), and then execute pre-written vulnerability tests. Paros comes with a built-in utility that can proxy traffic (Yay us!). The Paros Proxy utility can be used to tamper or manipulate any http or https traffic on the fly (even more Yay). It will help you isolate a sites potential areas of security concern and then manually perform additional testing based on your customer's requirements.

To get this running, we are going to use a combination of VirtualBox and Vagrant. The author is doing this to stay current with technology, but also to have a viable platform to re-launch these apps many, many times to test in different combinations with multiple tools. If one set of tools fails, then another can be brought up in minutes, and refocused on the target platform. That is the beauty of Vagrant -- repeatable instances.

# **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: <u>https://www.virtualbox.org/wiki/Downloads</u> and download VirtualBox.

The author is running on Ubuntu 17.04, so following to this URL: <u>https://www.virtualbox.org/wiki/Linux\_Downloads</u>

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 192.168.139-N TP: 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 192.168.139.3 IP: NetworkMask: 255.255.255.0 lowerIPAddress: 192.168.139.101 upperIPAddress: 192.168.139.254 Enabled: Yes NetworkName: HostInters 172.20.0.3 HostInterfaceNetworking-vboxnet0 NetworkMask: 255.255.255.0 lowerIPAddress: 172.20.0.101 upperIPAddress: 172.20.0.254 Enabled: Yes NetworkName: HostInterfaceNetworking-vboxnet1 0.0.0.0 IP: 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
```

## Add VirtualBox Networking

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

# repeat as many times as necessary to delete all of them.

```
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: <u>https://www.vagrantup.com/downloads.html</u>, 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, "--bootl", "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): state flome // 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
MYSOL 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
ł
   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 glen 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 glen 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.

## **Paros Proxy on Kali Linux**

Go ahead and launch both Kali Linux and DVWA in separate Terminals, with "vagrant up". You will need to change directory into each one, separately, and run vagrant up.

Log into the Kali Linux GUI:



#### Username: root Password: toor

Once logged in, let's open Paros Proxy. Upper left hand corner, click on Applications  $\rightarrow$  03 Web Application Analysis  $\rightarrow$  paros.

Applications 👻	Places 👻 🦉 org-parosproxy-paros-Paros 👻	Sun 13:15	
Applications •	Paces * 📽 org-parosprony-paros Paros *	Paros  Paros Paros License Agreement - the Clarified Artistic License. For the other libraries included in Paros, please refer to respective licenses of the libraries included in paros, please refer to respective licenses of the libraries included in status, please refer to respective licenses of the libraries included in status, please refer to respective licenses of the libraries included in status, please refer to respective licenses of the libraries included in places, please refer to respective licenses of the libraries included via the package. Sublice Delta Vedoya Parolio Parolio Decime Decime of artistic control over the development of the package, wide giving the users of the package the right to use and distribute the Paroshage medifications. Decime De	
		reasonable modifications.	
**			

Click on Accept.

Applications • Places • 📽 org-parosproxy-paros-Paros •	Sun 13:17	I # /4D+
Elle Edit View Analyse Report Tools Help Sites Sites	United Session - Paros	
	Options           Connection         Local proxy           Authentiation         Address (eg localitost, 127.0.0.1)           Certificate         Port (eg 6080)           Trip         Star of the same port as above.	
	OK Cancel	
History Solder Alerts Output		

Open Tools, Options. Make sure Address is set to localhost and Port is 8080. Click OK. Shrink Paros.

In the tool bar on the left hand side, click the Orange Firefox button on top.

Enter as a URL: about:preferences#Advanced, then select the middle tab Network, click on the Settings button. Select the radio button Manual proxy configuration. Input HTTP Proxy as 127.0.0.1 and Port 8080. Check the box for Use this proxy server for all protocols. Then click on OK.

		Preferences - Mozilla Firefox						0	0
Welcome to Firefox	Kali Linux, an Offensive S × DPrefere	nces × +							
Correfox about pre	ferences#advanced			a	Q. Search	4	- 0 +	ŵ	8
Most Visited~	ensive Security 🔨 Kali Linux 🥆 Kali Docs 🥆 Kali Too	is 🤏 Exploit-DB 🐚 Aircrack-ng 🔟 Kali Forums 🥆 NetHunter 📵 Getting S	tarted						
1	1	Connection Settings							
👩 General	Advanced	Connection and any			~				
Q Smitch	Advanced	Configure Proxies to Access the Internet			î.				
	General Data Choices Ne	Auto-detect provy settings for this network							
<ul> <li>Sections:</li> </ul>		Use system proxy settings							
Applications	Connection	Manual proxy configuration:							
🗢 Privacy	Configure now Firefox connects to th	HTTP Progy: 127.0.0.1	Port	8080 :					
🔒 Security	Cached Web Content	<ul> <li>Use this proxy server for all protocols</li> </ul>							
C) Sync	Your web content cache is currently	SSI_Proxy: 127.0.0.1	Port:	8080					
& Advanced	Qverride automatic cache manag	ETP Proxy: 127.0.0.1	Port:	8080					
		SOCKS Host 127.0.0.1	Port:	8080					
	Offline Web Content and User Da Your application cache is currently us	SOCKS v4 C SOCKS v5							
	Tell me when a website asks to st The following websites are allowed to	localhost, 127.0.0.1							
		Example: .mozilla.org, .net.nz, 192.168.1.0/24							
		Automatic proxy configuration URL:							
				Reload					
		Do not prompt for authentication if password is saved							
		Provi DNS when using SOCKS uS Help	Cance	ОК	Ť.				

Back in the Firefox browser [inside of Kali Linux], enter in the URL: 172.20.156.76 (the DVWA ip address you wrote down earlier for your local instance), hit enter. Login, username: admin, password: admin.



Click on the tab DVWA Security.



Change the pull down with Impossible to Low, then click on Submit. Shrink down the Firefox browser.

Back to the Paros user interface.

Applications • Places • 📽 org-parosproxy-paros-Paros •	Sun 13:31	I ~ / + D ·
	Untitled Session - Paros	- 0 - 0
Elle Edit View Analyse Report Tools Help Sites	Request Response Trap	
See     See     Mov/22220156.76     GET-about php     GET-about php     GET-inductions, php     GET-instructions, php     GET-instructions, php     GET-instructions, php(about, php)     GET-instructions, php)     GET-instructions, php(about, php)     GET-instructions, php)     GET-instructions, php(about, php)     GET-instructions, php(about, php)     GET-instructions, php(about, php)     GET-instructions, php)     GET	Control 172.20.154.76 Media 27.20.154.76 Media 26.1111.http://122.20.154.76 Media 26.1111.http://122.20.154.76 Control 101 Media 26.1111.http://122.20.154.76 Control 101 Media 26.1111.http://122.20.154.76 Control 101 Media 26.1111 Final Control 101 Media 26.1111 Control 101 Media 26.11111 Control 101 Media 26.11111 Control 101 Media 26.111111 Control 101 Medi	
	Raw View +	
Alerts		
History Spider Alerts Output		

You should have the IP address [for me, 172.20.156.76] highlight that line. Click on Analyse and select Spider, click on the Start button. This should take seconds, once complete, close the popup window.

Now click on Analyse and this time select Scan (not Scan All [be careful with Scan All, you don't want to scan default sites your browser might have for maintenance checks back to Mozilla.org]).

Applications 👻 Places 👻 📽 org-parosproxy-paros-Paros 👻	Sun 13:32	1 × × + + + + + + + + + + + + + + + + +
0	Untitled Session - Paros	000
Elle Edit View Analyse Report Tools Help		
Sites	Request Response Trap	
* Stec	GET http://172.20.156.76/instructions.php?doc=PHPIDS-license1N/ECTED_PARAM HTTP/1.1	
Mto://172.20.156.76	Host: 172.20.156.76	
GET allowst php GET favicon, ko GET index, php GET:index, php GET:index, php GET:instructions, php(gk), GET:instructions, php(gk), GET:ingoust, php GET:ingoust, php GET:ingoust, php GET:ingoust, php GET:ingoust, php(login, password, user, token, username) POST:ingon, php(login, user, token, username) POST:ingon, php(login, user, token, username) POST:ingon, php(login, user, token, username) POST:ingon, php(login, user, token, user, token) POST:ingon, php(login, user, token, user, tok	Program to excelle Content Type: applications wave form uncleaded Referen http://172.20.156.70/about.php Content-length: 0	
	Raw View 🔻	
Alerts     High     SQL Injection     SQL Injection     Http://17.2.20.156.76/instructions.php?doc=PHPID5-licensetWJECTED.J     http://17.2.20.156.76/instrup.php     http://17.2.20.156.76/70pen     Medium     Lotus Domino default files     http://17.2.20.156.76/70pen     http://17.2.20.156.76/70penServer	абам	
History Spider Alerts Output		

The above is output from the scan.

Click on the line with setup.php.

Applications • Places • 📽 org-parosproxy-paros-Paros •	Sun 13:35	1 4 / 4 8 -
	Untitled Session - Paros	000
Eile Edit View Analyse Report Tools Help		
Sites	Request Response Trap	
Sites     S	POC 1 Http://12.20.154.76/setup.php PrfTPL1 Program FOCATIon Uner-Agent: MocTalA-0 I compatible: MSIE 6-0: Windows NT 5-0:3 PerceP1.2.13 Program FOCATIon Content-Agent: Montal-News form unforceded Content-Anghth 96 Referen: http://12.2.0.156.76/setup.php create_db=Create+%2F+Reset+Database6user_token=2b8b13c9982eec8%c9839342a1b4a6c%2708jECTED_PARAM	
	Raw View v	
Alerts     High     Alerts     Alerts     Alerts     Alerts     SQL Injection     http://172.20.156.76/restructions.php?doc=PHPIDS-license1NJECTEI     http://172.20.156.76/restup.php     bttp://172.20.156.76/restup.php     volum     totus Domino detault files     http://172.20.156.76/POpenServer     http://172.20.156.76/POpenServer	С. Э. РАКАМ	
History Spider Alerts Output		č.

This shows us the syntax that was used to actually attack and inject a new database into the mariadb server [middle pane, right above Raw View].

Attack complete, save as needed or continue to play with the proxy.

#### Conclusion

By following this guide, you have installed VirtualBox, Vagrant, and downloaded two instances and configured DVWA into an operational Linux Web Server chock full of vulnerabilities to exploit and learn (hopefully) how to protect said web services. This is just the beginning; this is an eye opener. If you truly want to master exploitation, the author highly recommends reading The Shellcoders Handbook and setting up virtual instances, like above, to test your skills with. Reading will not give you the practical experience alone. Also, if you follow this path, setup a Github account, and use Jupyter Notebooks with Python to save your work for future reference and collaboration.