

Kali Linux and EyeWitness

Version 0.1, Last Updated: 8 Aug 2021



This site is dedicated to sharing information about the practice, ideas, concepts and patterns regarding computer security.

Table of Contents

1. Introduction	1
2. Requirements	2
2.1. Writing Conventions	2
2.2. VirtualBox	2
2.2.1. Clean VirtualBox Networking	2
2.2.2. Add VirtualBox Networking	3
2.3. Vagrant	4
2.4. Kali Linux and Damn Vulnerable Web Application (DVWA)	4
2.4.1. Vagrantfile	4
2.4.2. bootstrap.sh	7
3. Eyewitness	11
3.1. EyeWitness run	17
4. Conclusion	22
5. Appendix	23

1. Introduction

The motivation behind this paper is to explore using the tool EyeWitness that comes with the Kali Linux distrobution.

"EyeWitness is designed to take screenshots of websites, provide some server header info, and identify default credentials if possible.

...

EyeWitness is designed to run on Kali Linux. It will auto detect the file you give it with the -f flag as either being a text file with URLs on each new line, nmap xml output, or nessus xml output. The -t (timeout) flag is completely optional, and lets you provice the max time to wait when trying to render and screenshot a web page. The -open flag, which is optional, will open the URL in a new tab within Firefox."

source: <https://www.kali.org/tools/eyewitness/>

Let's get this moving!

2. Requirements

2.1. Writing Conventions

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; meaning an account that does not have administrative privileges. 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
```

2.2. VirtualBox

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

The author is running on Ubuntu 18.04, 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.

2.2.1. Clean VirtualBox Networking

This section is here in case you already had virtualbox installed from before. The intent is to clean up the previous networking. If you do not need to do this, skip to [Add VirtualBox Networking](#)

Run these two commands from a Terminal:

```
$ VBoxManage list natnetworks
$ VBoxManage list dhcpservers
```

Output (example):

```

NetworkName: 192.168.139-NAT
IP: 192.168.139.1
Network: 192.168.139.0/24
IPv6 Enabled: No
IPv6 Prefix: fd17:625c:f037:2::/64
DHCP Enabled: Yes
Enabled: Yes
loopback mappings (ipv4)
    127.0.0.1=2

NetworkName: 192.168.139-NAT
Dhcpd IP: 192.168.139.3
LowerIPAddress: 192.168.139.101
UpperIPAddress: 192.168.139.254
NetworkMask: 255.255.255.0
Enabled: Yes
Global Configuration:
    minLeaseTime: default
    defaultLeaseTime: default
    maxLeaseTime: default
    Forced options: None
    Suppressed opts.: None
    1/legacy: 255.255.255.0
Groups: None
Individual Configs: None

NetworkName: HostInterfaceNetworking-vboxnet0
Dhcpd IP: 172.20.0.3
LowerIPAddress: 172.20.0.101
UpperIPAddress: 172.20.0.254
NetworkMask: 255.255.255.0
Enabled: Yes
Global Configuration:
    minLeaseTime: default
    defaultLeaseTime: default
    maxLeaseTime: default
    Forced options: None
    Suppressed opts.: None
    1/legacy: 255.255.255.0
Groups: None
Individual Configs: None

```

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.

Now, delete ALL of the pre-installed DHCP services:

```

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.

2.2.2. 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

```

VirtualBox install complete.

2.3. 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.

2.4. Kali Linux and Damn Vulnerable Web Application (DVWA)

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/

```

From a Terminal, change directory to:

```

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

```

2.4.1. Vagrantfile

Inside of the kali-linux-vm directory, populate a new file with the exact name, "Vagrantfile". Case matters, uppercase the "V". This file will contain both virtual machines for Kali Linux as well as setting up the DVWA virtual machine.

Aggregating both virtual machines into one file has saved the author a lot of time. The coolness here is setting up the variables at the top of the Vagrantfile mimicing shell scripting inside of a virtual machine (passed in with provision: shell). I tested using: `apt-get update && apt-get upgrade -y`, but opted to take it out since it took over 45 minutes on my slower (old) hardware. See comment about downloading this file immediately preceding the code block.

```

# -*- mode: ruby -*-
# vi: set ft=ruby :

$os_update = <<SCRIPT
apt-get update
SCRIPT

VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.define "kali-linux-vagrant" do |conf|
    conf.vm.box = "kalilinux/rolling"

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

    # For macbook/OSx systems, uncomment the line and comment out the Linux Wireless network:
    #conf.vm.network "public_network", bridge: "en0: Wi-Fi (AirPort)", auto_config: true

    conf.vm.hostname = "kali-linux-vagrant"
    conf.vm.provider "virtualbox" do |vb|
      vb.gui = true
      vb.memory = "4096"
      vb.cpus = "2"
      vb.customize ["modifyvm", :id, "--vram", "32"]
      vb.customize ["modifyvm", :id, "--accelerate3d", "off"]
      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
    conf.vm.provision "shell", inline: $os_update
  end

  config.vm.define "dwaa-vagrant" do |conf|

    conf.vm.box = "ubuntu/xenial64"

    conf.vm.hostname = "dwaa-vagrant"

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

    # For macbook/OSx systems, uncomment the line and comment out the Linux Wireless network:
    #conf.vm.network "public_network", bridge: "en0: Wi-Fi (AirPort)", auto_config: true

    config.vm.network "forwarded_port", guest: 80, host: 8080, auto_correct: true
    config.vm.network "forwarded_port", guest: 3306, host: 3306, auto_correct: true

    conf.vm.provider "virtualbox" do |vb|
      vb.memory = "1024"
      vb.cpus = "2"
      vb.gui = false
      vb.customize ["modifyvm", :id, "--vram", "32"]
      vb.customize ["modifyvm", :id, "--accelerate3d", "off"]
      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
    conf.vm.provision "shell", inline: $os_update
    conf.vm.provision :shell, path: "bootstrap.sh"
  end
end

```

Save and write this file.

You can also download from:

```
$ curl -o Vagrantfile http://securityhardening.com/files/Vagrantfile_20200928.txt
```

2.4.2. bootstrap.sh

Inside of the kali-linux-vm directory, populate a new file with the exact name, `bootstrap.sh`. Case matters, all lowercase. See comment about downloading this file immediately preceding the code block. `bootstrap.sh` (include the shebang in your file: the first line with `#!/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'
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}';"

    systemctl enable mysql
    systemctl restart mysql
    sleep 2
}

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.

If you have issues with copying and pasting the above file because code blocks in PDFs always copy correctly [NOT!], you could use curl, i.e. Make sure the bootstrap.sh file ends up in the same directory as the Vagrantfile.

```
$ curl -o bootstrap.sh http://securityhardening.com/files/bootstrap_sh_20200928.txt
```

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 images and start the virtual machines. 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 prohibit-password` To: `PermitRootLogin yes` Meaning strip the comment out on the beginning of the line and alter `prohibit-password` to `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.

For the DVWA instance, I would first run 'vagrant status' to capture the name that vagrant is using for the running instance.

```
# vagrant status
```

Choose

```
Current machine states:
kali-linux-vagrant running (virtualbox)
dvwa-vagrant running (virtualbox)
```

This environment represents multiple VMs. The VMs are all listed above with their current state. For more information about a specific VM, run `vagrant status NAME`.

From there, log into the DVWA instance with:

```
$ vagrant ssh dvwa-vagrant
```

And then get the current IP address.

```
$ 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 test network used for this paper uses 172.20.156.0/24 as the network range [shown here in section 3]. 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.

3. Eyewitness

If you have not done so, go ahead and run `vagrant up` from inside of your directory containing the Vagrantfile.

Log into Kali Linux.

username:root

password:toor

From what I found this year, the software certificate has expired.

Run the following commands as the root user to update the Kali linux software certificate:

```
cd /root/  
curl -O https://archive.kali.org/archive-key.asc  
apt-key add ./archive-key.asc
```

In a terminal as the root user, run:

```
apt update  
apt upgrade -y
```

Log in again as the root user and run inside of a terminal:

```
apt install -y eyewitness
```

output:

```

root@kali-linux-vagrant:~# apt install -y eyewitness
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bsdmainutils cpp-8 cryptsetup-run dctrl-tools dh-python dkms easy-rsa enchant exfat-fuse exfat-utils finger
  firmware-qcom-media
  firmware-qcom-soc fonts-glyphicons-halflings gir1.2-clutter-gst-3.0 gir1.2-gtkclutter-1.0 gnome-brave-icon-theme
  gnome-colors-common gnome-orca
  gnome-shell-extension-easyscreencast gnome-theme-kali gnome-tweak-tool gtk2-engines-murrine hashcat-data
  libappstream-glib8 libasan5
  libbind9-161 libboost-atomic1.67.0 libboost-chrono1.67.0 libboost-date-time1.67.0 libboost-filesystem1.67.0
  libboost-iostreams1.67.0
  libboost-python1.67.0 libboost-regex1.67.0 libboost-system1.67.0 libboost-thread1.67.0 libbrlapi0.6 libcamel-1.2-
  62 libcapstone3 libcdio18
  libclang1-6.0 libclang1-7 libcodec2-0.8.1 libcroco3 libcupsfilters1 libcupsimage2 libdc1394-22 libdleyana-core-
  1.0-3 libdns1104 libdns1110
  libdvdread4 libebook-contacts-1.2-2 libecal-1.2-19 libedataserver-1.2-23 libedataserverui-1.2-2 libemu2
  libenchanted1c2a libevent-2.1-6
  libevent-core-2.1-6 libevent-openssl-2.1-6 libevent-pthreads-2.1-6 libexiv2-14 libffi-dev libfluidsynth1 libgeos-
  3.7.2 libgspell-1-1
  libgssdp-1.0-3 libgssglue1 libgtkmm-2.4-1v5 libgtksourceview2.0-0 libgtksourceview2.0-common libgupnp-1.0-4
  libgweather-3-15 libhavege1
  libhwloc5 libigdmm9 libilmbase23 libirs161 libisc1100 libisc1105 libisccc161 libisccfg163 libisl19 libjim0.77
  libjsoncpp1 liblinear3
  liblirc-client0 libllvm8 liblouis17 liblwres161 libmicrohttpd12 libmng1 libmozjs-60-0 libmpdec2 libmpx2
  libmusicbrainz5-2 libmusicbrainz5cc2v5
  libmysofa0 libncurses-dev libnfs12 libntfs-3g883 libobjc4 libomp-dev libopenexr23 libpgm-5.2-0 libphonenumbers7
  libpipewire-0.2-1
  libpkcs11-helper1 libplymouth4 libpocl2-common libpoppler82 libproj13 libprotobuf17 libpython2-dev libpython2-
  stdlib libpython2.7-dev
  libpython3.7 libpython3.7-dev libpython3.7-minimal libpython3.7-stdlib libqscintilla2-qt4-l10n libqscintilla2-
  qt5-13 libqt5concurrent5
  libqt5opengl5 libquvi-0.9-0.9.3 libquvi-scripts-0.9 libradare2-3.2 libre2-5 libruby2.5 libsane libsnmp30
  libssl1.0.2 libtagc0 libtinfo-dev
  libtracker-control-2.0-0 libtracker-miner-2.0-0 libtracker-sparql-2.0-0 libtsk13 libusbmuxd4 libvpx5
  libwirehark12 libwiretap9 libwscodecs2
  libwsutil10 libx264-155 libx265-165 libx265-176 libxcb-util0 libxcb-xf86dri0 libxdot4 libyara3 linux-headers-
  5.2.0-kali2-common
  linux-kbuild-5.2 llvm-6.0 llvm-6.0-dev llvm-6.0-runtime llvm-7 llvm-7-dev llvm-7-runtime lua-bitop lua-expat lua-
  json lua-socket mousetweaks
  ncal openjdk-8-jre opensc-pkcs11 openvpn python-fffi-backend python-pip-whl python-pkg-resources python-
  pylibemu python-setuptools
  python-six python-tk python2 python2-dev python2-minimal python2.7-dev python3-editor python3-entrypoints
  python3-gst-1.0
  python3-ipython-genutils python3-jeepney python3-jupyter-core python3-keyring python3-keyrings.alt python3-
  nbformat python3-pip python3-plotly
  python3-secretstorage python3-traitlets python3-wheel python3.7-minimal qdbus qdbus-qt5 qtcore4-l10n ruby-
  connection-pool ruby-did-you-mean
  ruby-molinillo ruby-net-http-persistent ruby-thor ruby2.5-dev ruby2.5-doc rwho rhod snmp testdisk tftp
  virtualbox-guest-dkms whois x11-apps
  xsltproc
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  python3-easyprocess python3-fuzzywuzzy python3-pyvirtualdisplay python3-selenium xvfb
The following NEW packages will be installed:
  eyewitness python3-easyprocess python3-fuzzywuzzy python3-pyvirtualdisplay python3-selenium xvfb
0 upgraded, 6 newly installed, 0 to remove and 15 not upgraded.
Need to get 1,417 kB/4,557 kB of archives.
After this operation, 10.3 MB of additional disk space will be used.

```

Once installed, run:

```
eyewitness -h
```

output:

```

#####
#                               EyeWitness                               #
#####
#                               FortyNorth Security - https://www.fortynorthsecurity.com #
#####

usage: EyeWitness.py [--web] [-f Filename] [-x Filename.xml] [--single Single URL] [--no-dns] [--timeout Timeout]
[--jitter # of Seconds]
                        [--delay # of Seconds] [--threads # of Threads] [--max-retries Max retries on a timeout] [-d
Directory Name]
                        [--results Hosts Per Page] [--no-prompt] [--user-agent User Agent] [--difference Difference
Threshold]
                        [--proxy-ip 127.0.0.1] [--proxy-port 8080] [--proxy-type socks5] [--show-selenium] [--resolve]
                        [--add-http-ports ADD_HTTP_PORTS] [--add-https-ports ADD_HTTPS_PORTS] [--only-ports
ONLY_PORTS] [--prepend-https]
                        [--selenium-log-path SELENIUM_LOG_PATH] [--resume ew.db]

EyeWitness is a tool used to capture screenshots from a list of URLs

Protocols:
  --web                HTTP Screenshot using Selenium

Input Options:
  -f Filename          Line-separated file containing URLs to capture
  -x Filename.xml      Nmap XML or .Nessus file
  --single Single URL  Single URL/Host to capture
  --no-dns             Skip DNS resolution when connecting to websites

Timing Options:
  --timeout Timeout    Maximum number of seconds to wait while requesting a web page (Default: 7)
  --jitter # of Seconds Randomize URLs and add a random delay between requests
  --delay # of Seconds Delay between the opening of the navigator and taking the screenshot
  --threads # of Threads Number of threads to use while using file based input
  --max-retries Max retries on a timeout
                        Max retries on timeouts

Report Output Options:
  -d Directory Name    Directory name for report output
  --results Hosts Per Page Number of Hosts per page of report
  --no-prompt          Don't prompt to open the report

Web Options:
  --user-agent User Agent User Agent to use for all requests
  --difference Difference Threshold Difference threshold when determining if user agent requests are close "enough" (Default:
50)
  --proxy-ip 127.0.0.1 IP of web proxy to go through
  --proxy-port 8080     Port of web proxy to go through
  --proxy-type socks5   Proxy type (socks5/http)
  --show-selenium       Show display for selenium
  --resolve             Resolve IP/Hostname for targets
  --add-http-ports ADD_HTTP_PORTS Comma-separated additional port(s) to assume are http (e.g. '8018,8028')
  --add-https-ports ADD_HTTPS_PORTS Comma-separated additional port(s) to assume are https (e.g. '8018,8028')
  --only-ports ONLY_PORTS Comma-separated list of exclusive ports to use (e.g. '80,8080')
  --prepend-https       Prepend http:// and https:// to URLs without either
  --selenium-log-path SELENIUM_LOG_PATH Selenium geckodriver log path

Resume Options:
  --resume ew.db        Path to db file if you want to resume

```

We need to target IP address for the DVWA we installed.

Open a terminal on your laptop/workstation running VirtualBox and Vagrant and navigate to the folder holding the Vagrantfile. Run:

```
vagrant status
```

output:

```
vagrant status
Current machine states:

kali-linux-vagrant    running (virtualbox)
dwwa-vagrant          running (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run `vagrant status NAME`.
```

We need to SSH into the DVWA node, run in the same terminal as you ran the `vagrant status`:

```
vagrant ssh dwwa-vagrant -c "ip a"
```

output:

```
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 02:95:cf:22:ea:76 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::95:cfff:fe22:ea76/64 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 08:00:27:51:02:58 brd ff:ff:ff:ff:ff:ff
    inet 172.20.156.79/24 brd 172.20.156.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe51:258/64 scope link
        valid_lft forever preferred_lft forever
Connection to 127.0.0.1 closed.
```

In this case, we want to use the ip `172.20.156.79` as my target URL for eyewitness.



Username

Password

Login

[Damn Vulnerable Web Application \(DVWA\)](#)

username: admin password: admin



Setup DVWA

Instructions

About

Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: `/var/www/html/config/config.inc.php`

If the database already exists, **it will be cleared and the data will be reset.**
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

Setup Check

Web Server SERVER_NAME:

Operating system: ***nix**

PHP version: **7.0.33-0ubuntu0.16.04.16**
PHP function display_errors: **Enabled** (Easy Mode!)
PHP function safe_mode: **Disabled**
PHP function allow_url_include: **Enabled**
PHP function allow_url_fopen: **Enabled**
PHP function magic_quotes_gpc: **Disabled**
PHP module gd: **Installed**
PHP module mysql: **Installed**
PHP module pdo_mysql: **Installed**

Backend database: **MySQL/MariaDB**
Database username: **dvwa**
Database password: *********
Database database: **dvwa**
Database host: **127.0.0.1**
Database port: **3306**

reCAPTCHA key: **u8392ihj32ki8hujalkshuil32**

[User: www-data] Writable folder /var/www/html/hackable/uploads/: **Yes**
[User: www-data] Writable file /var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: **Yes**

[User: www-data] Writable folder /var/www/html/config: **Yes**
Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your php.ini file and restart Apache.

```
allow_url_fopen = On
allow_url_include = On
```

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

Create / Reset Database

From the Admin Page, click on the button at the bottom of the page labeled, "Create / Reset Database".

Log in again.

172.20.156.79/index.php

DVWA

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as [VirtualBox](#) or [VMware](#)), which is set to NAT networking mode. Inside a guest machine, you can download and install [XAMPP](#) for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want more difficult challenges, you may wish to look into the following other projects:

3.1. EyeWitness run

Let's start our sequence to capture what DVWA provides from the Kali Linux server. here, the IP Adress is the DVWA home page that was setup with this lab. change the ip address `172.20.156.79` to match your own. as always, on Kali, running on root. Open a Terminal and run:

```
cd /root
echo "http://172.20.156.79/index.php" > urls.txt

eyewitness -f /root/urls.txt
```

output:

```

#####
#                               EyeWitness                               #
#####
#                               FortyNorth Security - https://www.fortynorthsecurity.com #
#####

Starting Web Requests (1 Hosts)
Attempting to screenshot http://172.20.156.79/index.php
Finished in 5.446681261062622 seconds

[*] Done! Report written in the /root/2021-11-07_123714 folder!
Would you like to open the report now? [Y/n]
y
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
M                                                                 M
M      ."cCCc".                                                                 M
M      /ccccccc\              Our Upcoming Trainings:                                                                 M
M      $ccccccc|                                                                 M
M      :cccccccP              BlackHat USA >> Aug 1 - Aug 4 2020                                                    M
M      \ccccccc()              Virtual                                                                                M
M      \cccccccD              https://www.blackhat.com                                                                M
M      |ccccccc\              _                                                                 M
M      |ccccccc|              //              BH ASIA >> Sep 28 - Oct 2 2020                                        M
M      |ccccccc|=             //              Singapore                                                                 M
M      /°°°°°°°°"-             (CCCC)                                                                                M
M      ;----.-_ -.-_ |cccc|                                                                 M
M      .*° °° °° °° °° °° °° °° °° \cccc/                                                                 M
M      / /      ( )/ccc/                                                                 M
M      |_/      | _ °cccc|                                                                 M
M      |/_      °^^^°ccccccc/                                                                 M
M      /      \ccccccc/                                                                 M
M      /      \ccccccc/                                                                 M
M      |      °*°                                                                 M
M      /      \              Psss. Follow us on >> Twitter                                                         M
M      °*-.-----.-*°° °° °° °° °° °° °° °° °° °° °° °° °° °° °° °° °° °° °° °° °° >> Facebook M
M      \wwwwwwwwwwwwwwww/                                                                 >> LinkedIn M
M      \wwwwwwwwwwwwwwww/                                                                 M
MMMMMM | wwwwwwwwwwwwwwwww | MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM

```

This created a new dir under /root called, 2021-11-07_123714

With the following files:

```

root@kali-linux-vagrant:~# ls -lR 2021-11-07_123714/
2021-11-07_123714/:
total 136
-rw-r--r-- 1 root root 16384 Nov  7 12:37 ew.db
-rw-r--r-- 1 root root 95957 Oct 25 12:36 jquery-1.11.3.min.js
-rw-r--r-- 1 root root   40 Nov  7 12:37 open_ports.csv
-rw-r--r-- 1 root root 2931 Nov  7 12:37 report.html
-rw-r--r-- 1 root root 226 Nov  7 12:37 Requests.csv
drwxr-xr-x 2 root root 4096 Nov  7 12:37 screens
drwxr-xr-x 2 root root 4096 Nov  7 12:37 source
-rw-r--r-- 1 root root 684 Nov  7 12:37 style.css

2021-11-07_123714/screens:
total 24
-rw-r--r-- 1 root root 23933 Nov  7 12:37 http.172.20.156.79.index.php.png

2021-11-07_123714/source:
total 4
-rw-r--r-- 1 root root 1283 Nov  7 12:37 http.172.20.156.79.index.php.txt

```



Username

Password

Login

[Damn Vulnerable Web Application \(DVWA\)](#)

output from the source file, `http.172.20.156.79.index.php.txt`

```

<html lang="en-GB"><head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Login :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*</title>
  <link rel="stylesheet" type="text/css" href="dvwa/css/login.css">
</head>
<body>
<div id="wrapper">
<div id="header">
<br>
<p></p>
<br>
</div> <!--<div id="header">-->
<div id="content">
<form action="login.php" method="post">
<fieldset>
  <label for="user">Username</label> <input type="text" class="loginInput" size="20" name="username"><br>
  <label for="pass">Password</label> <input type="password" class="loginInput" autocomplete="off"
size="20" name="password"><br>
  <br>
  <p class="submit"><input type="submit" value="Login" name="Login"></p>
</fieldset>
<input type="hidden" name="user_token" value="b163768b3a61fa868bd8813a6d8ad3aa">
</form>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<!--  -->
</div> <!--<div id="content">-->
<div id="footer">
<p><a href="https://github.com/digininja/DVWA/" target="_blank">Damn Vulnerable Web Application (DVWA)</a></p>
</div> <!--<div id="footer"> -->
</div> <!--<div id="wrapper"> -->
</body></html>

```

The eyewitness report.html page:


Table of Contents

• [Uncategorized \(Page 1\)](#)

Uncategorized	1
Errors	0
Total	1

Report Generated on 2021/11/07 at 12:37:14

Uncategorized

Web Request Info	Web Screenshot
<p>http://172.20.156.79/index.php</p> <p>Page Title: Login :: Damn Vulnerable Web Application (DVWA) v1.10 "Development"</p> <p>Server: nginx/1.16.1</p> <p>Date: Sun, 07 Nov 2021 17:37:18 GMT</p> <p>Content-Type: text/html; charset=utf-8</p> <p>Transfer-Encoding: chunked</p> <p>Connection: close</p> <p>Set-Cookie: PHPSESSID=met0epkd355aadic1e0u9d0t73; path=/</p> <p>Pragma: no-cache</p> <p>Cache-Control: no-cache, must-revalidate</p> <p>Expires: Tue, 23 Jun 2009 12:00:00 GMT</p> <p>Response Code: 200</p> <p>Source Code</p>	

4. Conclusion

From this tutorial/how-to we have installed Kali Linux, DVWA and launched a capture session with the tool EyeWitness to gather information about our target, DVWA. This is pretty straight forward and as I am thinking about this, this would be very simple to automate and put into a cronjob to capture daily or weekly changes to a set of web sites and store then inside of an archive.

A launch script <not tested> to run from inside of cron.

```
#!/usr/bin/env bash

DATE=$( date +%Y%m%dT%H%M )

/usr/bin/eyewitness \
  -f /home/awesomeuser1/urls.txt \
  --web \
  --timeout 30 \
  --threads 15 \
  --max-retries 5
  -d /home/awesomeuser1/archives/${DATE}/ \
  --no-prompt
```

This paper has been a positive adventure when using **eyewitness** as a tool. I plan on using this tool more to archive artifacts for security related scans of sites and in general just to know if someone changed something, I will have evidence to prove my point.

5. Appendix

References

<https://www.kali.org/tools/eyewitness/>

<https://www.christophertruncer.com/eyewitness-triage-tool/>

<https://www.slideshare.net/CTruncer/eyewitness>

<https://github.com/ChrisTruncer/EyeWitness>