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

How to securely isolate and execute Nuclei from Kali Linux

Version 0.1, Last Updated: 13th June, 2023

Table of Contents

Chapter 1. Introduction

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

Chapter 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.

NOTE Update vagrant vm: `vagrant box update`

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 "dvwa-vagrant" do |conf|

        conf.vm.box = "ubuntu/xenial64"

        conf.vm.hostname = "dvwa-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 prothibit-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.

Chapter 3. Nuclei

Nuclei is a configurable scanning tool that uses templates. Offering extensibility while reportadely having zero false positives.

Source Code for Nuclei: https://github.com/projectdiscovery/nuclei

Official Documentation: https://docs.nuclei.sh/getting-started/overview

A "fast scanner used to scan across the modern applications, infrastructure, cloud environments, and networks to help you find and remediate vulnerabilities. Under the hood, it operates on the concept of templates, which are essentially simple YAML file that describe how to detect, prioritize and remediate specific security vulnerabilities. .... Each template represents a potential attack vector and includes a detailed description of the vulnerability, its severity, priority score, and sometimes even trending exploits. The template-driven approach not only adds a high degree of flexibility but also ensures that the vulnerabilities detected by Nuclei are not just theoretical risks but are indicative of real-world exploitability." — Nuclei docs

3.1. install nuclei

Inside of Kali Linux, in a terminal, run the command as the root user:

```
apt install nuclei
```

Answer yes with a `y` to the question that asks if you are sure you want to install the software for `nuclei`.

3.2. basics with nuclei

As always, it helps to read the help page before we start:

```
┌──(root㉿kali-linux-vagrant)-[~]
└─# nuclei --help
Nuclei is a fast, template based vulnerability scanner focusing
on extensive configurability, massive extensibility and ease of use.

Usage:
  nuclei [flags]

Flags:
TARGET:
   -u, -target string[]       target URLs/hosts to scan
   -l, -list string           path to file containing a list of target URLs/hosts to
scan (one per line)
   -resume string             resume scan using resume.cfg (clustering will be
disabled)
   -sa, -scan-all-ips         scan all the IP's associated with dns record
```

```
   -iv, -ip-version string[]  IP version to scan of hostname (4,6) - (default 4)

TEMPLATES:
   -nt, -new-templates                   run only new templates added in latest
nuclei-templates release
   -ntv, -new-templates-version string[]  run new templates added in specific version
   -as, -automatic-scan                  automatic web scan using wappalyzer
technology detection to tags mapping
   -t, -templates string[]               list of template or template directory to
run (comma-separated, file)
   -turl, -template-url string[]         template url or list containing template
urls to run (comma-separated, file)
   -w, -workflows string[]               list of workflow or workflow directory to
run (comma-separated, file)
   -wurl, -workflow-url string[]         workflow url or list containing workflow
urls to run (comma-separated, file)
   -validate                             validate the passed templates to nuclei
   -nss, -no-strict-syntax               disable strict syntax check on templates
   -td, -template-display                displays the templates content
   -tl                                   list all available templates

FILTERING:
   -a, -author string[]              templates to run based on authors (comma-
separated, file)
   -tags string[]                    templates to run based on tags (comma-separated,
file)
   -etags, -exclude-tags string[]    templates to exclude based on tags (comma-
separated, file)
   -itags, -include-tags string[]    tags to be executed even if they are excluded
either by default or configuration
   -id, -template-id string[]        templates to run based on template ids (comma-
separated, file, allow-wildcard)
   -eid, -exclude-id string[]        templates to exclude based on template ids
(comma-separated, file)
   -it, -include-templates string[]  templates to be executed even if they are
excluded either by default or configuration
   -et, -exclude-templates string[]  template or template directory to exclude
(comma-separated, file)
   -em, -exclude-matchers string[]   template matchers to exclude in result
   -s, -severity value[]             templates to run based on severity. Possible
values: info, low, medium, high, critical, unknown
   -es, -exclude-severity value[]    templates to exclude based on severity. Possible
values: info, low, medium, high, critical, unknown
   -pt, -type value[]                templates to run based on protocol type.
Possible values: dns, file, http, headless, tcp, workflow, ssl, websocket, whois
   -ept, -exclude-type value[]       templates to exclude based on protocol type.
Possible values: dns, file, http, headless, tcp, workflow, ssl, websocket, whois
   -tc, -template-condition string[]  templates to run based on expression condition

OUTPUT:
   -o, -output string            output file to write found issues/vulnerabilities
```

```
   -sresp, -store-resp           store all request/response passed through nuclei to
output directory
   -srd, -store-resp-dir string  store all request/response passed through nuclei to
custom directory (default "output")
   -silent                       display findings only
   -nc, -no-color                disable output content coloring (ANSI escape codes)
   -j, -jsonl                    write output in JSONL(ines) format
   -irr, -include-rr -omit-raw   include request/response pairs in the JSON, JSONL,
and Markdown outputs (for findings only) [DEPRECATED use -omit-raw] (default true)
   -or, -omit-raw                omit request/response pairs in the JSON, JSONL, and
Markdown outputs (for findings only)
   -nm, -no-meta                 disable printing result metadata in cli output
   -ts, -timestamp               enables printing timestamp in cli output
   -rdb, -report-db string       nuclei reporting database (always use this to persist
report data)
   -ms, -matcher-status          display match failure status
   -me, -markdown-export string  directory to export results in markdown format
   -se, -sarif-export string     file to export results in SARIF format
   -je, -json-export string      file to export results in JSON format
   -jle, -jsonl-export string    file to export results in JSONL(ine) format

CONFIGURATIONS:
   -config string                        path to the nuclei configuration file
   -fr, -follow-redirects                enable following redirects for http templates
   -fhr, -follow-host-redirects          follow redirects on the same host
   -mr, -max-redirects int               max number of redirects to follow for http
templates (default 10)
   -dr, -disable-redirects               disable redirects for http templates
   -rc, -report-config string            nuclei reporting module configuration file
   -H, -header string[]                  custom header/cookie to include in all http
request in header:value format (cli, file)
   -V, -var value                        custom vars in key=value format
   -r, -resolvers string                 file containing resolver list for nuclei
   -sr, -system-resolvers                use system DNS resolving as error fallback
   -dc, -disable-clustering              disable clustering of requests
   -passive                              enable passive HTTP response processing mode
   -fh2, -force-http2                    force http2 connection on requests
   -ev, -env-vars                        enable environment variables to be used in
template
   -cc, -client-cert string              client certificate file (PEM-encoded) used
for authenticating against scanned hosts
   -ck, -client-key string               client key file (PEM-encoded) used for
authenticating against scanned hosts
   -ca, -client-ca string                client certificate authority file (PEM-
encoded) used for authenticating against scanned hosts
   -sml, -show-match-line                show match lines for file templates, works
with extractors only
   -ztls                                 use ztls library with autofallback to
standard one for tls13 [Deprecated] autofallback to ztls is enabled by default
   -sni string                           tls sni hostname to use (default: input
domain name)
```

```
   -lfa, -allow-local-file-access        allows file (payload) access anywhere on the
system
   -lna, -restrict-local-network-access  blocks connections to the local / private
network
   -i, -interface string                 network interface to use for network scan
   -at, -attack-type string              type of payload combinations to perform
(batteringram,pitchfork,clusterbomb)
   -sip, -source-ip string               source ip address to use for network scan
   -config-directory string              override the default config path
($home/.config)
   -rsr, -response-size-read int         max response size to read in bytes (default
10485760)
   -rss, -response-size-save int         max response size to read in bytes (default
1048576)
   -reset                                reset removes all nuclei configuration and
data files (including nuclei-templates)
   -tlsi, -tls-impersonate               enable experimental client hello (ja3) tls
randomization

INTERACTSH:
   -iserver, -interactsh-server string  interactsh server url for self-hosted instance
(default: oast.pro,oast.live,oast.site,oast.online,oast.fun,oast.me)
   -itoken, -interactsh-token string    authentication token for self-hosted
interactsh server
   -interactions-cache-size int         number of requests to keep in the interactions
cache (default 5000)
   -interactions-eviction int           number of seconds to wait before evicting
requests from cache (default 60)
   -interactions-poll-duration int      number of seconds to wait before each
interaction poll request (default 5)
   -interactions-cooldown-period int    extra time for interaction polling before
exiting (default 5)
   -ni, -no-interactsh                  disable interactsh server for OAST testing,
exclude OAST based templates

FUZZING:
   -ft, -fuzzing-type string  overrides fuzzing type set in template (replace, prefix,
postfix, infix)
   -fm, -fuzzing-mode string  overrides fuzzing mode set in template (multiple,
single)

UNCOVER:
   -uc, -uncover                   enable uncover engine
   -uq, -uncover-query string[]    uncover search query
   -ue, -uncover-engine string[]  uncover search engine (shodan,censys,fofa,shodan-
idb,quake,hunter,zoomeye,netlas,criminalip,publicwww,hunterhow) (default shodan)
   -uf, -uncover-field string     uncover fields to return (ip,port,host) (default
"ip:port")
   -ul, -uncover-limit int        uncover results to return (default 100)
   -ur, -uncover-ratelimit int    override ratelimit of engines with unknown ratelimit
(default 60 req/min) (default 60)
```

```
RATE-LIMIT:
   -rl, -rate-limit int              maximum number of requests to send per second
(default 150)
   -rlm, -rate-limit-minute int      maximum number of requests to send per minute
   -bs, -bulk-size int               maximum number of hosts to be analyzed in
parallel per template (default 25)
   -c, -concurrency int              maximum number of templates to be executed in
parallel (default 25)
   -hbs, -headless-bulk-size int     maximum number of headless hosts to be analyzed
in parallel per template (default 10)
   -headc, -headless-concurrency int  maximum number of headless templates to be
executed in parallel (default 10)

OPTIMIZATIONS:
   -timeout int                      time to wait in seconds before timeout (default
10)
   -retries int                      number of times to retry a failed request (default
1)
   -ldp, -leave-default-ports        leave default HTTP/HTTPS ports (eg.
host:80,host:443)
   -mhe, -max-host-error int         max errors for a host before skipping from scan
(default 30)
   -te, -track-error string[]        adds given error to max-host-error watchlist
(standard, file)
   -nmhe, -no-mhe                    disable skipping host from scan based on errors
   -project                          use a project folder to avoid sending same request
multiple times
   -project-path string              set a specific project path (default "/tmp")
   -spm, -stop-at-first-match        stop processing HTTP requests after the first
match (may break template/workflow logic)
   -stream                           stream mode - start elaborating without sorting
the input
   -ss, -scan-strategy value         strategy to use while scanning(auto/host-
spray/template-spray) (default auto)
   -irt, -input-read-timeout value   timeout on input read (default 3m0s)
   -nh, -no-httpx                    disable httpx probing for non-url input
   -no-stdin                         disable stdin processing

HEADLESS:
   -headless                         enable templates that require headless browser
support (root user on Linux will disable sandbox)
   -page-timeout int                 seconds to wait for each page in headless mode
(default 20)
   -sb, -show-browser                show the browser on the screen when running
templates with headless mode
   -ho, -headless-options string[]   start headless chrome with additional options
   -sc, -system-chrome               use local installed Chrome browser instead of
nuclei installed
   -lha, -list-headless-action       list available headless actions
```

```
DEBUG:
   -debug                    show all requests and responses
   -dreq, -debug-req         show all sent requests
   -dresp, -debug-resp       show all received responses
   -p, -proxy string[]       list of http/socks5 proxy to use (comma separated or file
input)
   -pi, -proxy-internal      proxy all internal requests
   -ldf, -list-dsl-function  list all supported DSL function signatures
   -tlog, -trace-log string  file to write sent requests trace log
   -elog, -error-log string  file to write sent requests error log
   -version                  show nuclei version
   -hm, -hang-monitor        enable nuclei hang monitoring
   -v, -verbose              show verbose output
   -profile-mem string       optional nuclei memory profile dump file
   -vv                       display templates loaded for scan
   -svd, -show-var-dump      show variables dump for debugging
   -ep, -enable-pprof        enable pprof debugging server
   -tv, -templates-version   shows the version of the installed nuclei-templates
   -hc, -health-check        run diagnostic check up

UPDATE:
   -ut, -update-templates            update nuclei-templates to latest released
version
   -ud, -update-template-dir string  custom directory to install / update nuclei-
templates
   -duc, -disable-update-check       disable automatic nuclei/templates update check

STATISTICS:
   -stats                    display statistics about the running scan
   -sj, -stats-json          display statistics in JSONL(ines) format
   -si, -stats-interval int  number of seconds to wait between showing a statistics
update (default 5)
   -m, -metrics              expose nuclei metrics on a port
   -mp, -metrics-port int    port to expose nuclei metrics on (default 9092)

CLOUD:
   -cloud                             run scan on nuclei cloud
   -ads, -add-datasource string       add specified data source (s3,github)
   -atr, -add-target string           add target(s) to cloud
   -atm, -add-template string         add template(s) to cloud
   -lsn, -list-scan                   list previous cloud scans
   -lso, -list-output string          list scan output by scan id
   -ltr, -list-target                 list cloud target by id
   -ltm, -list-template               list cloud template by id
   -lds, -list-datasource             list cloud datasource by id
   -lrs, -list-reportsource           list reporting sources
   -dsn, -delete-scan string          delete cloud scan by id
   -dtr, -delete-target string        delete target(s) from cloud
   -dtm, -delete-template string      delete template(s) from cloud
   -dds, -delete-datasource string    delete specified data source
   -drs, -disable-reportsource string  disable specified reporting source
```

```
   -ers, -enable-reportsource string   enable specified reporting source
   -gtr, -get-target string            get target content by id
   -gtm, -get-template string          get template content by id
   -nos, -no-store                     disable scan/output storage on cloud
   -no-tables                          do not display pretty-printed tables
   -limit int                          limit the number of output to display (default
100)
```

3.3. Updated Templates

Let's start with an update for the templates:

```
┌────(root㉿kali-linux-vagrant)-[~]
└──# nuclei -ut


                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.9.14

        projectdiscovery.io

[INF] nuclei-templates are not installed, installing...
[INF] Successfully installed nuclei-templates at /root/.local/nuclei-templates
[INF] Your current nuclei-templates v9.6.2 are outdated. Latest is v9.6.2
[INF] Successfully updated nuclei-templates (v9.6.2) to /root/.local/nuclei-templates.
GoodLuck!
```

3.4. Starting Analysis

Let's look at the base page on port 80, which should be just the Apache service:

command in the form:

```
nuclei -u http://<target> -as
```

with `-u` meaning our target URL.

with `-as` from the above help command meaning: automatic web scan using wappalyzer technology detection to tags mapping.

```
┌────(root㉿kali-linux-vagrant)-[~]
└──# nuclei -u http://192.168.99.19 -as


                     __     _
```

```
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.9.14

        projectdiscovery.io

[INF] Your current nuclei-templates v9.6.2 are outdated. Latest is v9.6.2
[INF] Successfully updated nuclei-templates (v9.6.2) to /root/.local/nuclei-templates.
GoodLuck!
[INF] Current nuclei version: v2.9.14 (outdated)
[INF] Current nuclei-templates version: v9.6.2 (outdated)
[INF] New templates added in latest release: 61
[INF] Templates loaded for current scan: 6719
[INF] Targets loaded for current scan: 1
[INF] Executing wappalyzer based tech detection on input urls
[INF] Executing tags (debian,apache,http,server) for host &{http://192.168.99.19 }
(163 templates)
[apache-detect] [http] [info] http://192.168.99.19 [Apache/2.4.10 (Debian)]
[default-apache-test-all] [http] [info] http://192.168.99.19 [Apache/2.4.10 (Debian)]
[default-apache2-page] [http] [info] http://192.168.99.19
[INF] Using Interactsh Server: oast.fun
```

A lot is going on here. The basic Apache page really doesn't have anything to offer.

Let us pivot over to the DVWA site

```
┌──(root㉿kali-linux-vagrant)-[~]
└─# nuclei -u http://192.168.99.19/dvwa/


                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.9.14

        projectdiscovery.io

[INF] Your current nuclei-templates v9.6.2 are outdated. Latest is v9.6.2
[INF] Successfully updated nuclei-templates (v9.6.2) to /root/.local/nuclei-templates.
GoodLuck!
[INF] Current nuclei version: v2.9.14 (outdated)
[INF] Current nuclei-templates version: v9.6.2 (outdated)
[INF] New templates added in latest release: 61
[INF] Templates loaded for current scan: 6719
[INF] Targets loaded for current scan: 1
[INF] Templates clustered: 1193 (Reduced 1132 Requests)
[httponly-cookie-detect] [http] [info] http://192.168.99.19/dvwa/
[INF] Using Interactsh Server: oast.live
[openssh-detect] [tcp] [info] 192.168.99.19:22 [SSH-2.0-OpenSSH_6.7p1 Debian-5+deb8u3]
```

```
[fingerprinthub-web-fingerprints:dvwa] [http] [info]
http://192.168.99.19/dvwa/login.php
[tech-detect:php] [http] [info] http://192.168.99.19/dvwa/login.php
[tech-detect:php] [http] [info] http://192.168.99.19/dvwa/
[http-missing-security-headers:permissions-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:referrer-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:cross-origin-resource-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:x-content-type-options] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:clear-site-data] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:cross-origin-opener-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:strict-transport-security] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:content-security-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:x-frame-options] [http] [info]
http://192.168.99.19/dvwa/login.php
[dvwa-default-login] [http] [critical] http://192.168.99.19/dvwa/index.php
[username="admin",password="password"]
[waf-detect:apachegeneric] [http] [info] http://192.168.99.19/dvwa/
[rpcbind-portmapper-detect] [tcp] [info] 192.168.99.19:111
```

Much more interesting!

We see here towards the bottom that an insecure password was setup. Tsssk! Tsssk! Such poor Governance of a vulnerabile resource.

We can load more targets into a single file, e.g. `domains.txt` and then run `nuclei` against a list of targets.

```
┌──(root㉿kali-linux-vagrant)-[~]
└─# printf "192.168.99.19\n192.168.99.19/dvwa/\n" > domains.txt
```

and then to run the analysis with the `-l` option:

```
┌──(root㉿kali-linux-vagrant)-[~]
└─# nuclei -l ./domains.txt

                     __     _
   ____  __  _______/ /__  (_)
```

```
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.9.14

        projectdiscovery.io

[INF] Your current nuclei-templates v9.6.2 are outdated. Latest is v9.6.2
[INF] Successfully updated nuclei-templates (v9.6.2) to /root/.local/nuclei-templates.
GoodLuck!
[INF] Current nuclei version: v2.9.14 (outdated)
[INF] Current nuclei-templates version: v9.6.2 (outdated)
[INF] New templates added in latest release: 61
[INF] Templates loaded for current scan: 6719
[INF] Targets loaded for current scan: 2
[INF] Running httpx on input host
[INF] Found 2 URL from httpx
[INF] Templates clustered: 1189 (Reduced 2260 Requests)
[httponly-cookie-detect] [http] [info] http://192.168.99.19/dvwa/
[fingerprinthub-web-fingerprints:dvwa] [http] [info]
http://192.168.99.19/dvwa/login.php
[default-apache-test-all] [http] [info] http://192.168.99.19 [Apache/2.4.10 (Debian)]
[tech-detect:php] [http] [info] http://192.168.99.19/dvwa/login.php
[apache-detect] [http] [info] http://192.168.99.19 [Apache/2.4.10 (Debian)]
[default-apache2-page] [http] [info] http://192.168.99.19
[tech-detect:php] [http] [info] http://192.168.99.19/dvwa/
[INF] Using Interactsh Server: oast.fun
[rpcbind-portmapper-detect] [tcp] [info] 192.168.99.19:111
[rpcbind-portmapper-detect] [tcp] [info] 192.168.99.19:111
[dns-saas-service-detection] [dns] [info] 192.168.99.19/dvwa/
[openssh-detect] [tcp] [info] 192.168.99.19:22 [SSH-2.0-OpenSSH_6.7p1 Debian-5+deb8u3]
[openssh-detect] [tcp] [info] 192.168.99.19:22 [SSH-2.0-OpenSSH_6.7p1 Debian-5+deb8u3]
[options-method] [http] [info] http://192.168.99.19 [GET,HEAD,POST,OPTIONS]
[http-missing-security-headers:content-security-policy] [http] [info]
http://192.168.99.19
[http-missing-security-headers:permissions-policy] [http] [info] http://192.168.99.19
[http-missing-security-headers:referrer-policy] [http] [info] http://192.168.99.19
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info]
http://192.168.99.19
[http-missing-security-headers:cross-origin-opener-policy] [http] [info]
http://192.168.99.19
[http-missing-security-headers:cross-origin-resource-policy] [http] [info]
http://192.168.99.19
[http-missing-security-headers:strict-transport-security] [http] [info]
http://192.168.99.19
[http-missing-security-headers:x-frame-options] [http] [info] http://192.168.99.19
[http-missing-security-headers:x-content-type-options] [http] [info]
http://192.168.99.19
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info]
http://192.168.99.19
[http-missing-security-headers:clear-site-data] [http] [info] http://192.168.99.19
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info]
```

```
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:content-security-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:permissions-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:clear-site-data] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:cross-origin-opener-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:cross-origin-resource-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:strict-transport-security] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:x-frame-options] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:x-content-type-options] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:referrer-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[waf-detect:apachegeneric] [http] [info] http://192.168.99.19/dvwa/
[waf-detect:apachegeneric] [http] [info] http://192.168.99.19/
[dvwa-default-login] [http] [critical] http://192.168.99.19/dvwa/index.php
[username="admin",password="password"]
```

This is getting much more interesting and I specifically like the reporting from `nuclei` with the entire URL composition to make it easier to cherry pick the meaningful targets out of said.

Using Automatic Selection

```
┌──(root㉿kali-linux-vagrant)-[~]
└─# nuclei -l ./domains.txt -as


                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.9.14

        projectdiscovery.io

[INF] Your current nuclei-templates v9.6.2 are outdated. Latest is v9.6.2
[INF] Successfully updated nuclei-templates (v9.6.2) to /root/.local/nuclei-templates.
GoodLuck!
[INF] Current nuclei version: v2.9.14 (outdated)
[INF] Current nuclei-templates version: v9.6.2 (outdated)
[INF] New templates added in latest release: 61
[INF] Templates loaded for current scan: 6719
[INF] Targets loaded for current scan: 2
```

```
[INF] Running httpx on input host
[INF] Found 2 URL from httpx
[INF] Executing wappalyzer based tech detection on input urls
[INF] No results found. Better luck next time!
```

I was expecting more using the Automatic Selection with this scan here.

The idea behind function is that it will attempt to fingerprint the technology stack and components used on the target, then select templates that have been tagged with those tech stack keywords.

I wanted to know what tags were available for the DVWA, so I ran:

```
┌──(root㉿kali-linux-vagrant)-[~/.local/nuclei-templates/http]
└─# grep -R dvwa *
default-logins/dvwa/dvwa-default-login.yaml:id: dvwa-default-login
default-logins/dvwa/dvwa-default-login.yaml:    - https://opensourcelibs.com/lib/dvwa
default-logins/dvwa/dvwa-default-login.yaml:  tags: dvwa,default-login
technologies/fingerprinthub-web-fingerprints.yaml:        name: dvwa
technologies/fingerprinthub-web-fingerprints.yaml:          - dvwa/css/login.css
technologies/fingerprinthub-web-fingerprints.yaml:        name: dvwa
technologies/fingerprinthub-web-fingerprints.yaml:          -
dvwa/images/login_logo.png
```

The next scan then looks like [to use the DVWA tag name we just discovered]:

```
┌──(root㉿kali-linux-vagrant)-[~]
└─# nuclei -l ./domains.txt -tags dvwa


                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.9.14

        projectdiscovery.io

[INF] Your current nuclei-templates v9.6.2 are outdated. Latest is v9.6.2
[INF] Successfully updated nuclei-templates (v9.6.2) to /root/.local/nuclei-templates.
GoodLuck!
[INF] Current nuclei version: v2.9.14 (outdated)
[INF] Current nuclei-templates version: v9.6.2 (outdated)
[INF] New templates added in latest release: 61
[INF] Templates loaded for current scan: 1
[INF] Targets loaded for current scan: 2
[INF] Running httpx on input host
[INF] Found 2 URL from httpx
[dvwa-default-login] [http] [critical] http://192.168.99.19/dvwa/index.php
[username="admin",password="password"]
```

Let us scan by Severity:

```
┌──(root㉿kali-linux-vagrant)-[~]
└─# nuclei -l ./domains.txt -s critical,high,medium,low,info


                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.9.14

        projectdiscovery.io

[INF] Your current nuclei-templates v9.6.2 are outdated. Latest is v9.6.2
[INF] Successfully updated nuclei-templates (v9.6.2) to /root/.local/nuclei-templates.
GoodLuck!
[INF] Current nuclei version: v2.9.14 (outdated)
[INF] Current nuclei-templates version: v9.6.2 (outdated)
[INF] New templates added in latest release: 61
[INF] Templates loaded for current scan: 6691
[INF] Targets loaded for current scan: 2
[INF] Running httpx on input host
[INF] Found 2 URL from httpx
[INF] Templates clustered: 1181 (Reduced 2240 Requests)
[httponly-cookie-detect] [http] [info] http://192.168.99.19/dvwa/
[default-apache-test-all] [http] [info] http://192.168.99.19 [Apache/2.4.10 (Debian)]
[default-apache2-page] [http] [info] http://192.168.99.19
[apache-detect] [http] [info] http://192.168.99.19 [Apache/2.4.10 (Debian)]
[fingerprinthub-web-fingerprints:dvwa] [http] [info]
http://192.168.99.19/dvwa/login.php
[tech-detect:php] [http] [info] http://192.168.99.19/dvwa/login.php
[tech-detect:php] [http] [info] http://192.168.99.19/dvwa/
[INF] Using Interactsh Server: oast.live
[http-missing-security-headers:x-content-type-options] [http] [info]
http://192.168.99.19
[http-missing-security-headers:referrer-policy] [http] [info] http://192.168.99.19
[http-missing-security-headers:clear-site-data] [http] [info] http://192.168.99.19
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info]
http://192.168.99.19
[http-missing-security-headers:cross-origin-opener-policy] [http] [info]
http://192.168.99.19
[http-missing-security-headers:cross-origin-resource-policy] [http] [info]
http://192.168.99.19
[http-missing-security-headers:strict-transport-security] [http] [info]
http://192.168.99.19
[http-missing-security-headers:content-security-policy] [http] [info]
http://192.168.99.19
[http-missing-security-headers:permissions-policy] [http] [info] http://192.168.99.19
[http-missing-security-headers:x-frame-options] [http] [info] http://192.168.99.19
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info]
```

```
http://192.168.99.19
[http-missing-security-headers:strict-transport-security] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:permissions-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:x-content-type-options] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:cross-origin-opener-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:content-security-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:x-frame-options] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:referrer-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:clear-site-data] [http] [info]
http://192.168.99.19/dvwa/login.php
[http-missing-security-headers:cross-origin-resource-policy] [http] [info]
http://192.168.99.19/dvwa/login.php
[dns-saas-service-detection] [dns] [info] 192.168.99.19/dvwa/
[options-method] [http] [info] http://192.168.99.19 [GET,HEAD,POST,OPTIONS]
[dvwa-default-login] [http] [critical] http://192.168.99.19/dvwa/index.php
[password="password",username="admin"]
[waf-detect:apachegeneric] [http] [info] http://192.168.99.19/dvwa/
[waf-detect:apachegeneric] [http] [info] http://192.168.99.19/
[rpcbind-portmapper-detect] [tcp] [info] 192.168.99.19:111
[rpcbind-portmapper-detect] [tcp] [info] 192.168.99.19:111
[openssh-detect] [tcp] [info] 192.168.99.19:22 [SSH-2.0-OpenSSH_6.7p1 Debian-5+deb8u3]
[openssh-detect] [tcp] [info] 192.168.99.19:22 [SSH-2.0-OpenSSH_6.7p1 Debian-5+deb8u3]
```

Whoa!

This is starting to expand in a good way [for knowledge sake].

Fuzzing

First, we need to clone the templates down to us.

```
┌──(root㉿kali-linux-vagrant)-[~]
└─# git clone https://github.com/projectdiscovery/fuzzing-templates.git
Cloning into 'fuzzing-templates'...
remote: Enumerating objects: 200, done.
remote: Counting objects: 100% (121/121), done.
remote: Compressing objects: 100% (80/80), done.
remote: Total 200 (delta 52), reused 67 (delta 33), pack-reused 79
Receiving objects: 100% (200/200), 60.09 KiB | 1.72 MiB/s, done.
```

```
Resolving deltas: 100% (67/67), done.
```

The why with the serparate fuzzing repository:

"Fuzzing improves security testing and uncovers bugs that mortal eyes would have often missed. But its thorough approach requires significant execution time. Thus, testing fuzzing templates against targets in Nuclei by default would increase the scanning time." — projectdiscovery.io

We need a special file for our targets.

```
cat <<'EOF' > fuzz_endpoints.txt
http://192.168.99.19/dvwa/info?name=test&another=value&random=data
http://192.168.99.19/dvwa/redirect?redirect_url=/info?name=redirected_from_url
http://192.168.99.19/dvwa/request?url=https://example.com
http://192.168.99.19/dvwa/email?text=important_user
http://192.168.99.19/dvwa/permissions?cmd=whoami
http://192.168.99.19/dvwa/info?name=redirected_from_url
EOF
```

To run the fuzzing scan:

```
┌──(root㉿kali-linux-vagrant)-[~]
└─# nuclei -t ./fuzzing-templates -list ./fuzz_endpoints.txt


                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.9.14

        projectdiscovery.io

[INF] Your current nuclei-templates v9.6.2 are outdated. Latest is v9.6.2
[INF] Successfully updated nuclei-templates (v9.6.2) to /root/.local/nuclei-templates.
GoodLuck!
[INF] Current nuclei version: v2.9.14 (outdated)
[INF] Current nuclei-templates version: v9.6.2 (outdated)
[INF] New templates added in latest release: 61
[INF] Templates loaded for current scan: 19
[INF] Targets loaded for current scan: 6
[INF] Using Interactsh Server: oast.me
[INF] No results found. Better luck next time!
```

That run was kind of a bust. I was hoping for more.

Chapter 4. Conclusion

In this paper, we reviewed Nuclei as a scanning tool for [web] services. Specifically targetting the Damn Vulnerable Web Application. The paper has focused on the basics of the tooling without jumping into the Advanced capabilities [e.g. building your custom templates, Extractors, and Request Conditions]. Beyond a developer or systems engineer doing something very stupid, like using the poor choice of `admin:password` for their username and password combination [this is 100% a critical finding], the tool didn't report immediately usable information that I can use to gain immediate access to the service without further research on exploits available [again, excluding the glaring critical finding]. The tool did find the weak username:password combination. It did print Missing Security Header information as well as Version numbers for further research for exploits. The tool did find other services, like SSH and RPCbind-portmapper. The onus is now on me to further research each of these potential weak points and find a solution to exploit the weakness on DVWA.

All of that stated, it was a pure pleasure digging into this tool and I foresee many years ahead using this tool for information collecting to help me discover weaknesses that need to be corrected/fixed with services that I setup in my Homelab and if my customer's allow in their environments, Pentesting their services. Always make sure you get approval from your customer before testing, and preferably in writing, such as an email from your work email documenting the customer's decision, documenting all of the steps you will be taking, including putting Kali Linux on their network.

Chapter 5. Appendix

References

Source Code: https://github.com/projectdiscovery/nuclei

Documentation: https://docs.nuclei.sh/getting-started/overview

This guide lead me through a lot of useful features in Nuclei: https://blog.projectdiscovery.io/ultimate-nuclei-guide/

DVWA gave me some grief this year with dependency issues for MySQL. I used the following as a quick work around for the problem:

```
# DVWA

mkdir vagrant-dvwa
cd vagrant-dvwa
vagrant init jaxmetalmax/dvwa-debian --box-version 1.0
```

Edit the Vagrantfile and insert the following lines to match your virtualbox NAT'ed network:

```
  config.vm.network "public_network", bridge: "wlo1", auto_config: true
```

AND

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

  config.vm.provider "virtualbox" do |vb|
    vb.memory = "1024"
    vb.cpus = "2"
    vb.gui = true
    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"]
  end
```

Finally, start the virtual machine for DVWA in the same folder you just created and modified:

```
vagrant up
```