How to securely isolate and execute Autorecon from Kali Linux

Version 0.1, Last Updated: 2024-07-13

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

Table of Contents

Chapter 1. Introduction

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

AutoRecon is an advanced, multi-threaded reconnaissance tool built for automating the enumeration of services in network environments. Designed primarily for penetration testers, Capture the Flag (CTF) participants, and security professionals, AutoRecon simplifies and speeds up the initial reconnaissance phase by performing automated scans and collecting valuable service data. Its primary use case is to reduce manual effort in environments such as the Offensive Security Certified Professional (OSCP) exam, but it is also applicable in real-world engagements where efficient and thorough service enumeration is critical. By streamlining this process, AutoRecon allows security professionals to focus on more complex aspects of penetration testing and vulnerability assessments.

AutoRecon offers several key features that make it a valuable tool for penetration testers and security professionals:

1. Multi-threaded Performance: AutoRecon leverages multi-threading to enhance speed and efficiency, enabling concurrent scanning of multiple services and reducing the time required for comprehensive network reconnaissance.
2. Automated Service Enumeration: The tool automates the enumeration of services on target systems, eliminating the need for manual intervention and ensuring that all open ports and services are identified and properly cataloged.
3. Customizable Scanning Capabilities: AutoRecon allows users to configure and customize scanning parameters to suit specific testing

environments or scenarios, offering flexibility in its application.

4. Integration with Popular Tools: It integrates seamlessly with widely-used security tools, further enriching its capabilities by enabling comprehensive information gathering and analysis through familiar toolsets.
5. Time-saving and Efficient: By automating routine reconnaissance tasks, AutoRecon saves valuable time, allowing penetration testers to focus on more in-depth testing, such as vulnerability analysis and exploitation.
6. Broad Applicability: Although primarily designed for use in Capture the Flag (CTF) competitions and certification exams like OSCP, AutoRecon's functionality is equally useful in real-world penetration testing and security assessments.

Let us begin our journey together.

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 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.

Chapter 3. Autorecon

3.1. Installation

Log in as the `root` user with the password, `toor`.

Open a terminal.

```
apt install -y autorecon
```

This process of installing autorecon and dependencies can take a while. The author's system needed 438 packages to install this one single package. One might consider getting a cup of coffee while this runs.

If you desire to clean up free space after this install, you might consider running the following command:

```
apt autoremove
```

3.2. Help system

```
└─# autorecon -h
usage: autorecon [-t TARGET_FILE] [-p PORTS] [-m MAX_SCANS]
                 [-mp MAX_PORT_SCANS] [-c CONFIG_FILE] [-g GLOBAL_FILE]
                 [--tags TAGS] [--exclude-tags TAGS]
                 [--port-scans PLUGINS] [--service-scans PLUGINS]
                 [--reports PLUGINS] [--plugins-dir PLUGINS_DIR]
                 [--add-plugins-dir PLUGINS_DIR] [-l [TYPE]]
                 [-o OUTPUT] [--single-target] [--only-scans-dir]
                 [--no-port-dirs] [--heartbeat HEARTBEAT] [--timeout TIMEOUT]
                 [--target-timeout TARGET_TIMEOUT]
                 [--nmap NMAP | --nmap-append NMAP_APPEND] [--proxychains]
                 [--disable-sanity-checks] [--disable-keyboard-control]
                 [--force-services SERVICE [SERVICE ...]]
                 [-mpti PLUGIN:NUMBER [PLUGIN:NUMBER ...]]
```

```
                [-mpgi PLUGIN:NUMBER [PLUGIN:NUMBER ...]] [--accessible]
                [-v] [--version] [--curl.path VALUE]
                [--dirbuster.tool {feroxbuster,gobuster,dirsearch,ffuf,dirb}]
                [--dirbuster.wordlist VALUE [VALUE ...]]
                [--dirbuster.threads VALUE] [--dirbuster.ext VALUE]
                [--dirbuster.recursive] [--dirbuster.extras VALUE]
                [--enum4linux.tool {enum4linux-ng,enum4linux}]
                [--onesixtyone.community-strings VALUE]
                [--subdomain-enum.domain VALUE]
                [--subdomain-enum.wordlist VALUE [VALUE ...]]
                [--subdomain-enum.threads VALUE] [--vhost-enum.hostname VALUE]
                [--vhost-enum.wordlist VALUE [VALUE ...]]
                [--vhost-enum.threads VALUE] [--wpscan.api-token VALUE]
                [--global.username-wordlist VALUE]
                [--global.password-wordlist VALUE] [--global.domain VALUE]
                [-h] [targets ...]

Network reconnaissance tool to port scan and automatically enumerate services
found on multiple targets.

positional arguments:
  targets               IP addresses (e.g. 10.0.0.1), CIDR notation
                        (e.g. 10.0.0.1/24), or resolvable hostnames
                        (e.g. foo.bar) to scan.

options:
  -t TARGET_FILE, --target-file TARGET_FILE
                        Read targets from file.
  -p PORTS, --ports PORTS
                        Comma separated list of ports / port ranges to scan.
                        Specify TCP/UDP ports by prepending list with T:/U:
                        To scan both TCP/UDP, put port(s) at start or specify
                        B: e.g. 53,T:21-25,80,U:123,B:123. Default: None
  -m MAX_SCANS, --max-scans MAX_SCANS
                        The maximum number of concurrent scans to run.
                        Default: 50
  -mp MAX_PORT_SCANS, --max-port-scans MAX_PORT_SCANS
                        The maximum number of concurrent port scans to run.
                        Default: 10 (approx 20% of max-scans unless specified)
  -c CONFIG_FILE, --config CONFIG_FILE
                        Location of AutoRecon's config file.
                        Default: /root/.config/AutoRecon/config.toml
  -g GLOBAL_FILE, --global-file GLOBAL_FILE
                        Location of AutoRecon's global file.
                        Default: /root/.config/AutoRecon/global.toml
  --tags TAGS           Tags to determine which plugins should be included.
                        Separate tags by a plus symbol (+) to group tags
                        together. Separate groups with a comma (,) to create
                        multiple groups. For a plugin to be included, it must
                        have all the tags specified in at least one group.
                        Default: default
  --exclude-tags TAGS   Tags to determine which plugins should be excluded.
                        Separate tags by a plus symbol (+) to group tags
```

```
                      together. Separate groups with a comma (,) to create
                      multiple groups. For a plugin to be excluded, it must
                      have all the tags specified in at least one group.
                      Default: None
--port-scans PLUGINS  Override --tags / --exclude-tags for the listed
                      PortScan plugins (comma separated). Default: None
--service-scans PLUGINS
                      Override --tags / --exclude-tags for the listed
                      ServiceScan plugins (comma separated). Default: None
--reports PLUGINS     Override --tags / --exclude-tags for the listed
                      Report plugins (comma separated). Default: None
--plugins-dir PLUGINS_DIR
                      The location of the plugins directory.
                      Default: /root/.local/share/AutoRecon/plugins
--add-plugins-dir PLUGINS_DIR
                      The location of an additional plugins directory to
                      add to the main one. Default: None
-l [TYPE], --list [TYPE]
                      List all plugins or plugins of a specific type.
                      e.g. --list, --list port, --list service
-o OUTPUT, --output OUTPUT
                      The output directory for results. Default: results
--single-target       Only scan a single target. A directory named after
                      the target will not be created. Instead, the directory
                      structure will be created within the output directory.
                      Default: False
--only-scans-dir      Only create the "scans" directory for results.
                      Other directories (e.g. exploit, loot, report) will
                      not be created. Default: False
--no-port-dirs        Don't create directories for ports (e.g. scans/tcp80,
                      scans/udp53). Instead store all results in the "scans"
                      directory itself. Default: False
--heartbeat HEARTBEAT Specifies the heartbeat interval (in seconds) for scan
                      status messages. Default: 60
--timeout TIMEOUT     Specifies the maximum amount of time in minutes that
                      AutoRecon should run for. Default: None
--target-timeout TARGET_TIMEOUT
                      Specifies the maximum amount of time in minutes that a
                      target should be scanned for before abandoning it and
                      moving on. Default: None
--nmap NMAP           Override the {nmap_extra} variable in scans.
                      Default: -vv --reason -Pn -T4
--nmap-append NMAP_APPEND
                      Append to the default {nmap_extra} variable in scans.
                      Default:
--proxychains         Use if you are running AutoRecon via proxychains.
                      Default: False
--disable-sanity-checks
                      Disable sanity checks that would otherwise prevent the
                      scans from running. Default: False
--disable-keyboard-control
                      Disables keyboard control ([s]tatus, Up, Down) if you
                      are in SSH or Docker.
```

```
  --force-services SERVICE [SERVICE ...]
                        A space separated list of services in the following
                        style: tcp/80/http tcp/443/https/secure
  -mpti PLUGIN:NUMBER [PLUGIN:NUMBER ...], --max-plugin-target-instances
                        PLUGIN:NUMBER [PLUGIN:NUMBER ...]
                        A space separated list of plugin slugs with the max
                        number of instances (per target) in the following
                        style: nmap-http:2 dirbuster:1. Default: None
  -mpgi PLUGIN:NUMBER [PLUGIN:NUMBER ...], --max-plugin-global-instances
                        PLUGIN:NUMBER [PLUGIN:NUMBER ...]
                        A space separated list of plugin slugs with the max
                        number of global instances in the following style:
                        nmap-http:2 dirbuster:1. Default: None
  --accessible          Attempts to make AutoRecon output more accessible to
                        screenreaders. Default: False
  -v, --verbose         Enable verbose output. Repeat for more verbosity.
  --version             Prints the AutoRecon version and exits.
  -h, --help            Show this help message and exit.

plugin arguments:
  These are optional arguments for certain plugins.

  --curl.path VALUE     The path on the web server to curl. Default: /
  --dirbuster.tool {feroxbuster,gobuster,dirsearch,ffuf,dirb}
                        The tool to use for directory busting.
                        Default: feroxbuster
  --dirbuster.wordlist VALUE [VALUE ...]
                        The wordlist(s) to use when directory busting.
                        Separate multiple wordlists with spaces.
                        Default: ['/root/.local/share/AutoRecon/wordlists/
                        dirbuster.txt']
  --dirbuster.threads VALUE
                        The number of threads to use when directory busting.
                        Default: 10
  --dirbuster.ext VALUE The extensions you wish to fuzz (no dot, comma
                        separated). Default: txt,html,php,asp,aspx,jsp
  --dirbuster.recursive Enables recursive searching (where available).
                        Warning: This may cause significant increases to scan
                        times. Default: False
  --dirbuster.extras

 VALUE
                        Any extra options you wish to pass to the tool when it
                        runs. e.g. --dirbuster.extras='-s 200,301 --discover-
                        backup'
  --enum4linux.tool {enum4linux-ng,enum4linux}
                        The tool to use for doing Windows and Samba
                        enumeration. Default: enum4linux-ng
  --onesixtyone.community-strings VALUE
                        The file containing a list of community strings to
                        try. Default: /usr/share/seclists/Discovery/SNMP/
                        common-snmp-community-strings-onesixtyone.txt
  --subdomain-enum.domain VALUE
```

```
                        The domain to use as the base domain (e.g.
                        example.com) for subdomain enumeration. Default: None
  --subdomain-enum.wordlist VALUE [VALUE ...]
                        The wordlist(s) to use when enumerating subdomains.
                        Separate multiple wordlists with spaces. Default:
                        ['/usr/share/seclists/Discovery/DNS/subdomains-
                        top1million-110000.txt']
  --subdomain-enum.threads VALUE
                        The number of threads to use when enumerating
                        subdomains. Default: 10
  --vhost-enum.hostname VALUE
                        The hostname to use as the base host (e.g.
                        example.com) for virtual host enumeration.
                        Default: None
  --vhost-enum.wordlist VALUE [VALUE ...]
                        The wordlist(s) to use when enumerating virtual hosts.
                        Separate multiple wordlists with spaces. Default:
                        ['/usr/share/seclists/Discovery/DNS/subdomains-
                        top1million-110000.txt']
  --vhost-enum.threads VALUE
                        The number of threads to use when enumerating virtual
                        hosts. Default: 10
  --wpscan.api-token VALUE
                        An API Token from wpvulndb.com to help search for more
                        vulnerabilities.

global plugin arguments:
  These are optional arguments that can be used by all plugins.

  --global.username-wordlist VALUE
                        A wordlist of usernames, useful for bruteforcing.
                        Default: /usr/share/seclists/Usernames/top-usernames-
                        shortlist.txt
  --global.password-wordlist VALUE
                        A wordlist of passwords, useful for bruteforcing.
                        Default: /usr/share/seclists/Passwords/darkweb2017-
                        top100.txt
  --global.domain VALUE The domain to use (if known). Used for DNS and/or
                        Active Directory. Default: None
```

3.3. To Scan our DVWA

At the moment, the IP address of my DVWA is running on `192.168.99.19`.

To launch the tool, we can: `autorecon <IP_Address_of_remote_target_system>`

```
┌──(root㉿kali-linux-vagrant)-[~]
└─# autorecon 192.168.99.19
[*] Scanning target 192.168.99.19
[*] [192.168.99.19/all-tcp-ports] Discovered open port tcp/111 on 192.168.99.19
[*] [192.168.99.19/all-tcp-ports] Discovered open port tcp/80 on 192.168.99.19
[*] [192.168.99.19/all-tcp-ports] Discovered open port tcp/22 on 192.168.99.19
[*] [192.168.99.19/all-tcp-ports] Discovered open port tcp/49187 on
    192.168.99.19
[*] [192.168.99.19/tcp/80/http/vhost-enum] The target was not a hostname, nor
    was a hostname provided as an option. Skipping virtual host enumeration.
[*] [192.168.99.19/tcp/80/http/known-security] [tcp/80/http/known-security]
    There did not appear to be a .well-known/security.txt file in the webroot
    (/).
[*] [192.168.99.19/tcp/80/http/curl-robots] [tcp/80/http/curl-robots] There
    did not appear to be a robots.txt file in the webroot (/).
[*] [192.168.99.19/top-100-udp-ports] Discovered open port udp/111 on
    192.168.99.19
[*] 15:26:20 - There are 2 scans still running against 192.168.99.19
[*] 15:27:20 - There is 1 scan still running against 192.168.99.19
[*] 15:28:20 - There is 1 scan still running against 192.168.99.19
[*] 15:29:20 - There is 1 scan still running against 192.168.99.19
[*] Finished scanning target 192.168.99.19 in 4 minutes, 38 seconds
[*] Finished scanning all targets in 4 minutes, 39 seconds!
[*] Don't forget to check out more commands to run manually in the
    _manual_commands.txt file in each target's scans directory!
```

3.3.1. While the scan is running

Open another terminal session

```
cd results/192.168.99.19/scans/

cat _quick_tcp_nmap.txt
```

Output:

```
# Nmap 7.94SVN scan initiated Sat Jul 13 15:25:20 2024 as:
# /usr/lib/nmap/nmap -vv --reason -Pn -T4 -sV -sC --version-all -A --osscan-
# guess -oN /root/results/192.168.99.19/scans/_quick_tcp_nmap.txt -oX
# /root/results/192.168.99.19/scans/xml/_quick_tcp_nmap.xml 192.168.99.19

Warning: Hit PCRE_ERROR_MATCHLIMIT when probing for service http with the
regex '^HTTP/1\.1 \d\d\d (?:[^\r\n]*\r\n(?!\r\n))*?.*\r\nServer: Virata-
EmWeb/R([\d_]+)\r\nContent-Type: text/html; ?charset=UTF-8\r\nExpires: .*<title>
```

```
HP (Color |)LaserJet ([\w._ -]+)&amp;nbsp;&amp;nbsp;&amp;nbsp;'

Nmap scan report for 192.168.99.19
Host is up, received arp-response (0.0011s latency).
Scanned at 2024-07-13 15:25:20 EDT for 11s
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE REASON         VERSION
22/tcp  open  ssh     syn-ack ttl 64 OpenSSH 6.7p1 Debian 5+deb8u3 (protocol
2.0)
| ssh-hostkey:
|   1024 ae:81:bc:7f:e3:f3:6b:e8:97:9d:93:17:1c:96:56:7b (DSA)
| ssh-dss AAAAB3NzaC1kc3MAAACBAIcOVzDiEICZZP219CpPnXPkdom4+77l/ZA3QPIcAkwid/
| ZkxMpA4tyeqwww2QzGAdFnZZzBl//rTqVNJ6CyWSboZpUBdl1giZ6FIW9rPjohtbxMPo7/qkcVxc
| KMmyjpfcUJd32SmwS/um/vtnIM/fA81+t042W2Oy8/howex+HlAAAAFQClybZKx4ih04cJEhDdd
| ZhgNuwRUwAAAIABs4t9fd31i+/6sOaIA3ZuWL7wdW+AcaRx5BOqtiAjWu8fT0pQy0rYIOmv+fQO
| 4xAyuEfiEzjVZ/ZhamnDJrJZu1JCNUHnjcfBZE+QvRD+xbD4rvdFkC8RSrrGLCgpWqExnlvZZrp
| 6UxGTOmsCnABIkF7o0QI6nd4xfXw6Ue6mMwAAAIBv73/s7+2lnfwxjBgk7sgMCfAlBwO0zEnrbt
| V9gLb86/TD3koHKZKfNyB0U2ybrSovzaX8CWLdEvnIQ2C+O99zxhV799VwWE/pO3ENWk4sTul5R
| Vf5VlPSZtnLC8cDvPz0njxpZ4dg/U5XXn6F3CN3m6rLmL4JGeU3H4xm5Ij8Vg==
|   2048 7d:bf:32:f2:11:c9:01:50:f3:06:f6:4b:7e:1b:d7:e6 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCwtVtdegw4ZWWQ4kXn0DtYSNHylL/JTF5tKmS
| EcgFTZdQN5Gh2ud7J5AHkcqkyzqXXxG236hcIDe3ELjlhEQX85BGTRcaUqYBNtPoLxGTcermchu
| wR++SRG5njaALfgZy6r3Q1m1t9C7yoVjKfUho6xpZxy3u7RDCt44QqCKzNndHa/1gB3iOTX1Ct2
| jabSHZFMoZVYmQLMU4VipNqhOpSWBqopVbjmYH4TTXkuUEFUwb2vhwWse9o+oMJIOPsddTCqLjH
| cFPw1eSFWOnaoF7ddEekupD83bmVnFxO3IzrXEr+OrJWEnCtvUytR6jSHHG00Bmd/EQq08sdoMC
| btJjh
|   256 1f:c0:c7:bd:bb:78:5d:6e:59:37:e4:32:df:81:3a:11 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEJ8
| gizqgZuhxViPC270v9SCH1L453AvgQ1UVvlfkJ4oqz5h9KRQS3e7Vrnj5H+I/zkjsZsb1JtbnO/
| XXR7sPc0=
|   256 01:e1:6e:1b:ef:99:ff:e3:4f:19:72:0d:8c:15:16:d7 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICe/em6Rv9cRCpIV2qvsCpeRmSyZRQp1NAUqBCKr
| 53Bw

80/tcp  open  http    syn-ack ttl 64 Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Apache2 Debian Default Page: It works

111/tcp open  rpcbind syn-ack ttl 64 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          38289/udp   status
|   100024  1          49187/tcp   status
|   100024  1          52020/udp6  status
|_  100024  1          53032/tcp6  status

MAC Address: 08:00:27:16:9E:22 (Oracle VirtualBox virtual NIC)
```

```
OS fingerprint not ideal because: Didn't receive UDP response. Please try
again with -sSU

Aggressive OS guesses: Linux 3.2 - 4.9 (98%), Linux 3.10 - 4.11 (94%), Linux
3.13 (94%), Linux 3.13 - 3.16 (94%), OpenWrt Chaos Calmer 15.05 (Linux 3.18)
or Designated Driver (Linux 4.1 or 4.4) (94%), Linux 4.10 (94%), Android 5.0
- 6.0.1 (Linux 3.4) (94%), Linux 3.10 (94%), Linux 3.2 - 3.10 (94%), Linux
3.2 - 3.16 (94%)

No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SCAN(V=7.94SVN%E=4%D=10/5%OT=22%CT=1%CU=%PV=Y%DS=1%DC=D%G=N%M=080027%TM=670192
AB%P=x86_64-pc-linux-gnu)
SEQ(SP=107%GCD=1%ISR=10B%TI=Z%CI=I%II=I%TS=8)
OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=M5B4ST11N
W7%O6=M5B4ST11)
WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)
ECN(R=Y%DF=Y%TG=40%W=7210%O=M5B4NNSNW7%CC=Y%Q=)
T1(R=Y%DF=Y%TG=40%S=O%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=N)
T4(R=Y%DF=Y%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T5(R=Y%DF=Y%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=Y%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T7(R=Y%DF=Y%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
U1(R=N)
IE(R=Y%DFI=N%TG=40%CD=S)

Uptime guess: 0.012 days (since Sat July 13 15:08:17 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   1.11 ms 192.168.99.19

Read data files from: /usr/share/nmap
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .

# Nmap done at Sat July 13 15:25:31 2024 -- 1 IP address (1 host up) scanned
in 11.18 seconds
```

Errors file

```
┌──(root㉿kali-linux-vagrant)-[~/results/192.168.99.19/scans]
```

```
└─# cat _errors.log
[*] Service scan showmount (tcp/111/rpcbind/showmount) ran a command which
returned a non-zero exit code (1).
[-] Command: showmount -e 192.168.99.19 2>&1
[-] Error Output:


[*] Service scan showmount (udp/111/rpcbind/showmount) ran a command which
returned a non-zero exit code (1).
[-] Command: showmount -e 192.168.99.19 2>&1
[-] Error Output:
```

Commands file

```
┌──(root㉿kali-linux-vagrant)-[~/results/192.168.99.19/scans]
└─# cat _commands.log
nmap -vv --reason -Pn -T4 -sV -sC --version-all -A --osscan-guess -oN
"/root/results/192.168.99.19/scans/_quick_tcp_nmap.txt" -oX
"/root/results/192.168.99.19/scans/xml/_quick_tcp_nmap.xml" 192.168.99.19

nmap -vv --reason -Pn -T4 -sV -sC --version-all -A --osscan-guess -p- -oN
"/root/results/192.168.99.19/scans/_full_tcp_nmap.txt" -oX
"/root/results/192.168.99.19/scans/xml/_full_tcp_nmap.xml" 192.168.99.19

nmap -vv --reason -Pn -T4 -sU -A --top-ports 100 -oN
"/root/results/192.168.99.19/scans/_top_100_udp_nmap.txt" -oX
"/root/results/192.168.99.19/scans/xml/_top_100_udp_nmap.xml" 192.168.99.19

nmap -vv --reason -Pn -T4 -sV -p 22 --script="banner,ssh2-enum-algos,
ssh-hostkey,ssh-auth-methods" -oN
"/root/results/192.168.99.19/scans/tcp22/tcp_22_ssh_nmap.txt" -oX
"/root/results/192.168.99.19/scans/tcp22/xml/tcp_22_ssh_nmap.xml" 192.168.99.19

feroxbuster -u http://192.168.99.19:80/ -t 10 -w
/root/.local/share/AutoRecon/wordlists/dirbuster.txt -x "txt,html,php,asp,aspx,
jsp" -v -k -n -q -e -r -o
"/root/results/192.168.99.19/scans/tcp80/tcp_80_http_feroxbuster_dirbuster.txt"

curl -sSikf http://192.168.99.19:80/.well-known/security.txt

curl -sSikf http://192.168.99.19:80/robots.txt

curl -sSik http://192.168.99.19:80/

nikto -ask=no -Tuning=x4567890ac -nointeractive -host
http://192.168.99.19:80 2>&1 | tee
"/root/results/192.168.99.19/scans/tcp80/tcp_80_http_nikto.txt"

nmap -vv --reason -Pn -T4 -sV -p 80 --script="banner,(http* or ssl*) and not
```

```
(brute or broadcast or dos or external or http-slowloris* or fuzzer)" -oN
"/root/results/192.168.99.19/scans/tcp80/tcp_80_http_nmap.txt" -oX
"/root/results/192.168.99.19/scans/tcp80/xml/tcp_80_http_nmap.xml" 192.168.99.19

whatweb --color=never --no-errors -a 3 -v http://192.168.99.19:80 2>&1

wkhtmltoimage --format png http://192.168.99.19:80/
/root/results/192.168.99.19/scans/tcp80/tcp_80_http_screenshot.png

nmap -vv --reason -Pn -T4 -sV -p 111 --script="banner,msrpc-enum,rpc-grind,
rpcinfo" -oN "/root/results/192.168.99.19/scans/tcp111/tcp_111_rpc_nmap.txt"
-oX "/root/results/192.168.99.19/scans/tcp111/xml/tcp_111_rpc_nmap.xml"
192.168.99.19

nmap -vv --reason -Pn -T4 -sV -p 111 --script="banner,(rpcinfo or nfs*) and
not (brute or broadcast or dos or external or fuzzer)" -oN
"/root/results/192.168.99.19/scans/tcp111/tcp_111_nfs_nmap.txt" -oX
"/root/results/192.168.99.19/scans/tcp111/xml/tcp_111_nfs_nmap.xml"
192.168.99.19

showmount -e 192.168.99.19 2>&1

nmap -vv --reason -Pn -T4 -sU -sV -p 111 --script="banner,msrpc-enum,
rpc-grind,rpcinfo" -oN "/root/results/192.168.99.19/scans/udp111/
udp_111_rpc_nmap.txt" -oX "/root/results/192.168.99.19/scans/udp111/xml/
udp_111_rpc_nmap.xml" 192.168.99.19

nmap -vv --reason -Pn -T4 -sU -sV -p 111 --script="banner,(rpcinfo or nfs*)
and not (brute or broadcast or dos or external or fuzzer)" -oN
"/root/results/192.168.99.19/scans/udp111/udp_111_nfs_nmap.txt" -oX
"/root/results/192.168.99.19/scans/udp111/xml/udp_111_nfs_nmap.xml"
192.168.99.19

showmount -e 192.168.99.19 2>&1
```

Manual Commands

The following maunal_commands text file shows us all of the manually run commands from AutoRecon.

This is a fantastic way to learn these tools if you are not familiar with the following:

```
┌──(root㉿kali-linux-vagrant)-[~/results/192.168.99.19/scans]
└─# cat _manual_commands.txt
```

```
[*] ssh on tcp/22

    [-] Bruteforce logins:

        hydra -L "/usr/share/seclists/Usernames/top-usernames-shortlist.txt" -P
"/usr/share/seclists/Passwords/darkweb2017-top100.txt" -e nsr -s 22 -o
"/root/results/192.168.99.19/scans/tcp22/tcp_22_ssh_hydra.txt"
ssh://192.168.99.19

        medusa -U "/usr/share/seclists/Usernames/top-usernames-shortlist.txt" -P
"/usr/share/seclists/Passwords/darkweb2017-top100.txt" -e ns -n 22 -O
"/root/results/192.168.99.19/scans/tcp22/tcp_22_ssh_medusa.txt" -M ssh -h
192.168.99.19

[*] http on tcp/80

    [-] (feroxbuster) Multi-threaded recursive directory/file enumeration for
web servers using various wordlists:

        feroxbuster -u http://192.168.99.19:80 -t 10 -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x
"txt,html,php,asp,aspx,jsp" -v -k -n -e -r -o
/root/results/192.168.99.19/scans/tcp80/tcp_80_http_feroxbuster_dirbuster.txt

    [-] Credential bruteforcing commands (don't run these without modifying
them):

        hydra -L "/usr/share/seclists/Usernames/top-usernames-shortlist.txt" -P
"/usr/share/seclists/Passwords/darkweb2017-top100.txt" -e nsr -s 80 -o
"/root/results/192.168.99.19/scans/tcp80/tcp_80_http_auth_hydra.txt" http-
get://192.168.99.19/path/to/auth/area

        medusa -U "/usr/share/seclists/Usernames/top-usernames-shortlist.txt" -P
"/usr/share/seclists/Passwords/darkweb2017-top100.txt" -e ns -n 80 -O
"/root/results/192.168.99.19/scans/tcp80/tcp_80_http_auth_medusa.txt" -M http -h
192.168.99.19 -m DIR:/path/to/auth/area

        hydra -L "/usr/share/seclists/Usernames/top-usernames-shortlist.txt" -P
"/usr/share/seclists/Passwords/darkweb2017-top100.txt" -e nsr -s 80 -o
"/root/results/192.168.99.19/scans/tcp80/tcp_80_http_form_hydra.txt" http-post-
form://192.168.99.19/path/to/login.php:"username=^USER^&password=^PASS^":"invali
d-login-message"

        medusa -U "/usr/share/seclists/Usernames/top-usernames-shortlist.txt" -P
"/usr/share/seclists/Passwords/darkweb2017-top100.txt" -e ns -n 80 -O
"/root/results/192.168.99.19/scans/tcp80/tcp_80_http_form_medusa.txt" -M web-
form -h 192.168.99.19 -m FORM:/path/to/login.php -m FORM-
DATA:"post?username=&password=" -m DENY-SIGNAL:"invalid login message"

    [-] (wpscan) WordPress Security Scanner (useful if WordPress is found):

        wpscan --url http://192.168.99.19:80/ --no-update -e vp,vt,tt,cb,dbe,u,m
--plugins-detection aggressive --plugins-version-detection aggressive -f cli-no-
```

```
color 2>&1 | tee
"/root/results/192.168.99.19/scans/tcp80/tcp_80_http_wpscan.txt"

[*] rpcbind on tcp/111

    [-] RPC Client:

        rpcclient -p 111 -U "" 192.168.99.19

[*] rpcbind on udp/111

    [-] RPC Client:

        rpcclient -p 111 -U "" 192.168.99.19
```

Top 100 UDP Nmap

```
┌──(root㉿kali-linux-vagrant)-[~/results/192.168.99.19/scans]
└─# cat _top_100_udp_nmap.txt
# Nmap 7.94SVN scan initiated Sat July 13 15:25:20 2024 as: /usr/lib/nmap/nmap
-vv --reason -Pn -T4 -sU -A --top-ports 100 -oN
/root/results/192.168.99.19/scans/_top_100_udp_nmap.txt -oX
/root/results/192.168.99.19/scans/xml/_top_100_udp_nmap.xml 192.168.99.19
Warning: 192.168.99.19 giving up on port because retransmission cap hit (6).
Increasing send delay for 192.168.99.19 from 100 to 200 due to 11 out of 11
dropped probes since last increase.
Increasing send delay for 192.168.99.19 from 200 to 400 due to 11 out of 11
dropped probes since last increase.
Increasing send delay for 192.168.99.19 from 400 to 800 due to 11 out of 11
dropped probes since last increase.
Nmap scan report for 192.168.99.19
Host is up, received arp-response (0.00061s latency).
Scanned at 2024-07-13 15:25:20 EDT for 277s

PORT      STATE         SERVICE         REASON              VERSION
7/udp     closed        echo            port-unreach ttl 64
9/udp     closed        discard         port-unreach ttl 64
17/udp    closed        qotd            port-unreach ttl 64
19/udp    closed        chargen         port-unreach ttl 64
49/udp    closed        tacacs          port-unreach ttl 64
53/udp    closed        domain          port-unreach ttl 64
67/udp    open|filtered dhcps           no-response
68/udp    open|filtered dhcpc           no-response
69/udp    open|filtered tftp            no-response
80/udp    closed        http            port-unreach ttl 64
88/udp    open|filtered kerberos-sec    no-response
111/udp   open          rpcbind         udp-response ttl 64 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
```

```
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          38289/udp   status
|   100024  1          49187/tcp   status
|   100024  1          52020/udp6  status
|_  100024  1          53032/tcp6  status
120/udp   closed        cfdptkt         port-unreach ttl 64
123/udp   closed        ntp             port-unreach ttl 64
135/udp   open|filtered msrpc           no-response
136/udp   open|filtered profile         no-response
137/udp   open|filtered netbios-ns      no-response
138/udp   closed        netbios-dgm     port-unreach ttl 64
139/udp   closed        netbios-ssn     port-unreach ttl 64
158/udp   open|filtered pcmail-srv      no-response
161/udp   closed        snmp            port-unreach ttl 64
162/udp   open|filtered snmptrap        no-response
177/udp   open|filtered xdmcp           no-response
427/udp   closed        svrloc          port-unreach ttl 64
443/udp   closed        https           port-unreach ttl 64
445/udp   open|filtered microsoft-ds    no-response
497/udp   open|filtered retrospect      no-response
500/udp   closed        isakmp          port-unreach ttl 64
514/udp   open|filtered syslog          no-response
515/udp   closed        printer         port-unreach ttl 64
518/udp   closed        ntalk           port-unreach ttl 64
520/udp   open|filtered route           no-response
593/udp   closed        http-rpc-epmap  port-unreach ttl 64
623/udp   closed        asf-rmcp        port-unreach ttl 64
626/udp   closed        serialnumberd   port-unreach ttl 64
631/udp   closed        ipp             port-unreach ttl 64
996/udp   closed        vsinet          port-unreach ttl 64
997/udp   closed        maitrd          port-unreach ttl 64
998/udp   closed        puparp          port-unreach ttl 64
999/udp   open|filtered applix          no-response
1022/udp  closed        exp2            port-unreach ttl 64
1023/udp  closed        unknown         port-unreach ttl 64
1025/udp  closed        blackjack       port-unreach ttl 64
1026/udp  closed        win-rpc         port-unreach ttl 64
1027/udp  closed        unknown         port-unreach ttl 64
1028/udp  open|filtered ms-lsa          no-response
1029/udp  open|filtered solid-mux       no-response
1030/udp  closed        iad1            port-unreach ttl 64
1433/udp  open|filtered ms-sql-s        no-response
1434/udp  open|filtered ms-sql-m        no-response
1645/udp  closed        radius          port-unreach ttl 64
1646/udp  closed        radacct         port-unreach ttl 64
1701/udp  closed        L2TP            port-unreach ttl 64
1718/udp  open|filtered h225gatedisc    no-response
1719/udp  closed        h323gatestat    port-unreach ttl 64
1812/udp  closed        radius          port-unreach ttl 64
1813/udp  open|filtered radacct         no-response
```

```
1900/udp  closed        upnp            port-unreach ttl 64
2000/udp  open|filtered cisco-sccp      no-response
2048/udp  closed        dls-monitor     port-unreach ttl 64
2049/udp  open|filtered nfs             no-response
2222/udp  open|filtered msantipiracy    no-response
2223/udp  open|filtered rockwell-csp2   no-response
3283/udp  closed        netassistant    port-unreach ttl 64
3456/udp  closed        IISrpc-or-vat   port-unreach ttl 64
3703/udp  closed        adobeserver-3   port-unreach ttl 64
4444/udp  closed        krb524          port-unreach ttl 64
4500/udp  closed        nat-t-ike       port-unreach ttl 64
5000/udp  closed        upnp            port-unreach ttl 64
5060/udp  closed        sip             port-unreach ttl 64
5353/udp  closed        zeroconf        port-unreach ttl 64
5632/udp  closed        pcanywherestat  port-unreach ttl 64
9200/udp  closed        wap-wsp         port-unreach ttl 64
10000/udp closed        ndmp            port-unreach ttl 64
17185/udp open|filtered wdbrpc          no-response
20031/udp open|filtered bakbonenetvault no-response
30718/udp closed        unknown         port-unreach ttl 64
31337/udp closed        BackOrifice     port-unreach ttl 64
32768/udp closed        omad            port-unreach ttl 64
32769/udp closed        filenet-rpc     port-unreach ttl 64
32771/udp closed        sometimes-rpc6  port-unreach ttl 64
32815/udp closed        unknown         port-unreach ttl 64
33281/udp open|filtered unknown         no-response
49152/udp closed        unknown         port-unreach ttl 64
49153/udp closed        unknown         port-unreach ttl 64
49154/udp closed        unknown         port-unreach ttl 64
49156/udp closed        unknown         port-unreach ttl 64
49181/udp open|filtered unknown         no-response
49182/udp closed        unknown         port-unreach ttl 64
49185/udp closed        unknown         port-unreach ttl 64
49186/udp open|filtered unknown         no-response
49188/udp open|filtered unknown         no-response
49190/udp closed        unknown         port-unreach ttl 64
49191/udp closed        unknown         port-unreach ttl 64
49192/udp open|filtered unknown         no-response
49193/udp closed        unknown         port-unreach ttl 64
49194/udp open|filtered unknown         no-response
49200/udp closed        unknown         port-unreach ttl 64
49201/udp closed        unknown         port-unreach ttl 64
65024/udp open|filtered unknown         no-response
MAC Address: 08:00:27:16:9E:22 (Oracle VirtualBox virtual NIC)
Too many fingerprints match this host to give specific OS details
TCP/IP fingerprint:
SCAN(V=7.94SVN%E=4%D=10/5%OT=%CT=%CU=7%PV=Y%DS=1%DC=D%G=N%M=080027%TM=670193B5%P
=x86_64-pc-linux-gnu)
SEQ(CI=I%II=I)
T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
```

```
IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.61 ms 192.168.99.19

Read data files from: /usr/share/nmap
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Sat Jul 13 15:29:57 2024 -- 1 IP address (1 host up) scanned in
276.91 seconds
```

Full TCP Nmap

```
┌──(root㉿kali-linux-vagrant)-[~/results/192.168.99.19/scans]
└─# cat _full_tcp_nmap.txt
# Nmap 7.94SVN scan initiated Sat Jul 13 15:25:20 2024 as: /usr/lib/nmap/nmap
-vv --reason -Pn -T4 -sV -sC --version-all -A --osscan-guess -p- -oN
/root/results/192.168.99.19/scans/_full_tcp_nmap.txt -oX
/root/results/192.168.99.19/scans/xml/_full_tcp_nmap.xml 192.168.99.19

adjust_timeouts2: packet supposedly had rtt of -528220 microseconds.  Ignoring
time.
adjust_timeouts2: packet supposedly had rtt of -528220 microseconds.  Ignoring
time.
adjust_timeouts2: packet supposedly had rtt of -528256 microseconds.  Ignoring
time.
adjust_timeouts2: packet supposedly had rtt of -528256 microseconds.  Ignoring
time.

Nmap scan report for 192.168.99.19
Host is up, received arp-response (0.00027s latency).
Scanned at 2024-07-13 15:25:20 EDT for 18s
Not shown: 65531 closed tcp ports (reset)

PORT      STATE SERVICE REASON         VERSION
22/tcp    open  ssh     syn-ack ttl 64 OpenSSH 6.7p1 Debian 5+deb8u3
                                       (protocol 2.0)
| ssh-hostkey:
|   1024 ae:81:bc:7f:e3:f3:6b:e8:97:9d:93:17:1c:96:56:7b (DSA)
| ssh-dss AAAAB3NzaC1kc3MAAACBAIcOVzDiEICZZP219CpPnXPkdom4+77l/ZA3QPIcAkwid/
| ZkxMpA4tyeqwww2QzGAdFnZZzBl//rTqVNJ6CyWSboZpUBdl1giZ6FIW9rPjohtbxMPo7/qkcVxc
| KMmyjpfcUJd32SmwS/um/vtnIM/fA81+t042W2Oy8/howex+HlAAAAFQClybZKx4ih04cJEhDdd
| ZhgNuwRUwAAAIABs4t9fd31i+/6sOaIA3ZuWL7wdW+AcaRx5BOqtiAjWu8fT0pQy0rYIOmv+fQO
| 4xAyuEfiEzjVZ/ZhamnDJrJZu1JCNUHnjcfBZE+QvRD+xbD4rvdFkC8RSrrGLCgpWqExnlvZZrp
| 6UxGTOmsCnABIkF7o0QI6nd4xfXw6Ue6mMwAAAIBv73/s7+2lnfwxjBgk7sgMCfAlBwO0zEnrbt
| V9gLb86/TD3koHKZKfNyB0U2ybrSovzaX8CWLdEvnIQ2C+O99zxhV799VwWE/pO3ENWk4sTul5R
```

```
| Vf5VlPSZtnLC8cDvPz0njxpZ4dg/U5XXn6F3CN3m6rLmL4JGeU3H4xm5Ij8Vg==
|   2048 7d:bf:32:f2:11:c9:01:50:f3:06:f6:4b:7e:1b:d7:e6 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCwtVtdegw4ZWWQ4kXn0DtYSNHylL/JTF5tKmS
| EcgFTZdQN5Gh2ud7J5AHkcqkyzqXXxG236hcIDe3ELjlhEQX85BGTRcaUqYBNtPoLxGTcermchu
| wR++SRG5njaALfgZy6r3Q1m1t9C7yoVjKfUho6xpZxy3u7RDCt44QqCKzNndHa/1gB3iOTX1Ct2
| jabSHZFMoZVYmQLMU4VipNqhOpSWBqopVbjmYH4TTXkuUEFUwb2vhwWse9o+oMJIOPsddTCqLjH
| cFPw1eSFWOnaoF7ddEekupD83bmVnFxO3IzrXEr+OrJWEnCtvUytR6jSHHG00Bmd/EQq08sdoMC
| btJjh
|   256 1f:c0:c7:bd:bb:78:5d:6e:59:37:e4:32:df:81:3a:11 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEJ8
| gizqgZuhxViPC270v9SCH1L453AvgQ1UVvlfkJ4oqz5h9KRQS3e7Vrnj5H+I/zkjsZsb1JtbnO/
| XXR7sPc0=
|   256 01:e1:6e:1b:ef:99:ff:e3:4f:19:72:0d:8c:15:16:d7 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICe/em6Rv9cRCpIV2qvsCpeRmSyZRQp1NAUqBCKr
| 53Bw

80/tcp    open  http    syn-ack ttl 64 Apache httpd 2.4.10 ((Debian))
|_http-title: Apache2 Debian Default Page: It works
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.10 (Debian)

111/tcp   open  rpcbind syn-ack ttl 64 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          38289/udp   status
|   100024  1          49187/tcp   status
|   100024  1          52020/udp6  status
|_  100024  1          53032/tcp6  status

49187/tcp open  status  syn-ack ttl 64 1 (RPC #100024)

MAC Address: 08:00:27:16:9E:22 (Oracle VirtualBox virtual NIC)

OS fingerprint not ideal because: Didn't receive UDP response. Please try
again with -sSU
Aggressive OS guesses: Linux 3.2 - 4.9 (95%), Linux 3.13 (91%), OpenWrt Chaos
Calmer 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4) (91%), Linux
4.10 (91%), Linux 3.10 - 4.11 (91%), Linux 3.2 - 3.16 (91%), Linux 3.13 - 3.16
(90%), Linux 3.16 - 4.6 (90%), Android 5.0 - 6.0.1 (Linux 3.4) (90%), Linux
3.2 - 3.10 (90%)
No exact OS matches for host (test conditions non-ideal).

TCP/IP fingerprint:
SCAN(V=7.94SVN%E=4%D=10/5%OT=22%CT=1%CU=%PV=Y%DS=1%DC=D%G=N%M=080027%TM=670192
B2%P=x86_64-pc-linux-gnu)
SEQ(SP=FA%GCD=1%ISR=108%TI=Z%CI=I%II=I%TS=8)
OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=M
```

```
5B4ST11N
W7%O6=M5B4ST11)
WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)
ECN(R=Y%DF=Y%TG=40%W=7210%O=M5B4NNSNW7%CC=Y%Q=)
T1(R=Y%DF=Y%TG=40%S=O%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=N)
T4(R=Y%DF=Y%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T4(R=Y%DF=Y%TG=40%W=0%S=O%A=Z%F=R%O=%RD=0%Q=)
T5(R=N)
T5(R=Y%DF=Y%TG=40%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=Y%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T6(R=Y%DF=Y%TG=40%W=0%S=O%A=Z%F=R%O=%RD=0%Q=)
T7(R=Y%DF=Y%TG=40%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)
T7(R=Y%DF=Y%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
U1(R=N)
IE(R=Y%DFI=N%TG=40%CD=S)

Uptime guess: 0.012 days (since Sat Jul 13 15:08:17 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=250 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.27 ms 192.168.99.19

Read data files from: /usr/share/nmap
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/.

# Nmap done at Sat Jul 13 15:25:38 2024 -- 1 IP address (1 host up) scanned
in 18.28 seconds
```

3.3.2. Next Steps

Consider potentially running MetaSploit to go after the system.

Per running service and saids version banner, consider:

```
┌──(root㉿kali-linux-vagrant)-[~]
└─# searchsploit OpenSSH 6.7p1
------------------------------------------------------------------------------
-----------------------
Exploit Title
|  Path
```

```
---------------------------------------------------------------------------------
----------------------
OpenSSH 2.3 < 7.7 - Username Enumeration
| linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)
| linux/remote/45210.py
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading
| linux/remote/40963.txt
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets
Privilege Escalation   | linux/local/40962.txt
OpenSSH < 7.7 - User Enumeration (2)
| linux/remote/45939.py
---------------------------------------------------------------------------------
----------------------
Shellcodes: No Results
```

3.3.3. Advanced Usuage

It might be worth testing some of the parameters in the help file.

```
autorecon \
  --max-scans 100 \
  --max-port-scans 20 \
  --exclude-tags="unsafe,http+long" \
  --profile thorough \
  -t <target_IP_Address> \
  -vvv
```

Where:

1. --max-scans:

   ```
   The maximum number of concurrent scans to run.
   Default: 50
   ```

2. --max-port-scans:

   ```
   The maximum number of concurrent port scans to run.
   Default: 10 (approx 20% of max-scans unless specified).
   ```

3. --exclude-tags:

```
The --exclude-tags option is used to determine which
plugins should be excluded. Group tags together by separating them with
a plus symbol (+), and separate groups with a comma (,) to create
multiple groups. For a plugin to be excluded, it must have all the tags
specified in at least one group.
```

```
Example: autorecon --exclude-tags="unsafe,http+long" -t <target_IP_Address>
```

4. -t (Target IP or CIDR):

```
Specifies the target IP address or range in
CIDR notation. This is a required parameter that defines the scope of
the scan.
```

```
Example: autorecon -t 192.168.1.1 or autorecon -t 192.168.1.0/24
```

5. -vvv (Verbose Mode):

```
Enables increased verbosity for detailed output
during the scan. This is useful for troubleshooting or getting more
insight into the scanning process.
```

```
Example: autorecon -t 192.168.1.1 -vvv
```

6. --no-recommends:

```
Skips recommended scan commands and only runs basic
service scans. This can save time if you only want essential information.
```

```
Example: autorecon -t 192.168.1.1 --no-recommends
```

7. --profile:

```
Allows the selection of a predefined scan profile, such as
quick, full, or thorough. Different profiles adjust the depth and speed
of the scan.
```

```
Example: autorecon -t 192.168.1.1 --profile full
```

8. -p (Port Specification):

```
Limits the scan to specific ports, either by
listing individual ports or ranges.
```

```
Example: autorecon -t 192.168.1.1 -p 80,443 or autorecon -t 192.168.1.1
-p 1-65535
```

9. -n (No DNS Resolution):

```
Disables DNS resolution for faster scans or in
environments where DNS is not needed.
```

```
Example: autorecon -t 192.168.1.1 -n
```

Chapter 4. Conclusion

In conclusion, AutoRecon serves as a powerful, automated network reconnaissance tool tailored to the needs of penetration testers and security professionals. Its ability to streamline the enumeration of services through multi-threading and integration with current and relevant security tools makes it an invaluable asset in both Capture the Flag (CTF) challenges and real-world penetration testing engagements. The tool's extensive configuration options, such as concurrent scan management, exclusion of specific plugins, and customizable scan profiles, provide flexibility and efficiency, allowing pen testers to focus on more in-depth vulnerability analysis and exploitation.

By simplifying the reconnaissance phase, AutoRecon enhances the speed and comprehensiveness of network scanning, making it ideal for both novice and experienced professionals. The detailed output, error logs, and manual command suggestions contribute to its educational value, enabling users to better understand and utilize a variety of scanning tools. Overall, AutoRecon is not only a time-saving solution but also an effective platform for mastering the intricacies of network service enumeration and security assessments.

Thank you for taking the time to read this white paper. As we conclude, we ask that you prepare for the end by metaphorically returning to your seat and securing any loose thoughts. Please fasten your mental seatbelt, ensuring your focus is in an upright and attentive position.

As you close this paper, we hope you retain the insights shared. The author sincerely appreciates your time and attention, knowing that you have countless options for reading material online. Thank you once again for choosing this white paper.

Chapter 5. Appendix

References

https://www.kali.org/tools/autorecon/

https://github.com/Tib3rius/AutoRecon

https://gitlab.com/kalilinux/packages/autorecon

https://github.com/Tib3rius/AutoRecon/wiki/Usage