**ANSIBLE PLAYBOOK**

root@e668a6213f08:/etc/ansible# cat **pentest.yml**

---

- name: Config Web VM with Docker

hosts: webservers

become: true

tasks:

- name: docker.io

apt:

force\_apt\_get: yes

name: docker.io

state: present

- name: Install pip

apt:

force\_apt\_get: yes

name: python-pip

state: present

- name: Install Docker python module

pip:

name: docker

state: present

- name: download and launch a docker web container

docker\_container:

name: dvwa

image: cyberxsecurity/dvwa

state: started

published\_ports: 80:80

**ANSIBLE CONFIGURATION**

**root@e668a6213f08:/etc/ansible# cat ansible.cfg**

# config file for ansible -- https://ansible.com/

# ===============================================

# nearly all parameters can be overridden in ansible-playbook

# or with command line flags. ansible will read ANSIBLE\_CONFIG,

# ansible.cfg in the current working directory, .ansible.cfg in

# the home directory or /etc/ansible/ansible.cfg, whichever it

# finds first

[defaults]

# some basic default values...

#inventory = /etc/ansible/hosts

#library = /usr/share/my\_modules/

#module\_utils = /usr/share/my\_module\_utils/

#remote\_tmp = ~/.ansible/tmp

#local\_tmp = ~/.ansible/tmp

#plugin\_filters\_cfg = /etc/ansible/plugin\_filters.yml

#forks = 5

#poll\_interval = 15

#sudo\_user = root

#ask\_sudo\_pass = True

#ask\_pass = True

#transport = smart

#remote\_port = 22

#module\_lang = C

#module\_set\_locale = False

# plays will gather facts by default, which contain information about

# the remote system.

#

# smart - gather by default, but don't regather if already gathered

# implicit - gather by default, turn off with gather\_facts: False

# explicit - do not gather by default, must say gather\_facts: True

#gathering = implicit

# This only affects the gathering done by a play's gather\_facts directive,

# by default gathering retrieves all facts subsets

# all - gather all subsets

# network - gather min and network facts

# hardware - gather hardware facts (longest facts to retrieve)

# virtual - gather min and virtual facts

# facter - import facts from facter

# ohai - import facts from ohai

# You can combine them using comma (ex: network,virtual)

# You can negate them using ! (ex: !hardware,!facter,!ohai)

# A minimal set of facts is always gathered.

#gather\_subset = all

# some hardware related facts are collected

# with a maximum timeout of 10 seconds. This

# option lets you increase or decrease that

# timeout to something more suitable for the

# environment.

# gather\_timeout = 10

# Ansible facts are available inside the ansible\_facts.\* dictionary

# namespace. This setting maintains the behaviour which was the default prior

# to 2.5, duplicating these variables into the main namespace, each with a

# prefix of 'ansible\_'.

# This variable is set to True by default for backwards compatibility. It

# will be changed to a default of 'False' in a future release.

# ansible\_facts.

# inject\_facts\_as\_vars = True

# additional paths to search for roles in, colon separated

#roles\_path = /etc/ansible/roles

# uncomment this to disable SSH key host checking

#host\_key\_checking = False

# change the default callback, you can only have one 'stdout' type enabled at a time.

#stdout\_callback = skippy

## Ansible ships with some plugins that require whitelisting,

## this is done to avoid running all of a type by default.

## These setting lists those that you want enabled for your system.

## Custom plugins should not need this unless plugin author specifies it.

# enable callback plugins, they can output to stdout but cannot be 'stdout' type.

#callback\_whitelist = timer, mail

# Determine whether includes in tasks and handlers are "static" by

# default. As of 2.0, includes are dynamic by default. Setting these

# values to True will make includes behave more like they did in the

# 1.x versions.

#task\_includes\_static = False

#handler\_includes\_static = False

# Controls if a missing handler for a notification event is an error or a warning

#error\_on\_missing\_handler = True

# change this for alternative sudo implementations

#sudo\_exe = sudo

# What flags to pass to sudo

# WARNING: leaving out the defaults might create unexpected behaviours

#sudo\_flags = -H -S -n

# SSH timeout

#timeout = 10

# default user to use for playbooks if user is not specified

# (/usr/bin/ansible will use current user as default)

remote\_user = markertle

# logging is off by default unless this path is defined

# if so defined, consider logrotate

#log\_path = /var/log/ansible.log

# default module name for /usr/bin/ansible

#module\_name = command

# use this shell for commands executed under sudo

# you may need to change this to bin/bash in rare instances

# if sudo is constrained

#executable = /bin/sh

# if inventory variables overlap, does the higher precedence one win

# or are hash values merged together? The default is 'replace' but

# this can also be set to 'merge'.

#hash\_behaviour = replace

# by default, variables from roles will be visible in the global variable

# scope. To prevent this, the following option can be enabled, and only

# tasks and handlers within the role will see the variables there

#private\_role\_vars = yes

# list any Jinja2 extensions to enable here:

#jinja2\_extensions = jinja2.ext.do,jinja2.ext.i18n

# if set, always use this private key file for authentication, same as

# if passing --private-key to ansible or ansible-playbook

#private\_key\_file = /path/to/file

# If set, configures the path to the Vault password file as an alternative to

# specifying --vault-password-file on the command line.

#vault\_password\_file = /path/to/vault\_password\_file

# format of string {{ ansible\_managed }} available within Jinja2

# templates indicates to users editing templates files will be replaced.

# replacing {file}, {host} and {uid} and strftime codes with proper values.

#ansible\_managed = Ansible managed: {file} modified on %Y-%m-%d %H:%M:%S by {uid} on {host}

# {file}, {host}, {uid}, and the timestamp can all interfere with idempotence

# in some situations so the default is a static string:

#ansible\_managed = Ansible managed

# by default, ansible-playbook will display "Skipping [host]" if it determines a task

# should not be run on a host. Set this to "False" if you don't want to see these "Skipping"

# messages. NOTE: the task header will still be shown regardless of whether or not the

# task is skipped.

#display\_skipped\_hosts = True

# by default, if a task in a playbook does not include a name: field then

# ansible-playbook will construct a header that includes the task's action but

# not the task's args. This is a security feature because ansible cannot know

# if the \*module\* considers an argument to be no\_log at the time that the

# header is printed. If your environment doesn't have a problem securing

# stdout from ansible-playbook (or you have manually specified no\_log in your

# playbook on all of the tasks where you have secret information) then you can

# safely set this to True to get more informative messages.

#display\_args\_to\_stdout = False

# by default (as of 1.3), Ansible will raise errors when attempting to dereference

# Jinja2 variables that are not set in templates or action lines. Uncomment this line

# to revert the behavior to pre-1.3.

#error\_on\_undefined\_vars = False

# by default (as of 1.6), Ansible may display warnings based on the configuration of the

# system running ansible itself. This may include warnings about 3rd party packages or

# other conditions that should be resolved if possible.

# to disable these warnings, set the following value to False:

#system\_warnings = True

# by default (as of 1.4), Ansible may display deprecation warnings for language

# features that should no longer be used and will be removed in future versions.

# to disable these warnings, set the following value to False:

#deprecation\_warnings = True

# (as of 1.8), Ansible can optionally warn when usage of the shell and

# command module appear to be simplified by using a default Ansible module

# instead. These warnings can be silenced by adjusting the following

# setting or adding warn=yes or warn=no to the end of the command line

# parameter string. This will for example suggest using the git module

# instead of shelling out to the git command.

# command\_warnings = False

# set plugin path directories here, separate with colons

#action\_plugins = /usr/share/ansible/plugins/action

#become\_plugins = /usr/share/ansible/plugins/become

#cache\_plugins = /usr/share/ansible/plugins/cache

#callback\_plugins = /usr/share/ansible/plugins/callback

#connection\_plugins = /usr/share/ansible/plugins/connection

#lookup\_plugins = /usr/share/ansible/plugins/lookup

#inventory\_plugins = /usr/share/ansible/plugins/inventory

#vars\_plugins = /usr/share/ansible/plugins/vars

#filter\_plugins = /usr/share/ansible/plugins/filter

#test\_plugins = /usr/share/ansible/plugins/test

#terminal\_plugins = /usr/share/ansible/plugins/terminal

#strategy\_plugins = /usr/share/ansible/plugins/strategy

# by default, ansible will use the 'linear' strategy but you may want to try

# another one

#strategy = free

# by default callbacks are not loaded for /bin/ansible, enable this if you

# want, for example, a notification or logging callback to also apply to

# /bin/ansible runs

#bin\_ansible\_callbacks = False

# don't like cows? that's unfortunate.

# set to 1 if you don't want cowsay support or export ANSIBLE\_NOCOWS=1

#nocows = 1

# set which cowsay stencil you'd like to use by default. When set to 'random',

# a random stencil will be selected for each task. The selection will be filtered

# against the `cow\_whitelist` option below.

#cow\_selection = default

#cow\_selection = random

# when using the 'random' option for cowsay, stencils will be restricted to this list.

# it should be formatted as a comma-separated list with no spaces between names.

# NOTE: line continuations here are for formatting purposes only, as the INI parser

# in python does not support them.

#cow\_whitelist=bud-frogs,bunny,cheese,daemon,default,dragon,elephant-in-snake,elephant,eyes,\

# hellokitty,kitty,luke-koala,meow,milk,moofasa,moose,ren,sheep,small,stegosaurus,\

# stimpy,supermilker,three-eyes,turkey,turtle,tux,udder,vader-koala,vader,www

# don't like colors either?

# set to 1 if you don't want colors, or export ANSIBLE\_NOCOLOR=1

#nocolor = 1

# if set to a persistent type (not 'memory', for example 'redis') fact values

# from previous runs in Ansible will be stored. This may be useful when

# wanting to use, for example, IP information from one group of servers

# without having to talk to them in the same playbook run to get their

# current IP information.

#fact\_caching = memory

#This option tells Ansible where to cache facts. The value is plugin dependent.

#For the jsonfile plugin, it should be a path to a local directory.

#For the redis plugin, the value is a host:port:database triplet: fact\_caching\_connection = localhost:6379:0

#fact\_caching\_connection=/tmp

# retry files

# When a playbook fails a .retry file can be created that will be placed in ~/

# You can enable this feature by setting retry\_files\_enabled to True

# and you can change the location of the files by setting retry\_files\_save\_path

#retry\_files\_enabled = False

#retry\_files\_save\_path = ~/.ansible-retry

# squash actions

# Ansible can optimise actions that call modules with list parameters

# when looping. Instead of calling the module once per with\_ item, the

# module is called once with all items at once. Currently this only works

# under limited circumstances, and only with parameters named 'name'.

#squash\_actions = apk,apt,dnf,homebrew,pacman,pkgng,yum,zypper

# prevents logging of task data, off by default

#no\_log = False

# prevents logging of tasks, but only on the targets, data is still logged on the master/controller

#no\_target\_syslog = False

# controls whether Ansible will raise an error or warning if a task has no

# choice but to create world readable temporary files to execute a module on

# the remote machine. This option is False by default for security. Users may

# turn this on to have behaviour more like Ansible prior to 2.1.x. See

# https://docs.ansible.com/ansible/become.html#becoming-an-unprivileged-user

# for more secure ways to fix this than enabling this option.

#allow\_world\_readable\_tmpfiles = False

# controls the compression level of variables sent to

# worker processes. At the default of 0, no compression

# is used. This value must be an integer from 0 to 9.

#var\_compression\_level = 9

# controls what compression method is used for new-style ansible modules when

# they are sent to the remote system. The compression types depend on having

# support compiled into both the controller's python and the client's python.

# The names should match with the python Zipfile compression types:

# \* ZIP\_STORED (no compression. available everywhere)

# \* ZIP\_DEFLATED (uses zlib, the default)

# These values may be set per host via the ansible\_module\_compression inventory

# variable

#module\_compression = 'ZIP\_DEFLATED'

# This controls the cutoff point (in bytes) on --diff for files

# set to 0 for unlimited (RAM may suffer!).

#max\_diff\_size = 1048576

# This controls how ansible handles multiple --tags and --skip-tags arguments

# on the CLI. If this is True then multiple arguments are merged together. If

# it is False, then the last specified argument is used and the others are ignored.

# This option will be removed in 2.8.

#merge\_multiple\_cli\_flags = True

# Controls showing custom stats at the end, off by default

#show\_custom\_stats = True

# Controls which files to ignore when using a directory as inventory with

# possibly multiple sources (both static and dynamic)

#inventory\_ignore\_extensions = ~, .orig, .bak, .ini, .cfg, .retry, .pyc, .pyo

# This family of modules use an alternative execution path optimized for network appliances

# only update this setting if you know how this works, otherwise it can break module execution

#network\_group\_modules=eos, nxos, ios, iosxr, junos, vyos

# When enabled, this option allows lookups (via variables like {{lookup('foo')}} or when used as

# a loop with `with\_foo`) to return data that is not marked "unsafe". This means the data may contain

# jinja2 templating language which will be run through the templating engine.

# ENABLING THIS COULD BE A SECURITY RISK

#allow\_unsafe\_lookups = False

# set default errors for all plays

#any\_errors\_fatal = False

[inventory]

# enable inventory plugins, default: 'host\_list', 'script', 'auto', 'yaml', 'ini', 'toml'

#enable\_plugins = host\_list, virtualbox, yaml, constructed

# ignore these extensions when parsing a directory as inventory source

#ignore\_extensions = .pyc, .pyo, .swp, .bak, ~, .rpm, .md, .txt, ~, .orig, .ini, .cfg, .retry

# ignore files matching these patterns when parsing a directory as inventory source

#ignore\_patterns=

# If 'true' unparsed inventory sources become fatal errors, they are warnings otherwise.

#unparsed\_is\_failed=False

[privilege\_escalation]

#become=True

#become\_method=sudo

#become\_user=root

#become\_ask\_pass=False

[paramiko\_connection]

# uncomment this line to cause the paramiko connection plugin to not record new host

# keys encountered. Increases performance on new host additions. Setting works independently of the

# host key checking setting above.

#record\_host\_keys=False

# by default, Ansible requests a pseudo-terminal for commands executed under sudo. Uncomment this

# line to disable this behaviour.

#pty=False

# paramiko will default to looking for SSH keys initially when trying to

# authenticate to remote devices. This is a problem for some network devices

# that close the connection after a key failure. Uncomment this line to

# disable the Paramiko look for keys function

#look\_for\_keys = False

# When using persistent connections with Paramiko, the connection runs in a

# background process. If the host doesn't already have a valid SSH key, by

# default Ansible will prompt to add the host key. This will cause connections

# running in background processes to fail. Uncomment this line to have

# Paramiko automatically add host keys.

#host\_key\_auto\_add = True

[ssh\_connection]

# ssh arguments to use

# Leaving off ControlPersist will result in poor performance, so use

# paramiko on older platforms rather than removing it, -C controls compression use

#ssh\_args = -C -o ControlMaster=auto -o ControlPersist=60s

# The base directory for the ControlPath sockets.

# This is the "%(directory)s" in the control\_path option

#

# Example:

# control\_path\_dir = /tmp/.ansible/cp

#control\_path\_dir = ~/.ansible/cp

# The path to use for the ControlPath sockets. This defaults to a hashed string of the hostname,

# port and username (empty string in the config). The hash mitigates a common problem users

# found with long hostnames and the conventional %(directory)s/ansible-ssh-%%h-%%p-%%r format.

# In those cases, a "too long for Unix domain socket" ssh error would occur.

#

# Example:

# control\_path = %(directory)s/%%h-%%r

#control\_path =

# Enabling pipelining reduces the number of SSH operations required to

# execute a module on the remote server. This can result in a significant

# performance improvement when enabled, however when using "sudo:" you must

# first disable 'requiretty' in /etc/sudoers

#

# By default, this option is disabled to preserve compatibility with

# sudoers configurations that have requiretty (the default on many distros).

#

#pipelining = False

# Control the mechanism for transferring files (old)

# \* smart = try sftp and then try scp [default]

# \* True = use scp only

# \* False = use sftp only

#scp\_if\_ssh = smart

# Control the mechanism for transferring files (new)

# If set, this will override the scp\_if\_ssh option

# \* sftp = use sftp to transfer files

# \* scp = use scp to transfer files

# \* piped = use 'dd' over SSH to transfer files

# \* smart = try sftp, scp, and piped, in that order [default]

#transfer\_method = smart

# if False, sftp will not use batch mode to transfer files. This may cause some

# types of file transfer failures impossible to catch however, and should

# only be disabled if your sftp version has problems with batch mode

#sftp\_batch\_mode = False

# The -tt argument is passed to ssh when pipelining is not enabled because sudo

# requires a tty by default.

#usetty = True

# Number of times to retry an SSH connection to a host, in case of UNREACHABLE.

# For each retry attempt, there is an exponential backoff,

# so after the first attempt there is 1s wait, then 2s, 4s etc. up to 30s (max).

#retries = 3

[persistent\_connection]

# Configures the persistent connection timeout value in seconds. This value is

# how long the persistent connection will remain idle before it is destroyed.

# If the connection doesn't receive a request before the timeout value

# expires, the connection is shutdown. The default value is 30 seconds.

#connect\_timeout = 30

# The command timeout value defines the amount of time to wait for a command

# or RPC call before timing out. The value for the command timeout must

# be less than the value of the persistent connection idle timeout (connect\_timeout)

# The default value is 30 second.

#command\_timeout = 30

[accelerate]

#accelerate\_port = 5099

#accelerate\_timeout = 30

#accelerate\_connect\_timeout = 5.0

# The daemon timeout is measured in minutes. This time is measured

# from the last activity to the accelerate daemon.

#accelerate\_daemon\_timeout = 30

# If set to yes, accelerate\_multi\_key will allow multiple

# private keys to be uploaded to it, though each user must

# have access to the system via SSH to add a new key. The default

# is "no".

#accelerate\_multi\_key = yes

[selinux]

# file systems that require special treatment when dealing with security context

# the default behaviour that copies the existing context or uses the user default

# needs to be changed to use the file system dependent context.

#special\_context\_filesystems=nfs,vboxsf,fuse,ramfs,9p,vfat

# Set this to yes to allow libvirt\_lxc connections to work without SELinux.

#libvirt\_lxc\_noseclabel = yes

[colors]

#highlight = white

#verbose = blue

#warn = bright purple

#error = red

#debug = dark gray

#deprecate = purple

#skip = cyan

#unreachable = red

#ok = green

#changed = yellow

#diff\_add = green

#diff\_remove = red

#diff\_lines = cyan

[diff]

# Always print diff when running ( same as always running with -D/--diff )

# always = no

# Set how many context lines to show in diff

# context = 3

**ElKPLAYBOOK**

**root@e668a6213f08:/etc/ansible# cat elkplaybook.yml**

---

- name: Configure Elk VM with Docker

hosts: elkservers

remote\_user: elk

become: true

tasks:

# Use apt module

- name: Install docker.io

apt:

force\_apt\_get: yes

name: #TODO

state: #TODO

# Use apt module

- name: Install python-pip

apt:

force\_apt\_get: yes

name: #TODO

state: #TODO

# Use pip module

- name: Install Docker module

pip:

name: #TODO

state: #TODO

# Use command module

- name: Increase virtual memory

command: sysctl -w vm.max\_map\_count=262144

# Use docker\_container module

- name: download and launch a docker elk container

docker\_container:

name: elk

image: sebp/elk

state: started

# Please list the ports that ELK runs on

published\_ports:

- #TODO

- #TODO

- #TODO

**ANSIBLE HOST FILE**

root@e668a6213f08:/etc/ansible# cat hosts

# This is the default ansible 'hosts' file.

#

# It should live in /etc/ansible/hosts

#

# - Comments begin with the '#' character

# - Blank lines are ignored

# - Groups of hosts are delimited by [header] elements

# - You can enter hostnames or ip addresses

# - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com

## blue.example.com

## 192.168.100.1

## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

[webservers]

## alpha.example.org

## beta.example.org

## 192.168.1.100

## 192.168.1.110

10.0.0.5

10.0.0.7

[elkservers]

10.0.0.4

# If you have multiple hosts following a pattern you can specify

# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]

##

## db01.intranet.mydomain.net

## db02.intranet.mydomain.net

## 10.25.1.56

## 10.25.1.57

# Here's another example of host ranges, this time there are no

# leading 0s:

## db-[99:101]-node.example.com