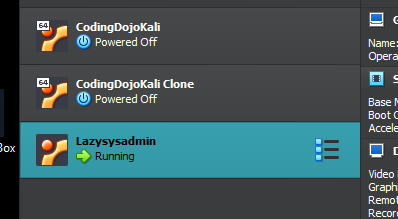
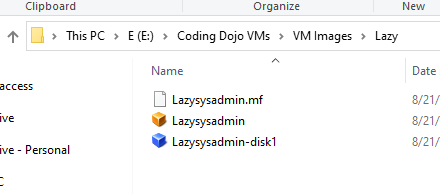
Name: Jon Murray

Assignment: Incident Response Lab

**Part 1: Attack**

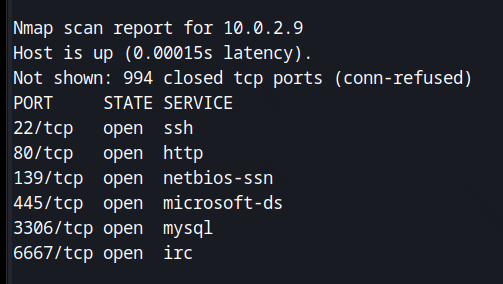
I began this lab by downloading the LazySysAdmin VM to my host machine, and then importing it into VirtualBox:



I then spun up the LazySysAdmin machine along with my Kali machine. Now that both machines are up and running, its time to attack the LazySysAdmin machine.

Phase 1 – Gathering Info:

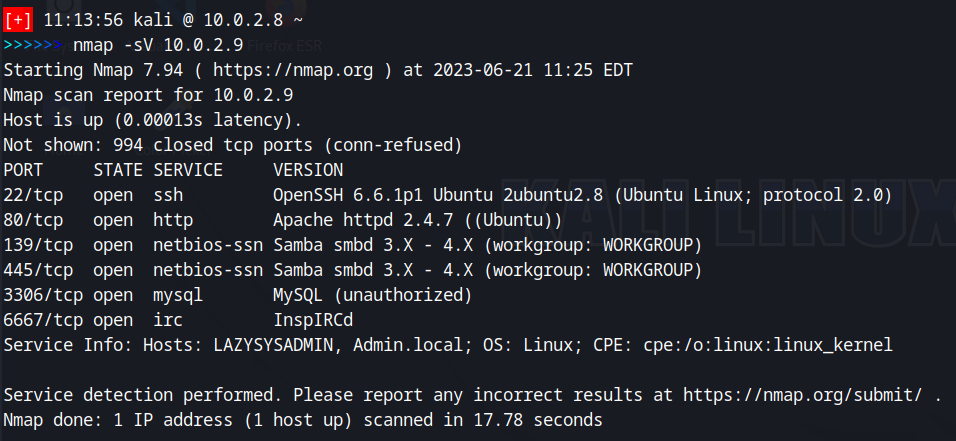
I pulled the terminal on Kali and conducted an nmap scan on 10.0.2.0, the network that these machines are on, to discover the ip address of the LazySysAdmin machine. Prior to that, I ran ifconfig though to get Kali’s ip address which was 10.0.2.8. LazySysAdmin was identified as 10.0.2.9:



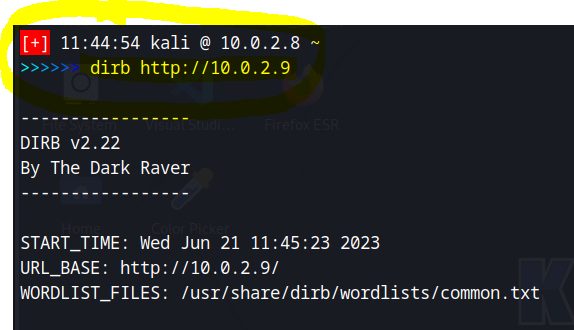
To gain further info on 10.0.2.9, I ran another nmap scan on just this host using the -sV option, which will scan the host and give info related to open ports and the service version info (from nmap -h, the help section on nmap):



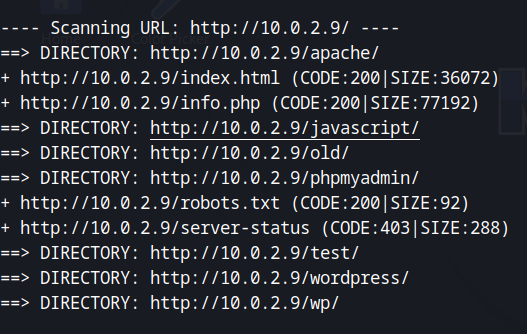
And we now have some additional info on LazySysAdmin:



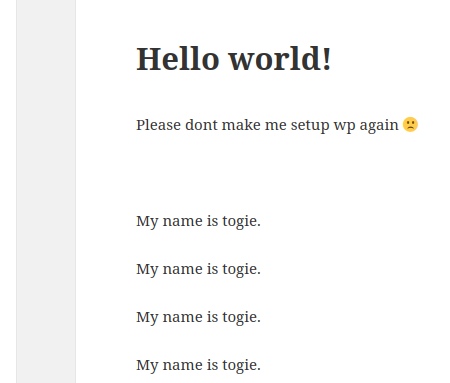
From this info, we can see that Apache vs 2.4.7 is running. This tells me that this is more than likely a website/server. We can use a tool called dirb to enumerate the directories of LazySysAdmin:



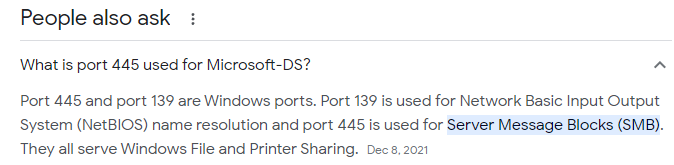
And we find the following directories:



I entered all of those directories. There does appear to be a php login page but as of yet, I do not have the credentials to login. There is also a WordPress page.

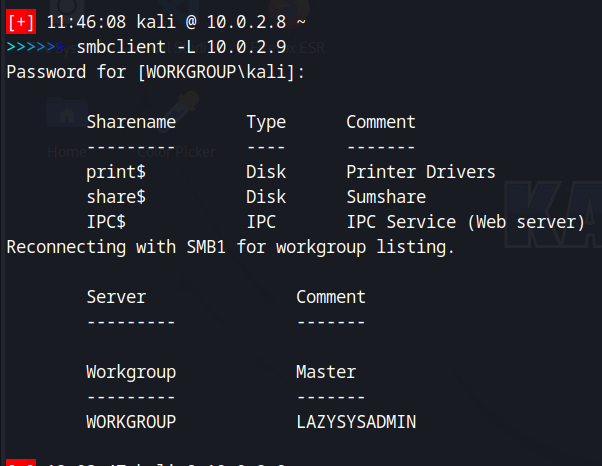


Maybe this “togie” is a username?? Going back to the nmap scan of this host, we see ports 139 and 445 are open. A google search of these ports shows:

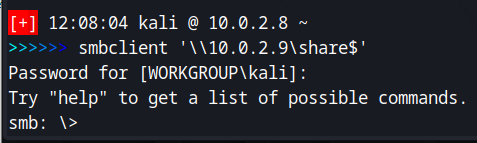


Phase 2 – Attempt to gain access

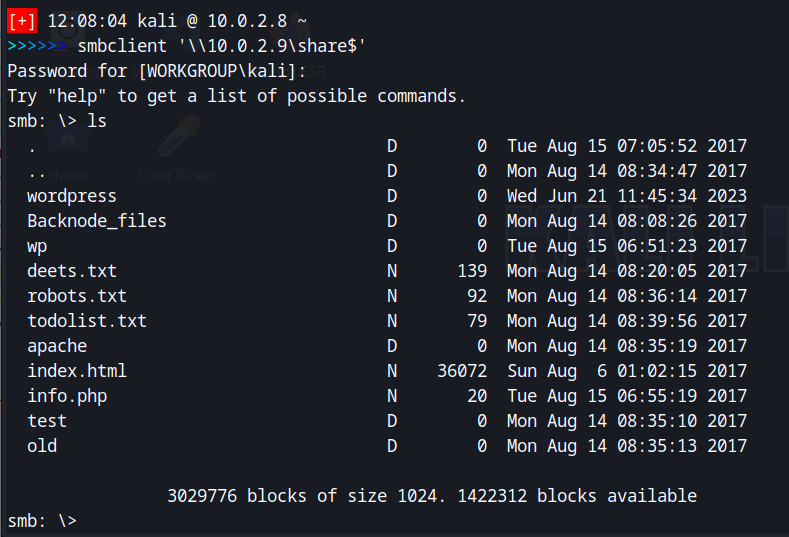
Maybe we can gain access through SMB on port 445. I will use “smbclient -L 10.0.2.9”, which returns:



Let’s try to look at the share$ directory using “smbclient ‘\\10.0.2.9\share$’”. And that returns:

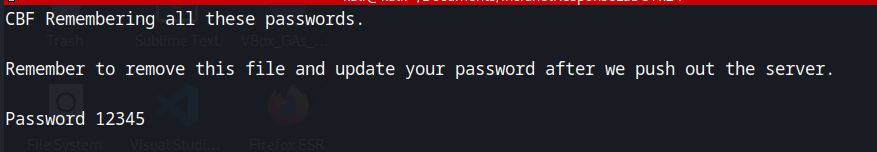


Now that we are in under SMB, let’s look at the files just as we would on the Kali terminal, using ‘ls’:

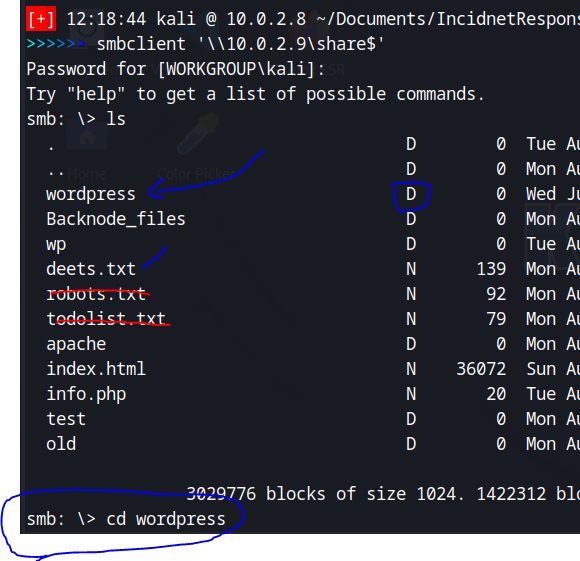


Everything with a “D” is a directory, an N is a file. I can ‘cd’ into each of these directories. I tied using cat and nano to view the files but that didn’t seem to work so I downloaded them using get and then cat each file piping to a less to look at them. Robots.txt and todolist.txt both were useless, however, deets.txt contained a password and msg to delete the file before pushing out the server, which apparently wasn’t done by the admin. This is a security breach we can exploit later??:

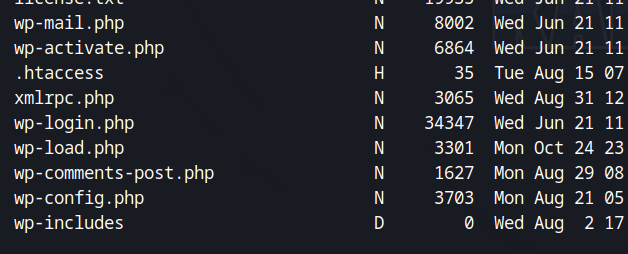




Let’s go into the wordpress directory and see what’s in there:

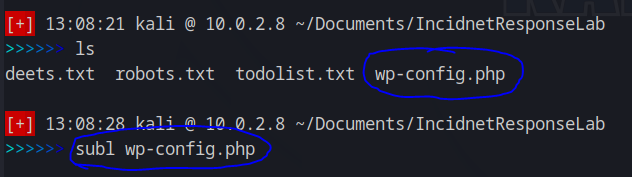


And then if we ‘ls’ that directory, we can see there are a number of files in this directory, but of interesting note, these is a wp-config file in here:

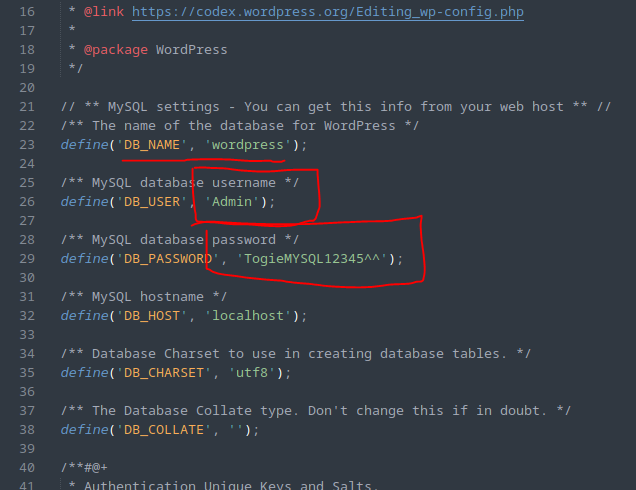


I downloaded that file using ‘get’ and then opened it in my editor of choice, Sublime Text:



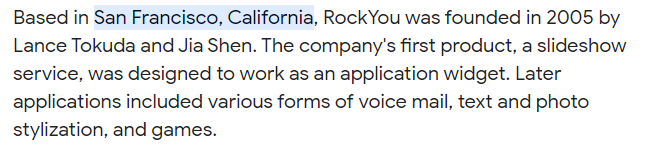


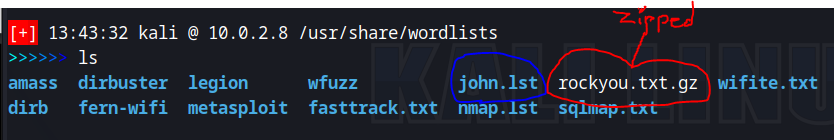
And we have found MySQL (a database) username and passwords. Yet another security breach that can be exploited:



Phase 3 – attempt further intrusion

Hydra is a tool that we can use to brute force (a technique of finding passwords by trying many different common passwords from a database of known passwords) into the website on SSH. We already know SSH is active on the website because we saw it earlier on the ‘nmap -sV’ scan. I will use the rockyou wordlist that comes pre-installed on Kali. Rockyou is a wordlist containing almost 14M different commonly used passwords. The list was acquired when a San Francisco company, RockYou, was hacked and leaked these passwords in plain text form. Since that time, it was incorporated into Kali software and is a very popular wordlist for brute forcing, and can be found in the /usr/share/wordlists directory. Before we do this brute force attack thought, we need to navigate to that directory and unzip the rockyou wordlist. Instead of doing this, I’m going to use a different, smaller wordlist, ‘john.lst’, which is already available in the same directory. Hydra has a couple of options we will use; ‘-l’, which is the login name we want to use, and ‘-P’, which is the wordlist we are going to use.

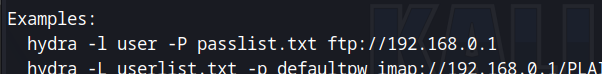




The following is from the help section of the hydra command:

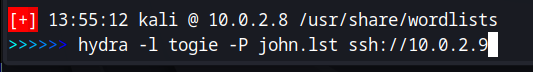


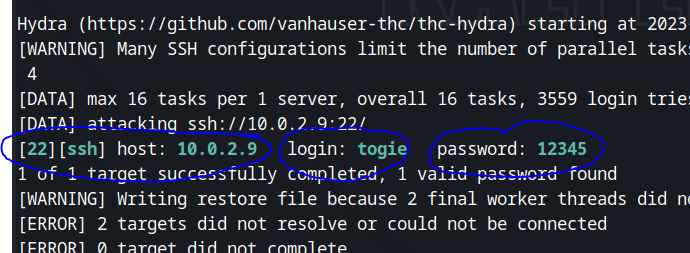




The full command to use will be: hydra -l togie -P john.lst ssh://10.0.2.9

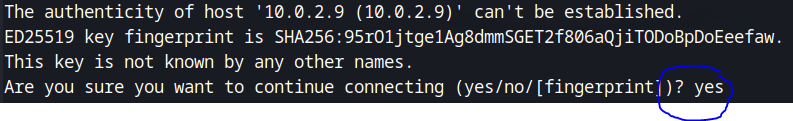
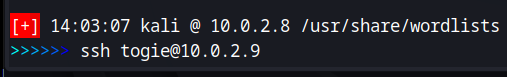
And we can see that Hydra was able to brute force a password for the “togie” user, of “12345”, a very weak password indicating that the admin for LazySysAdmin has not implemented ACLs for strong passwords:

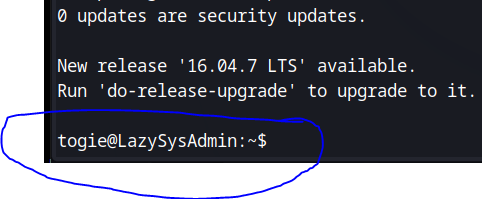




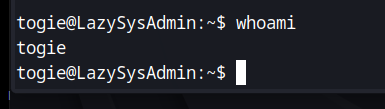
This password is also the same password we saw when we looked at the deets.txt file, so not only was hydra able to find a weak password, but they had stored the password in plain text form in the deets.txt file.

Lets now try to login via ssh using the username togie and password as 12345:



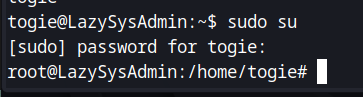


And now we can see that we have been able to successfully login to the server. We can confirm our username using the ‘whoami’ command:



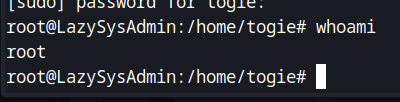
So far we have gone through 3 phases of the attack. Now we attempt Phase 4 – Privilege Escalation

Lets see if togie has the ability to change into a root user by using the command ‘sudo su’:

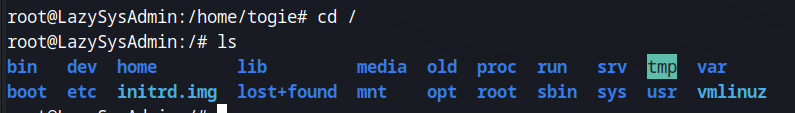


And sure enough, we now are a root user!!! We now have ultimate control!!!

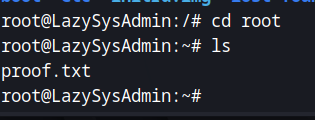
Again using the ‘whoami’ command:



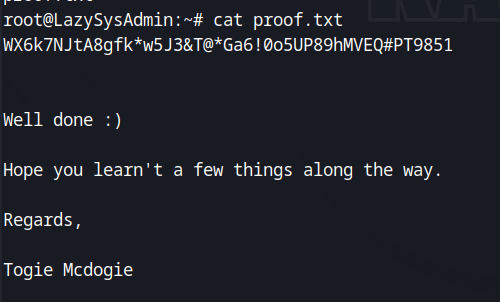
Lets move to the home directory using ‘cd /’ and ‘ls’ that:



And lets see if we can move to the root directory:



We can!! Lets look at this file:



Phase 5 – Persistence

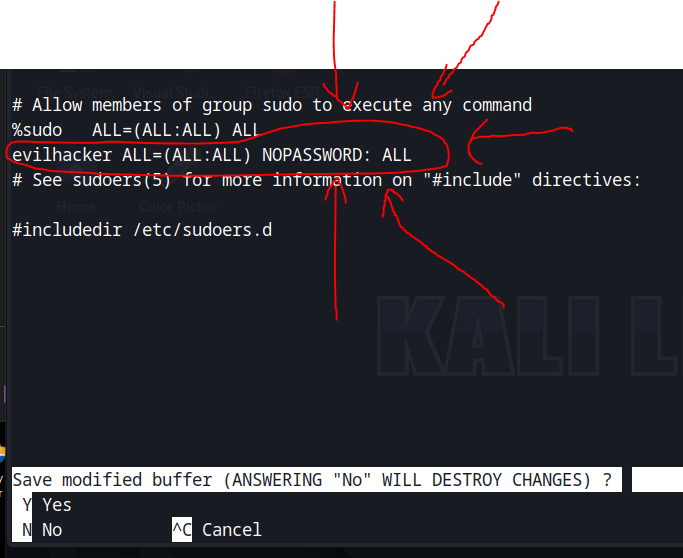
Now we move on to Persistence to make sure we can log back in to this server, even if the admin realizes the error and changes the passwords. What I am going to do is create a new user in the system, who has root privileges. In real life, this name would be obfuscated, but I am going to go with “evilhacker”, lol. So to do that lets use the adduser command:



And I have set the password to “12345”. Now, lets edit the sudoers file and give evilhacker root privileges. To do this, we will use the command ‘visudo’:

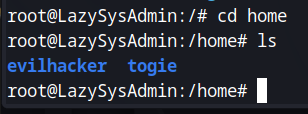


And now I will create an entry for evilhacker and give him sudo privileges under the sudo group section:

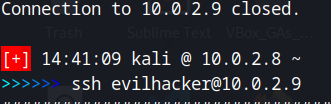


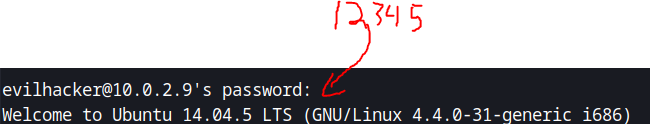
I actually change the entry from ALL=(ALL:ALL) NOPASSWORD: ALL to ALL=(ALL:ALL) ALL

Saving and exiting, then we can ‘cd’ into the home directory and ‘ls’ that and we see that evilhacker has been created!!:

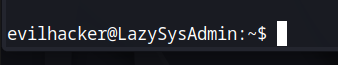


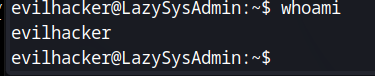
From here, lets test to make sure we can access evilhacker over ssh. I close the current ssh connection and then re – ssh into LazySysAdmin as evilhacker:





And now I am evilhacker in the system!!!!!:



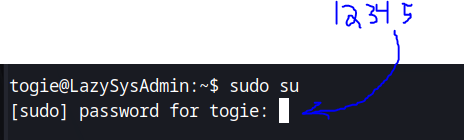


OH YESSSSS!!!!!!!!!!!! We have totally compromised this system as a threat actor!!!!!!!!!

AND NOW, LET’S BEGIN A REAL WORLD ATTACK ON THIS MACHINE!!!!!!!!!!!!

PHASE 1: Change the passwords for the users

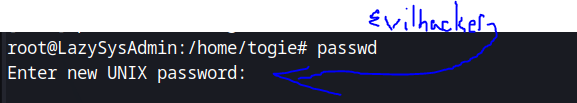
The first thing I’m going to do is move to the root user using ‘sudo su’ from togie’s login:

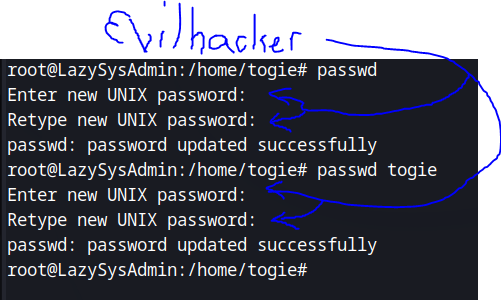


Which brings me to:



Now, I’m going to change root’s password using the command ‘passwd’, and I’ll change it to ‘evilhacker’:



Next, I’ll change togie’s password to ‘evilhacker’ as well using the command ‘passwd togie’. This command specifies the user whose password I am changing by adding the user’s name to the end of passwd:  


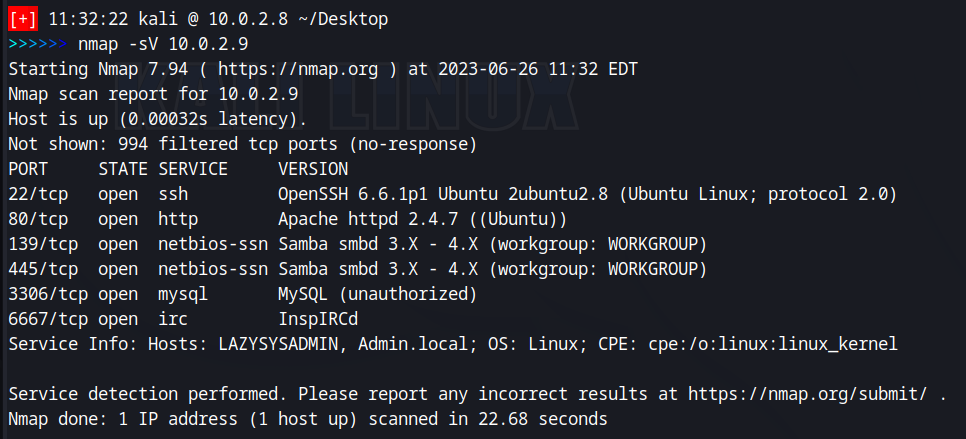
Perfect!!, now we have the following information:

* user: evilhacker password: 12345
* user: root password: evilhacker
* user: togie password: evilhacker

Why did we change these passwords? Because now the admins cannot log into their machine using the old passwords of “12345”. We have effectively blocked their access, and now, only WE (evilhacker) has access to the machine.

Phase 2: Change the website being hosted by the apache service

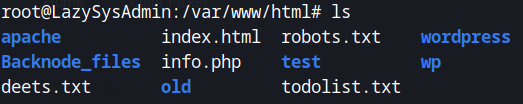
As we saw from detailed scan of 10.0.2.9, using ‘nmap -sV 10.0.2.9’ we have an apache web service running on port 80. We know this because the service listed on port 80 is http, which is the internet:



We can use this to our advantage now by removing the website that Lazysysadmin had placed, and putting in our own, EVIL HACKED, website!!! Let’s do that!, because, ya know, we are an evil hacker….. LOL

Let’s begin this phase of the attack by ssh-ing into the machine using the command ‘ssh root@10.0.2.9’ and entering our new root password of ‘evilhacker’:

And then let’s move to the directory where the website is located using the command ‘cd /var/www/html’ and then ‘ls’ our location:

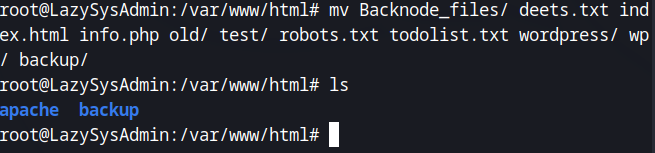


As we can see here, all of the information for the website is here, along with the files we saw in the smb portion of our attack. Now, in the real world, a hacker would probably want something, most likely money, in exchange for “giving back” the admins their website. So as the hacker, we don’t just want to delete everything. Why?, well what if word got out about our attack. The next time we attack someone, they might say, hey well when evilhacker attacked Lazysysadmin, he just deleted Lazysysadmin’s content anyhow, so not only did Lazysysadmin lose the money they paid to evilhacker to get their info back, they also lost their info. If that’s going to happen to us, and we are going to lose our info whether we pay evilhacker or not, why would WE pay evilhacker for our stuff??…. So you can see why a hacker would not want to totally destroy info. So instead of destroying it, lets just move it to a subdirectory.

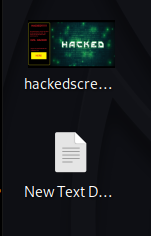
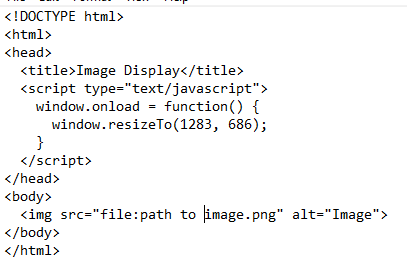
We can do this by creating a subdirectory in the current location using the ‘mkdir’ command:



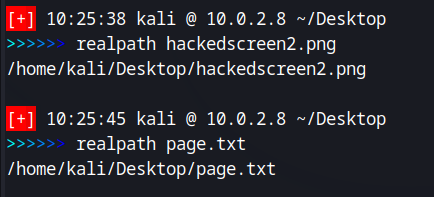
And then using the ‘mv’ command we can move all of the files into that directory, instead of destroying them:



Perfect, now we have a clean html directory to work with for our new page, and Lazysysadmin can still get all his original info back upon his payment to evilhacker!! Next, we need the information for our new “HaCkED!” page, LOL. To do this, I went to the internet and searched for images related to “hacked”. I picked an image, converted it to a .png file using a free online service. And then added a “hacked” msg to the picture. I then resized it to 1283x686 (just a arbitrary size I picked out that fit decently on my monitor, a 27” standard monitor, and didn’t take up the whole screen). I then downloaded the new .png image file to my desktop. I then created a notepad document on my desktop with the following html code:



So now we have an ultra basic website that just displays our “hacked” image, which I will show you in a few more steps. Now we have to 2 files that we will use to put up a new page in place of the old page that Lazysysadmin made. Let’s do it. We need to send these two files to Lazysysadmin. We will do this over the ssh service. First we will need the full location of our files. We can use the ‘realpath’ command to do this:



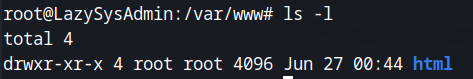
Now we can send these pages to Lazysysadmin using ssh. The command to use for this is:

‘scp /home/kali/Desktop/hackedscreen2.png [root@10.0.2.9:/var/www/html](mailto:root@10.0.2.9:/var/www/html)’

In order to run the scp command, you must be logged out of Lazysysadmin. This command is run from the terminal of your Kali machine

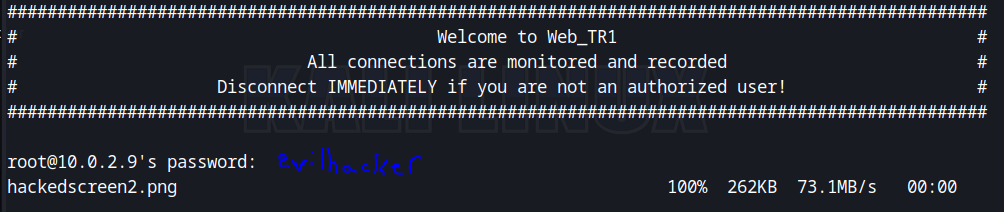


Let me break this command down for you. Scp is a secure file transfer tool for the ssh service. Essentially, we are just sending files. Scp calls the service. Next is the /home/kali…. portion. This is simply the full file path of the file that we are going to send via ssh. Next is the [root@10.0.2.9](mailto:root@10.0.2.9)... portion. This is the user that we are going to send the information to. And finally, the :/var/www.html. This is the location where we are going to send the files to. Think of this scp command as a move command. They are very similar in what they do. Now, why did use the root user? Well, if we look at the permission of the html directory that we are moving these files into, we can see that that the root user is the only one with permissions to edit this directory:

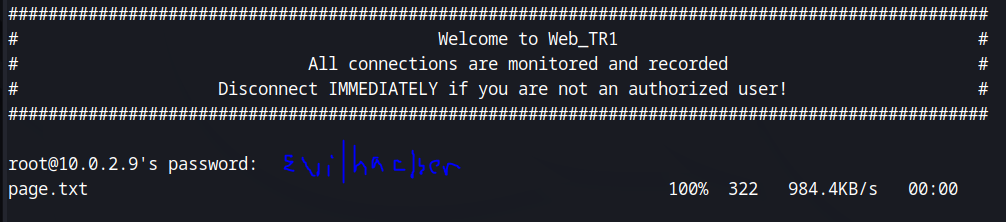


So we need to be root in order to move our files into this directory. When using the scp command, it may ask for root’s password, which we already know is ‘evilhacker’ because we changed it previously in our attack in Phase 1. So here is the commands ran in order:

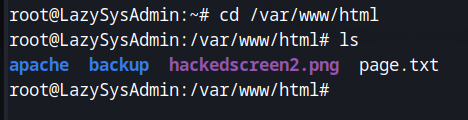




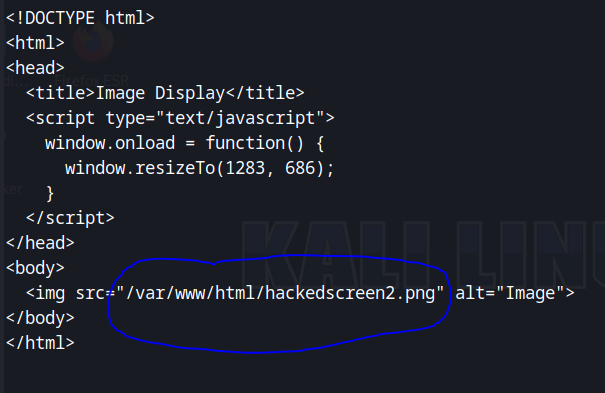




And if we again ssh into Lazysysadmin using ‘ssh root@10.0.2.9’ and ‘cd /var/www/html’ directory, we now see that both of our files have been moved to the Lazysysadmin machine:

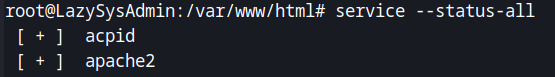


Next we need to rename page.txt to index.html. This is a simple ‘mv page.txt index.html’. Now we need to edit out index.html file to properly load the “hacked” image we made. I used the text editor nano for this as it was simple change, even though my editor of choice is Sublime Text, which you will not have installed on your Kali machine. The command is ‘nano index.html’:

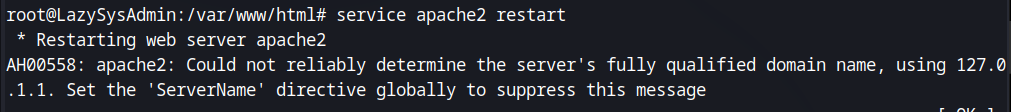


We are going to change the img=src variable to the location of our “hacked” image, as is circled in blue. Hitting ‘ctrl x’ will exit the editor, a msg will pop up at the bottom asking to save the changes, just hit ‘y’ and then ‘enter’.

Great!, now if you type ‘service –status-all’ in the command line, it will display a list of all services running on Lazysysadmin, and as you can see, apache2 is one of the first services listed. This is the service that is running the website:



We need to restart this service so that our new webpage will take effect. The command for this is very simply, ‘service apache2 restart’:



Apache will restart and now we will have our HaCkEd website up and running!!!!! We can check this by opening firefox on our kali machine, and instead of typing a website address in the address bar, just type the ip address of the Lazysysadmin machine, 10.0.2.9:



HAHAHAHAAHAHAHAHAHA!!!!!!!!!!!!!!!!!!!!

EAT THAT LAZYSYSADMIN!!!!!!!!

THAT’LL TEACH YA TO NOT HAVE GOOD SECURITY PRACTICES!!!!

EVIL HACKERS ARISE!!!!!!!

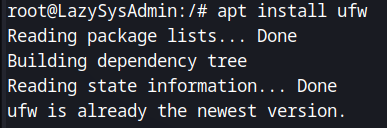
OK….. going a little overboard there… anyhow, lets get back to our attack, WE AREN’T DONE YET muahah! We got more hacking to do!!

Let’s move on:

Phase 3: Locking down the machine

Now we need to lock down the Lazysysadmin machine so that the admins cannot perform the same attack we used to gain access in the first place. This should have been done by the admins in the first place, but, they didn’t, so now we are going to implement security controls of our own!

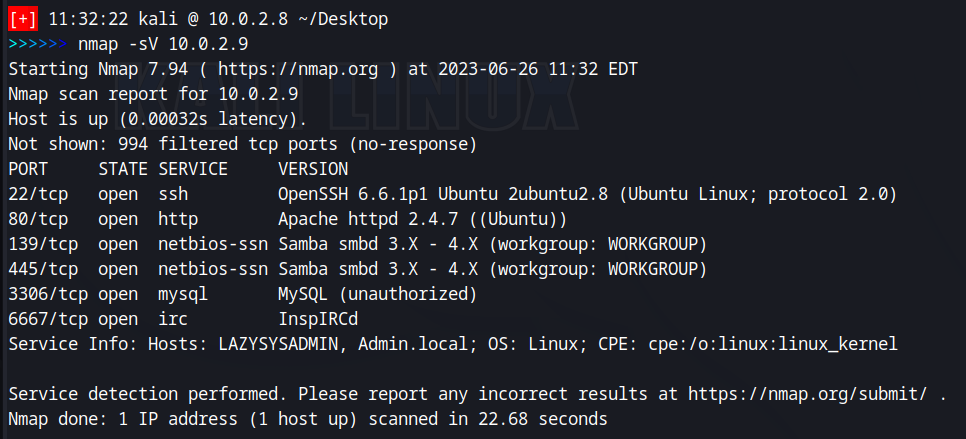
In a prior assignment, we used the firewall UFW. We are going to use that again here so this might be a bit of a refresher. Now I use UFW a little differently than we did in the UFW assignment, just because I find it easier. I didn’t check to see if UFW was installed on Lazysysadmin before I tried to install it as the root user, but as you can see, it’s already there for us, ready to go:



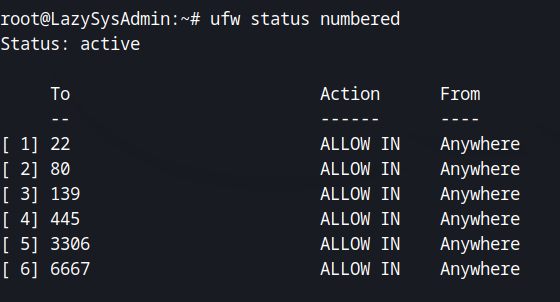
And note that I am on the Lazysysadmin machine as the root user. I have ssh’d in this way as I previously described. We can check UFW’s status using ‘ufw status’:



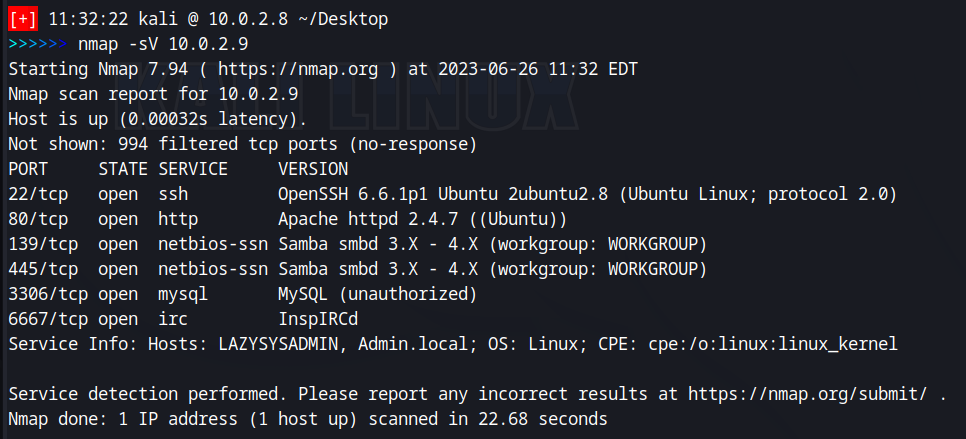
And we see it is inactive. Let’s activate it by typing ‘enable ufw’. This will make UFW start up every time Lazysysadmin is booted. Having UFW inactive is bad security practices for the admins. Let’s abuse this for our advantage as the hacker! If we go back to our ‘nmap -sV …..’ command, you will remember that the following ports were open:



So let’s mirror this in UFW b entering the following command, ‘ufw allow 22’. This will set a rule in UFW to allow activity on port 22. As a side note, it will also allow a rule on port 22 with an IPv6 portion. Essentially this will set up 2 rules. I then entered ‘ufw status numbered’ to see these rules. You will see two rules there, each with a number next them. I then removed the IPv6 rule by typing ‘ufw delete 2’ which you will see as the rule for the IPv6 version. Once you have entered the rule for each port, and then deleted the IPv6 rule for each port, you will be left with the following rules (type ‘ufw status numbered’ to see your rules):

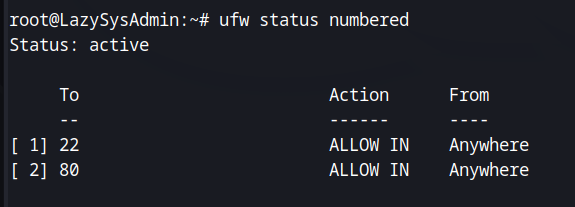


Now if you re-run the namp scan from your Kali machine, you will see that the same services are up and running on Lazysysadmin:



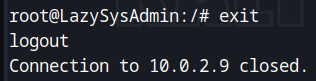
Excellent, now comes the lockdown portion. First let’s identify the ports we still need. We need ssh access to the machine, as that is the only way we want to be back in, so let’s leave port 22 open. We DO NOT want to get in thru SMB as we did in our original attack, so we need to shutdown the SMB ports 139 and 445. We want http up and running so our “HaCkEd” website will still be active, so let’s leave port 80 up and running. We DO NOT want the admins to access any other portion of their info, so let’s shutdown ports 3306 and 6667 as well. This leaves us with only ports 22 for SSH and 80 for HTTP. We know the admins cant get in on SSH because we changed their password LOL. After the admins pay evilhacker, evilhacker will give them the root users new password, and they will be able to access all their info, and set their site back up!

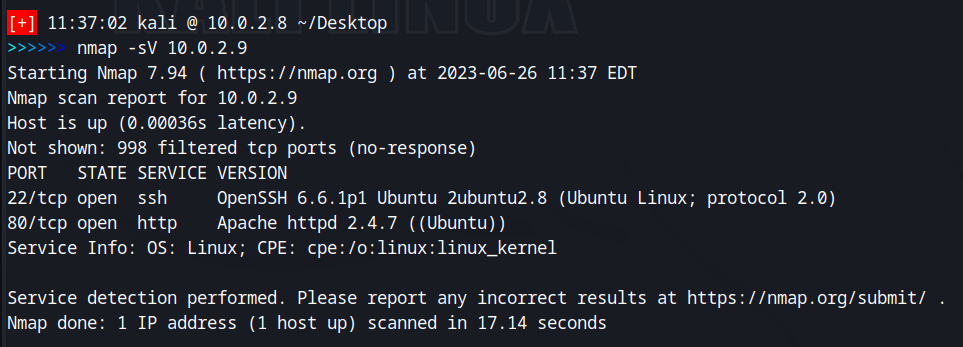
Ok so how do we shut down ports 139, 445, 3306, and 6667? Easy, all we are going to do is go back into Lazysysadmin as the root user and enter the command ‘ufw delete (the number of the rule corresponding to the port)’, so for SMB ports 139 and 445, that is rule 3 and 4 as seen above. ‘ufw delete 3’ gets rid of port 139. The rules below that move up in order. So rule 4 which was 445, now becomes rule 3 for 445, as the previous rule was deleted. We enter ‘ufw delete 3’ again and that gets rid of 445. The rules for the rest of the ports reorder themselves again and as you can see, we just keep entering ‘ufw delete 3’ until we don’t have anymore rules other than 1 and 2 which are ports 22 and 80 respectively. After each time you enter ‘ufw delete 3’, you check the reordering by entering ‘ufw status numbered’ to get an updated listing of the rules. And finally, we should be left with the following rules in place:



YES!!!! We have effectively locked the admins out of their own machine HAHAHAAHAHAH!!!!!

Now let’s logout of the Lazysysadmin (by typing ‘exit’) machine, go back to our Kali machine, and check our status. Let’s run the ‘nmap -sV 10.0.2.9’ command again and see what we get:



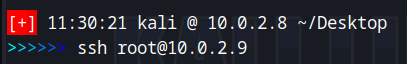


And as you can see, we are left with just the website on port 80, and our ssh access on port 22!!

And if we open firefox and navigate to the web address 10.0.2.9:



Oh yea!! Our Hacked website is still up. And we can still login on SSH:







**SUCCESS!!!!! EVIL HACKER NOW HAS THIS SITE UNDER OUR CONTROL!!!!!**

**HEY ADMINS, IF YOU GIVE US $1000.00 USD IN MONERO COIN, WE’LL GIVE YOU YOUR SITE BACK!!**

**MUAHAHAHAH!!!!!!!**

**And that is how a real world attack happens!!!**

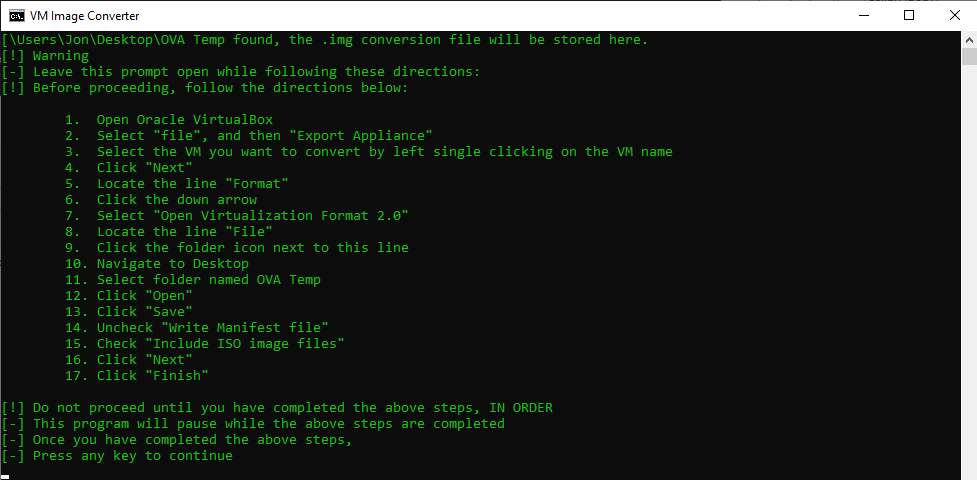
Man, this has been a fun assignment, and a fun machine to attack!! I hope you guys learned something from this. Now go to my github, download my VM\_Img\_Converter tool and run Autopsy and see if you can find the Indicator’s of Attack that we used on this machine. That will complete the assignment!!!

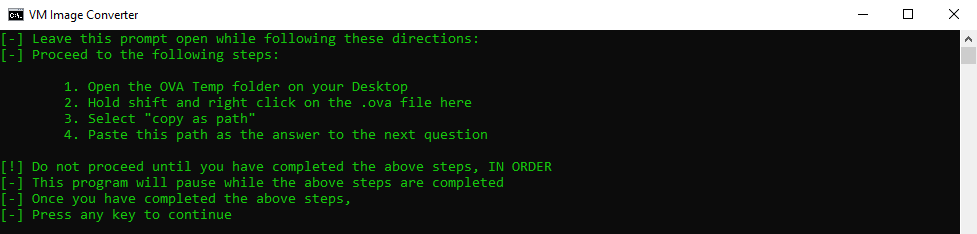
**Part 2: Autopsy**

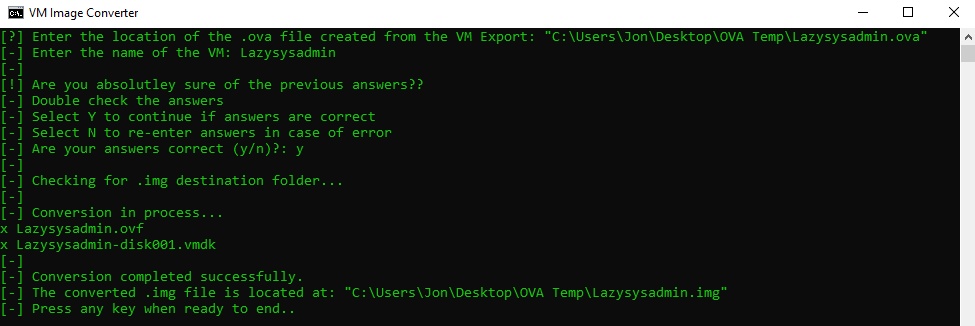
Now I am moving on to Autopsy in Kali Linux. Let’s see if we can find some indicators of compromise to see what the attacker did from the admin standpoint.

I wrote a batch script tool called VM\_Img\_Converter and used it to create an .img file of the Lazysysadmin machine

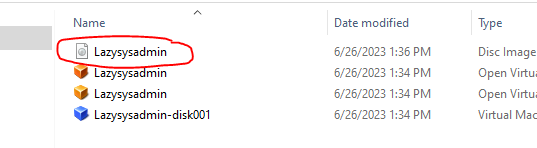
Here’s a few pics of that tool:



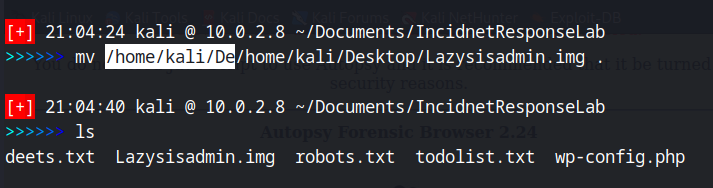




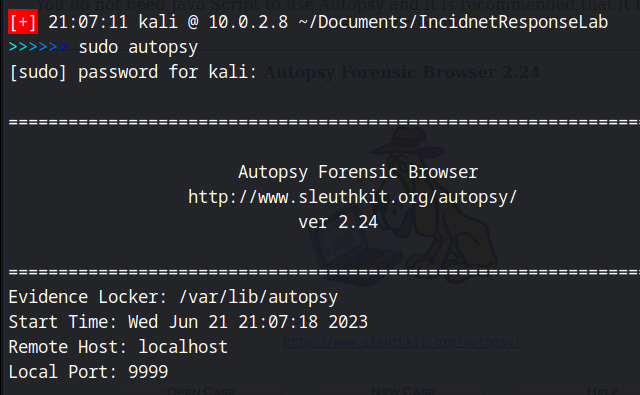
And the result is an image file that has been created that we can use with our Kali machine in Autopsy:

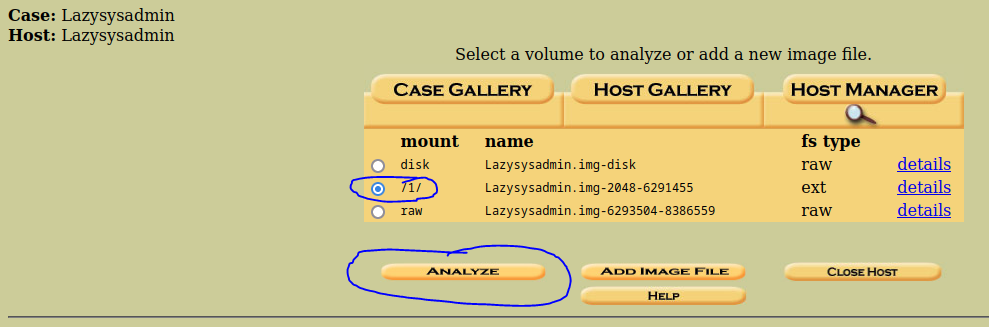


Now let’s spin up the Kali machine and place this image in our assignment directory:

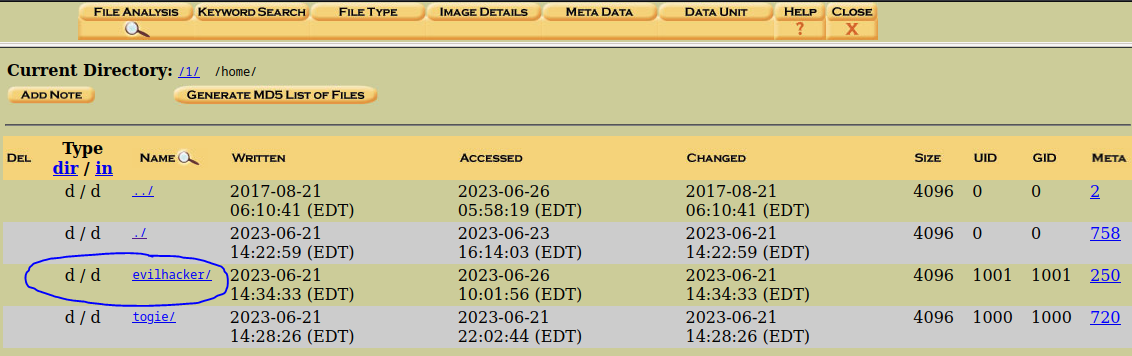


We can use ‘sudo Autopsy’ and start a new case on this image:





If we move to the home directory, we can see our 1st IoC, a new user called evilhacker. That’s not good!:

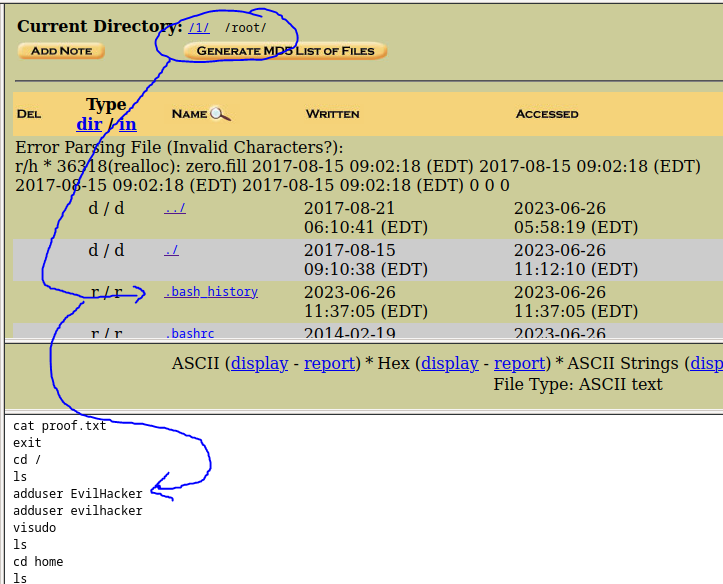


If we look at that users bash history, we can see commands that he was running:

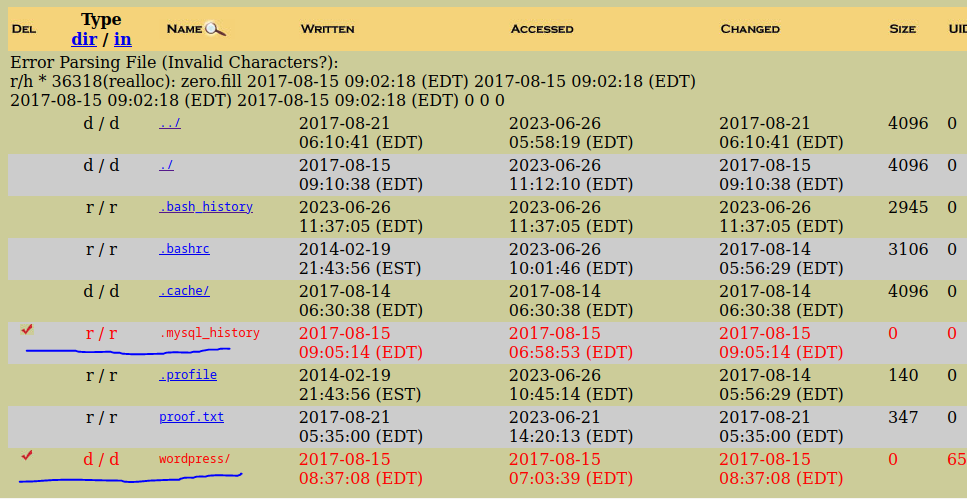


Looks like he was switching to the root user!!!! That’s REALLY not good!!!

Let’s see what the root user was doing by looking at his bash history:

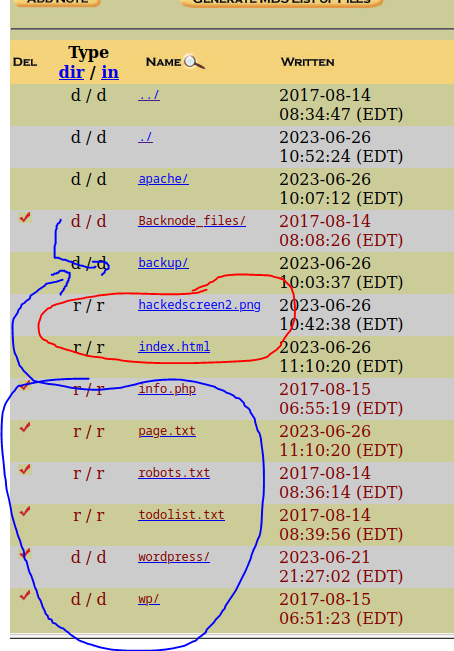


And here we can see where the user evilhacker was created!! Oh now we have an unauthorized user on the system! Let’s check our directories and see if anything happened:



Looks like tampering occurred on the sql and wordpress directories

Let’s dive further into our directories and see what happened

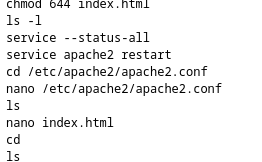
 

Looks like two files were created, a new website called index.html and a “hackedscreen2.png”. Also a subdirectory called backup was created and all of our web files were moved to that directory!!! The hacker must have put up a new website!!

Lets check out some logs to see more about whats happening!

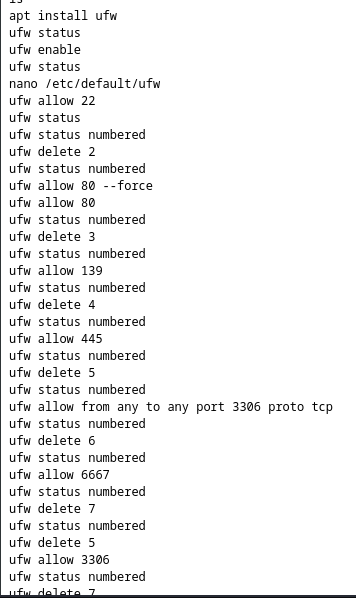


Here’s the creation of the backup subdirectory and the movement of the files.



And here’s some activity where permissions were changed.

Here’s some more logs where a firewall was put up and ports were manipulated!!!



And here’s some logs from the ssh service showing unauthorized activity where the hacker was doing malicious things!!!



Conclusion:

As we can see from Autopsy, someone gained access to our webserver/machine and made unauthorized changes to our system. They reset our passwords, created a new user, removed permissions from our files and gave themselves permissions of their own. The hacker was able to gain root access and take down our website, installing a new website of his own. Then he manipulated the firewall to block us out of our own system!!!

How did all this happen? The admins failed to implement good security practices, left passwords in plain text form in files that were easily accessible. And eventually that led to a hacker gaining unauthorized access to the server where he took total control as a root user!!!!

This all could have been prevented by using good security practices. Now the company may have to pay $1000 to this hacker to get their information back!!!