**Przygotowanie stanowiska testowego**

Do celów testowych korzystam z Systemu Ubuntu Desktop 20.04.3, początkowo z jądrem 5.11.0-105,

Konto:

Login: client

Password: password

Pomimo, że teoretycznie podatne są jądra od 5.8 do 5.16 próba dla tego kernela zakończyła się niepowodzeniem

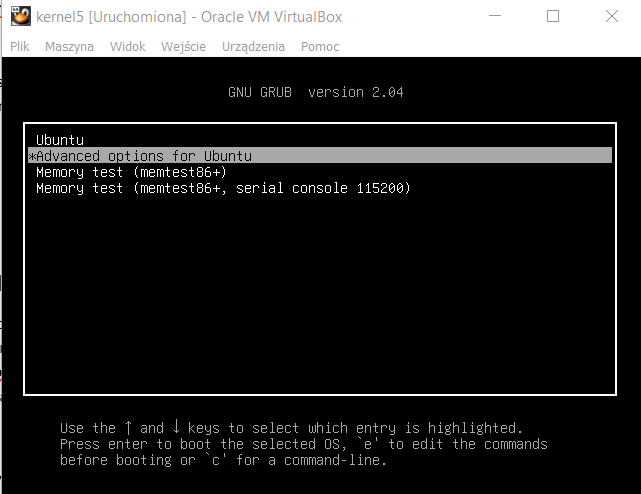
W związku z tym pobrałem kernel w wersji 5.11.0-27-generic używając komend:

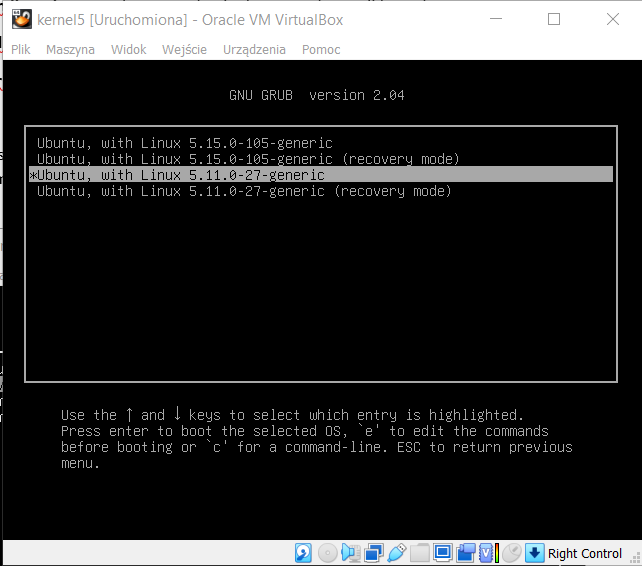
**sudo apt search linux-image** – do przeszukania dostępnych wersji kernela

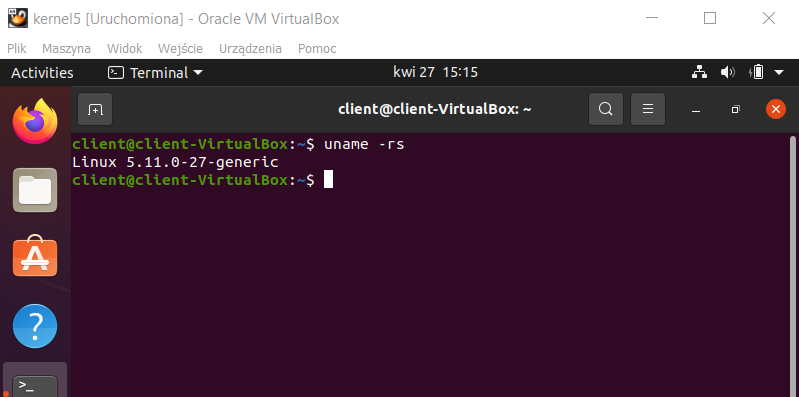
**sudo apt install linux-image-5.11.0-27-generic**

**sudo update-grub** – aktualizacja bootloadera

Natępnie po restarcie systemu, w czasie jego uruchamiania użyłem klawiszy „shift” oraz „esc” w celu uruchomienia menu bootloadera oraz wybrałem odpowiednią wersję kernela w „Advanced options for Ubuntu”





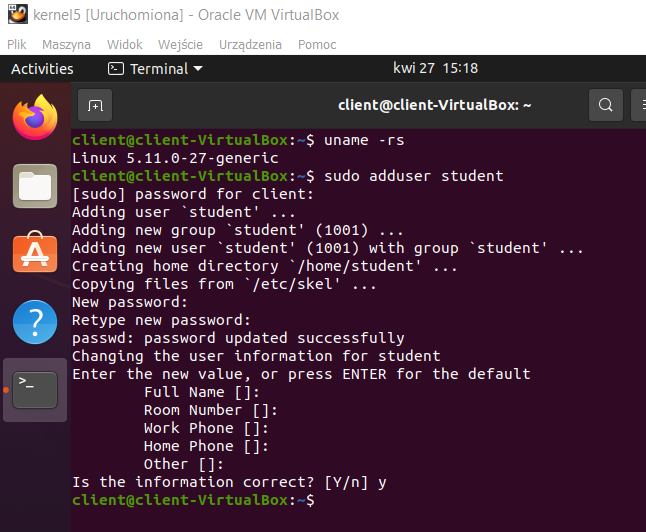


**Na czym polega ta podatność**

Podatność ta pozwala na eskalację uprawnień oraz nadpisywanie danych w plikach tylko do odczytu przez nieuprawnionych użytkowników. Problem wynika z nieprawidłowego zarządzania buforem strony w Linuxowym systemie plików i pozwala na manipulację danymi w pamięci. Luka ta mogła być wykorzystywana do zdobycia uprawnień roota w systemie.

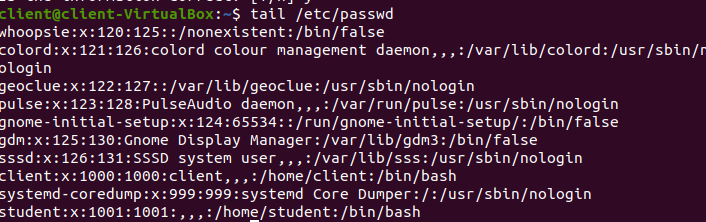
**Testowanie exploita nr.1**

Stworzyłem użytkownika „student” bez uprawnień



**Login:** student

**Haslo:** student



**Exploit 1: Zmiana hasła roota w pliku /etc/passwd**

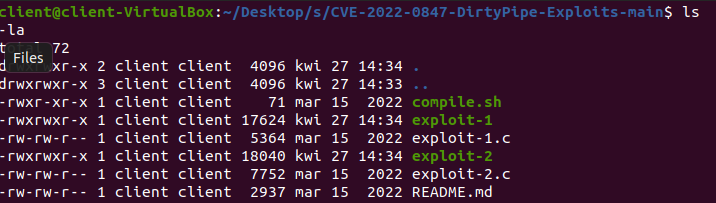
**Exploit ten pozwala nadpisywać pliki tylko i wyłącznie do odczytu,takie jak np. /etc/passwd**

Hasło zostanie zmienione na ‘piped’ oraz pozwala wpisywać w czasie wykonywania programu komendy jako **root**. Dla celów testowych, exploit po zakończeniu działania przywraca stare ustawienia.

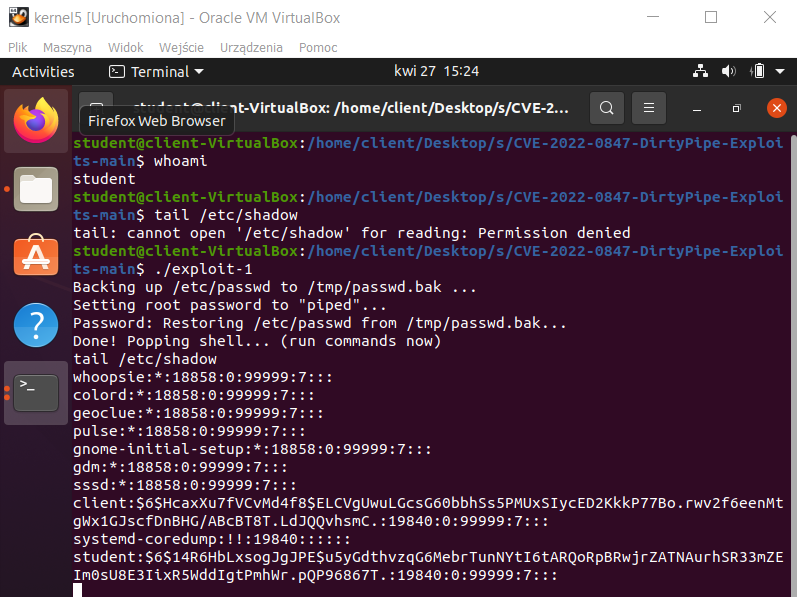
Najpierw należy skompilować kod exploita, ponieważ jest on napisany w jezyku C.

Zrobiłem to użyciem gcc

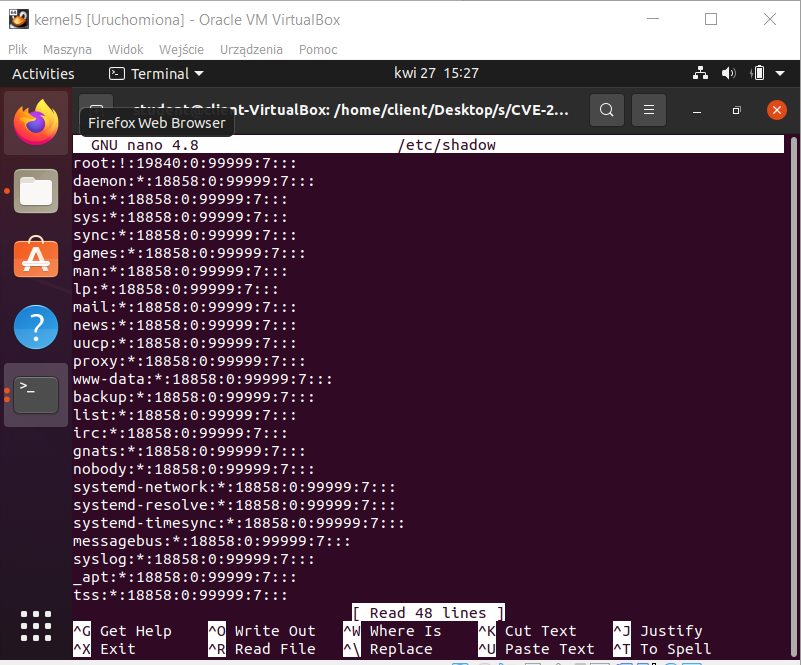
Po kompilacji, do pliku binarnego wystarczy uruchomić exploit-1



**Demonstracja działania exploita-1**



Początkowo jako użytkownik bez uprawnień, student, nie można wyświetlić zawartości pliku /etc/shadow, natomiast po wykonaniu **exploit-1** , jest to możliwe. Plik ten zawiera zahaszowane hasła użytkowników

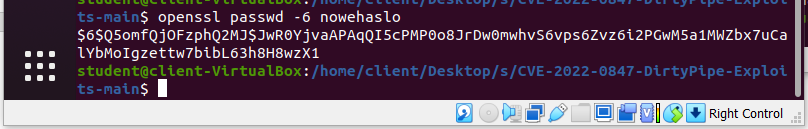


Można dzięki temu zmienić hasła dla każdego z użytkowników

Wystarczy zahaszować przy pomocy algorytmu sha512 oczekiwane hasło oraz zapisać je w pliku.

**Można to zrobić przy pomocy komendy:**

openssl passwd -6 student



* gdzie **6** to haszowanie 512,
* **student** to hasło które chce zahaszować

**Testowanie exploita nr.2**

**User:**

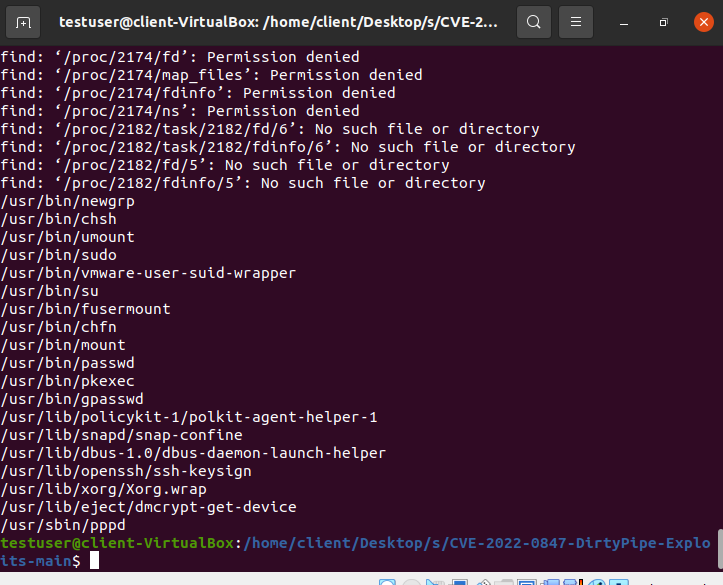
(z katalogiem domowym, bo zapomniałem)

**Login:** testuser

**Password:** testuser

**Umożliwia nadpisywanie plików binarnych SUID, które są „read-only”**

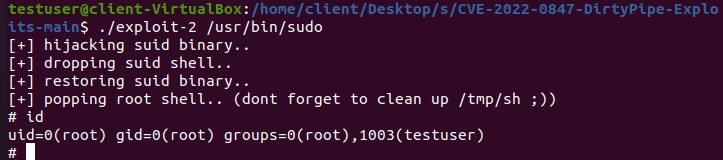
Szukanie SUID binaries



Wyświetla to pliki, które maja ustawiony bit SUID, gdy jest ustawiony na pliku wykonywalnym, powoduje, że program uruchamia się z uprawnieniami właściciela pliku, a nie uprawnienami użytkownika, który go uruchamia

**Uruchamianie exploit 2**

./exploit-2 /usr/bin/sudo



Wykonałem polecenie sudo – bez wprowadzania hasła