**Na czym polega podatność oraz gdzie występuje**

Podatność jest związana z konfiguracją **sudo**.

Jest ono narzędziem umożliwiającym użytkownikom uruchamianie poleceń z uprawnieniami innego użytkownika, najczęściej roota.

Błąd dotyczy wersji sudo < 1.8.28.

Problem występuje, gdy w pliku sudoers wpisano specyfikację, która pozwala użytkownikowi uruchamiać polecenia jako dowolny użytkownik, ale z wyłączeniem użytkownika root

Ważne by użytkownik miał cząstkowe uprawnienia sudo.

W pliku w którym konfigurowane są uprawnienia prezentuje się to w sposób następujący:

root alexis (ALL, **!root**) /usr/bin/id (lokalizacja programu-id)

W normalnej sytuacji użytkownik nie powinien być w stanie wykonać polecenia jako root.

Dla tych wersji sudo użytkownik może ominąć to ograniczenie określając identyfikator użytkownika (UID), który jest równy 0, w nieoczekiwany sposób. Wystarczy wykonać komendę sudo z opcja –u:

sudo –u#-1 (jakas komenda,np. powerOff)

lub

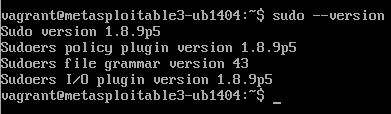
sudo –u# 4294967295 (jakas komenda)

W systemach z 32-bitowym UID, takie cyfry są traktowane jako 0. Czyli UID roota.

**Wykorzystanie podatności**

Testy wykonano na **metasploitable3**, Ubuntu (maszynie wirtualnej zawierającej podatności do celów treningowych)

**Wersja sudo**



Każdy użytkownik ma swój numer UID(user id) GID(group id) , własny home directory oraz user shell.

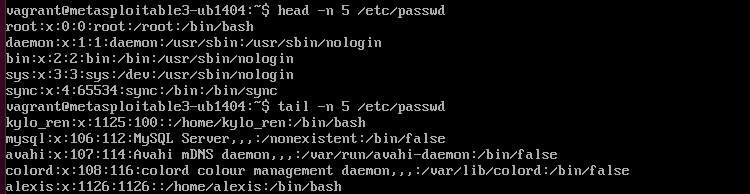
**Numer 0** - root

**Numer 1 do 999** – Konta systemowe, takie na które nie można się zalogować, dla usług

PowłokaShella = (/nologin lub /false)

**Numer 1000** – wzwyż – Użytkownicy

**Informacje te dostępne są w pliku /etc/passwd**



Przykładowy wiersz:

root:x:0:0:/root:/bin/bash

Znaczenie kolumn:

userName:hasłoJestZahaszowane:UID:GID:/homeDirectory:/powłokaShella

**Hasła są zahaszowane w pliku etc/shadow**

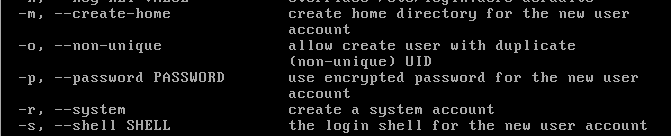


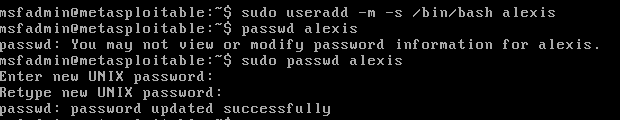
$6$ oznacza haszowanie przy pomocy SHA-512

<https://www.baeldung.com/linux/shadow-passwords>

**Na potrzeby testu przygotowano odpowiedniego użytkownika**

Userradd:





**Użytkownik:**

Login: Alexis

Password: Alexis

**Dodano odpowiednie uprawnieni użytkownikowi w pliku: sudoers**

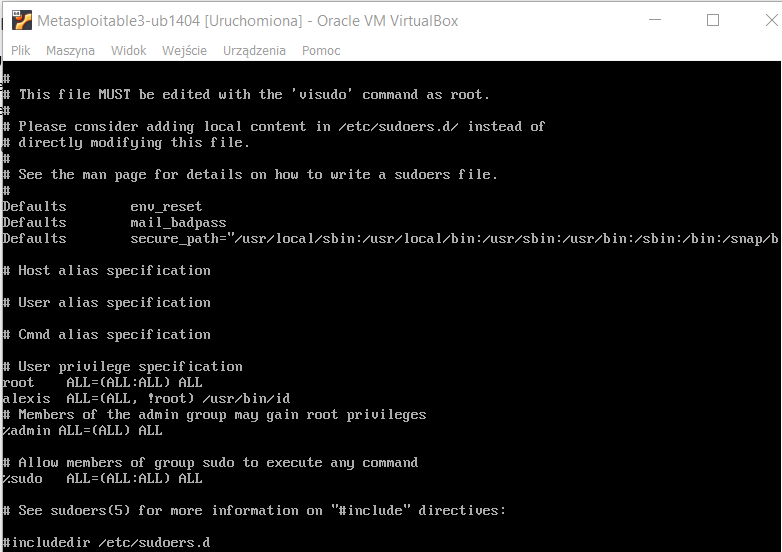
Jest to plik, w którym są wymienieni wszyscy użytkownicy, którzy mogą korzystać z sudo oraz programy jakie mogą przy jego pomocy uruchomić

<https://pl.wikipedia.org/wiki/Sudo>

**Plik sudoers**

Użyto polecenia „visudo” do edycji, by zmienić plik bezpiecznie. Komenda ta zapewnia, że tylko jeden user obecnie modyfikuje ten plik.

Oprócz tego polecenie zapewnia, że żadne niepoprawne konfiguracje nie zostaną zapisane.



**W sekcji User privilege specification:**

*root ALL=(ALL:ALL) ALL*

root- reguła dotyczy roota

ALL(pierwsze) – może wykonywać z wszystkich hostów

ALL(drugie) - jako każdy użytkownik w systemie może uruchamiać polecenie

ALL(trzecie) - z każdej grupy

ALL(czwarte)- dozwolone polecenie



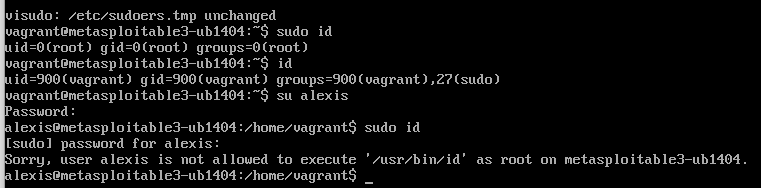
Możesz uruchomić komendy jako każdy użytkownik z wyłączeniem „root”, komendę id

Do celów demonstracji, przypisano uprawnienia do uruchomienia komendy id. Przy pomocy exploita uruchomiona zostanie inna komenda (inna niż zdefiniowana id)

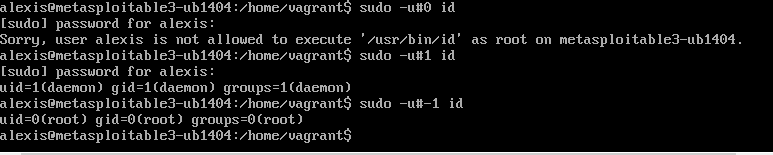
**Testy**

Oczekiwane działanie, bez wykorzystywania podatności

Test dla użytkownika vagrant, root, oraz alexis(Stworzony user do celów testowych)



Specyfikacja usera(-u), najpierw jako root, potem jako deamon, na końcu korzystanie z podatności, wpisanie -1 i wykonie polecenie jako root .

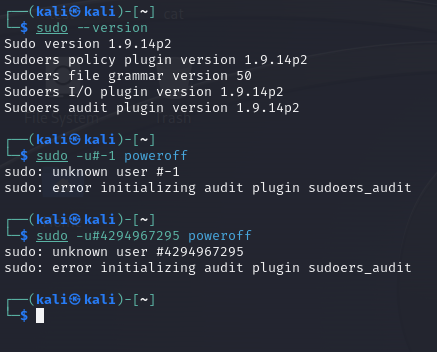


Podatność:



**Mitygacja?**

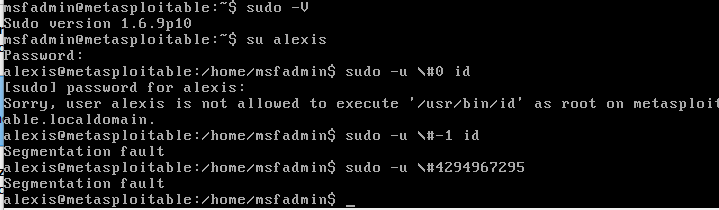
Aktualizacja sudo.



**Ciekawostka**

Starsze wersje niż 1.8 sudo nie miały tego problemu również, np. 1.6.

Tutaj przykład z Metasploitable2

man

Dodano escapowanie znakow, zgodnie z manualem.

