DC-1

1. Rozpoznanie

Pierwszym etapem było zidentyfikowanie adresu IP maszyny w sieci lokalnej oraz sprawdzenie otwartych portów.

Skanowanie portów

Wykonano skanowanie narzędziem **Nmap** z flagą `-sV` w celu identyfikacji wersji usług: `nmap -sV [IP_OFIARY]`

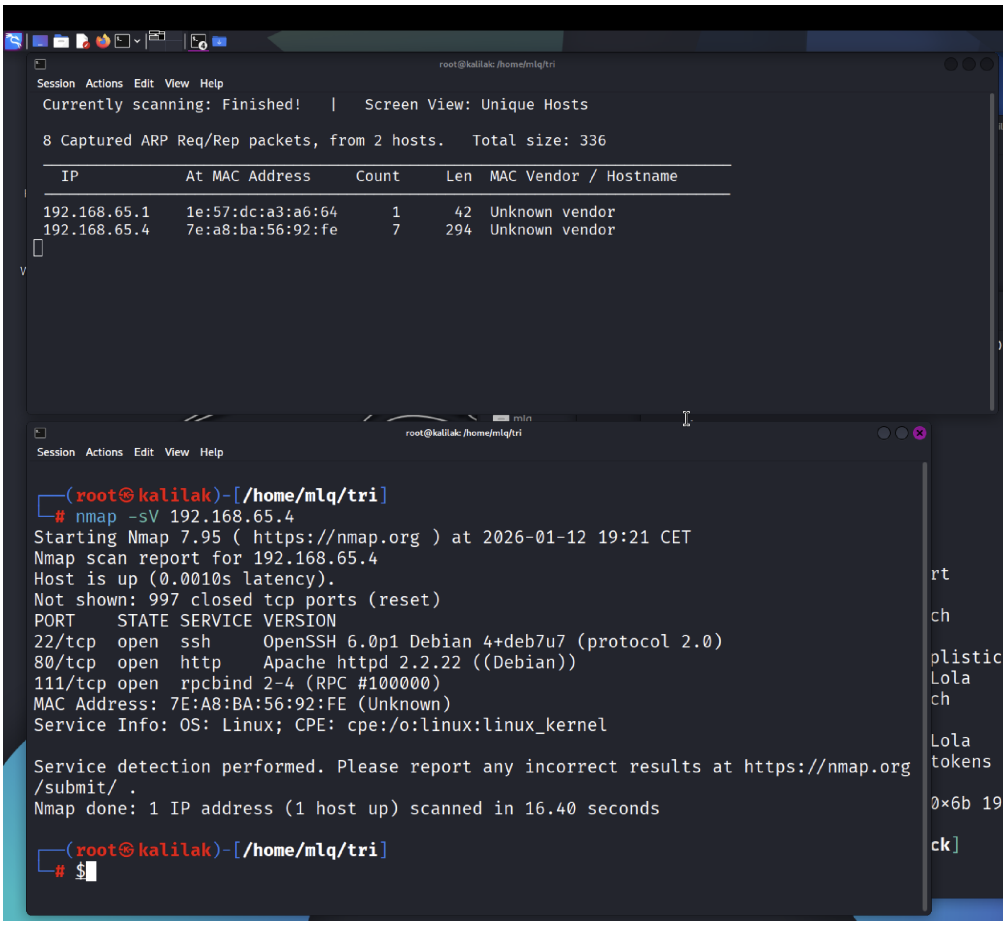

Figure 1: Rysunek 1: Wynik skanowania Nmap - otwarty port 80 (HTTP) oraz system CMS Drupal 7

Wynik skanowania ujawnił otwarty port **80 (HTTP)** działający na serwerze Apache oraz informację o systemie CMS: **Drupal 7**.

Analiza podatności webowych

W celu potwierdzenia wersji i znalezienia potencjalnych błędów konfiguracyjnych użyto skanera **Nikto**: `nikto -h [IP_OFIARY]`

Narzędzie potwierdziło, że mamy do czynienia z przestarzałą wersją Drupala 7, co sugeruje wysoką podatność na znane ataki.

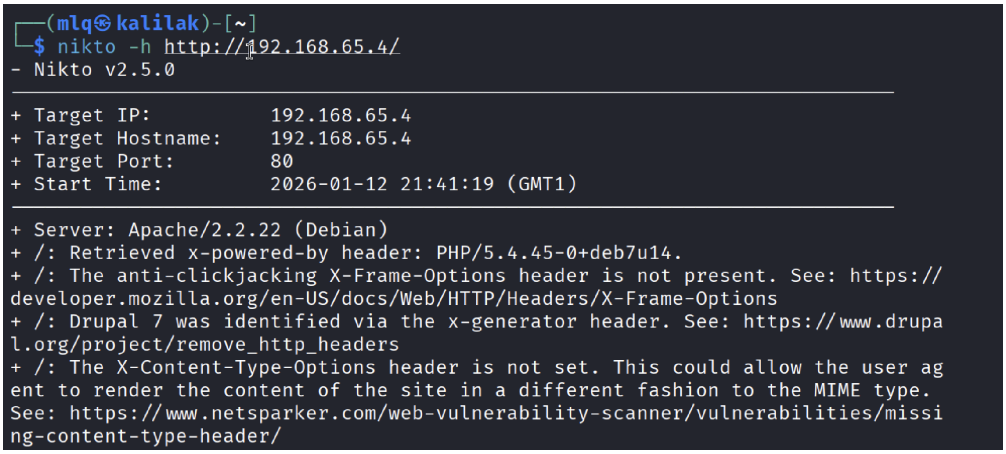

Figure 2: Rysunek 2: Skanowanie Nikto - potwierdzenie przestarzałej wersji Drupal 7 i wykrytych podatności

2. Eksploitacja - Uzyskanie dostępu do panelu

Przeszukano bazę exploitów pod kątem Drupala 7. Znaleziono podatność pozwalającą na wstrzyknięcie SQL lub zdalne wykonanie kodu (Drupalgeddon).

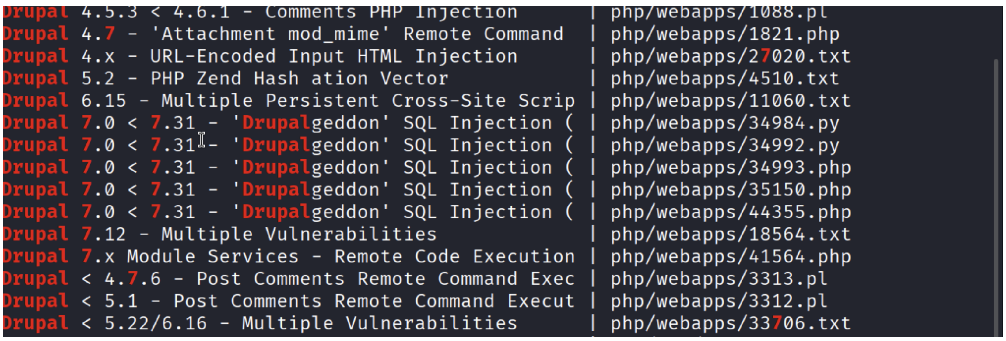

Figure 3: Rysunek 3: Wyszukiwanie exploitów dla Drupal 7 - znalezienie podatności Drupalgeddon

Wybrano exploit pozwalający na utworzenie nowego konta administratora poprzez manipulację bazą danych (SQLi). Uruchomienie skryptu:

Atak zakończył się sukcesem. Hasło administratora zostało zresetowane/dodano nowego użytkownika, co pozwoliło na zalogowanie się do panelu webowego.

Zdobycie Flagi nr 3

Po zalogowaniu się do panelu administracyjnego (GUI), w sekcji **Dashboard** odnaleziono jedną z flag widoczną tylko dla zalogowanych użytkowników.

---

3. Eskalacja do powłoki systemowej

Dostęp przez stronę WWW był niewystarczający. W celu uzyskania dostępu do terminala (CLI) zdecydowano się użyć exploita **EDB-44449** (Ruby), który pozwala na zdalne wykonanie kodu (Remote Code Execution).

Figure 4: Rysunek 4: Uruchomienie exploita SQLi - wstrzyknięcie SQL w celu dodania konta administratora

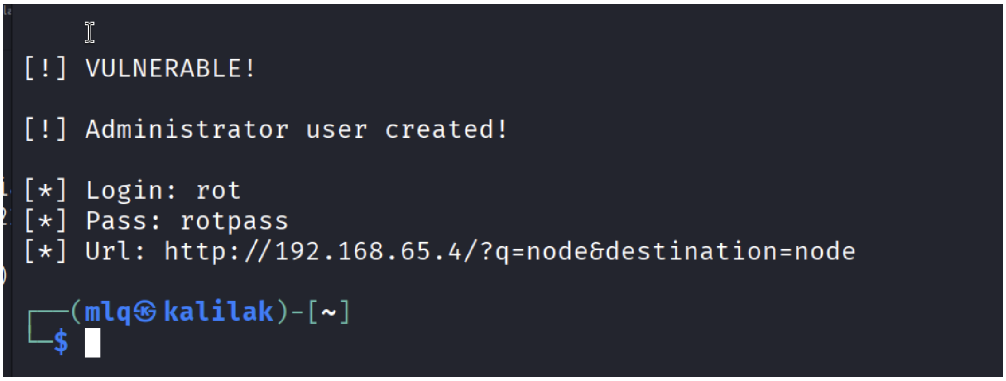

Figure 5: Rysunek 5: Sukces exploita - potwierdzenie utworzenia/zmodyfikowania konta administratora

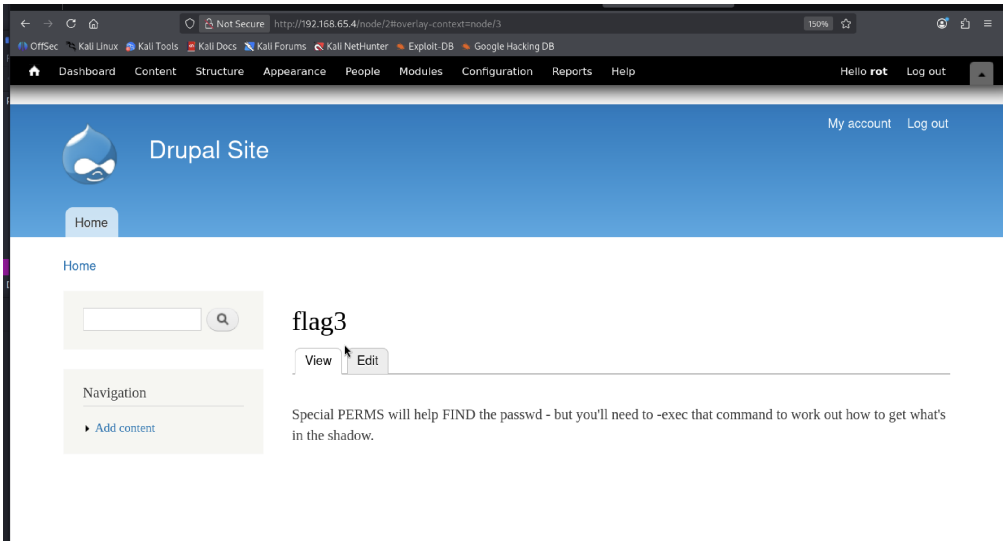

Figure 6: Rysunek 6: Flaga nr 3 widoczna w Dashboardzie po zalogowaniu jako administrator

Problemy z zależnościami

Podczas próby uruchomienia exploita wystąpił błąd związany z brakiem odpowiedniej biblioteki w środowisku Ruby:

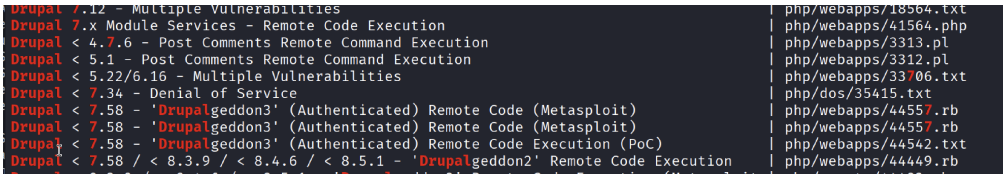

Figure 7: Rysunek 7: Błąd uruchomienia exploita - brak biblioteki highline w Ruby

Analiza błędu wykazała, że zainstalowana wersja Ruby jest zbyt nowa i brakuje kompatybilnej wersji biblioteki `highline`.

Figure 8: Rysunek 8: Analiza błędu - niezgodność wersji biblioteki highline z nową wersją Ruby

Rozwiązanie: Ręczna instalacja starszej wersji biblioteki `highline`: `sudo gem install highline -v 2.1.0`

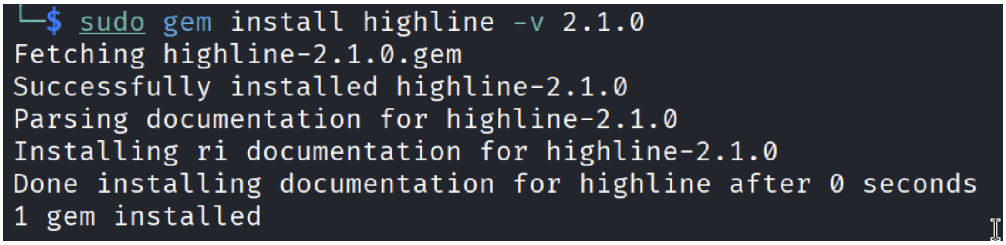

Figure 9: Rysunek 9: Naprawa zależności - instalacja kompatybilnej wersji biblioteki highline

Uzyskanie powłoki

Po naprawieniu zależności, exploit został uruchomiony ponownie, celując w maszynę ofiary. Tym razem skrypt zadziałał poprawnie, otwierając sesję powłoki.

---

4. Post-eksploitacja

Będąc w systemie, przeprowadzono rekonesans katalogów. W głównym katalogu serwera WWW (`/var/www`) odnaleziono plik `flag1.txt`.

Treść flagi nr 1:

No i koniec

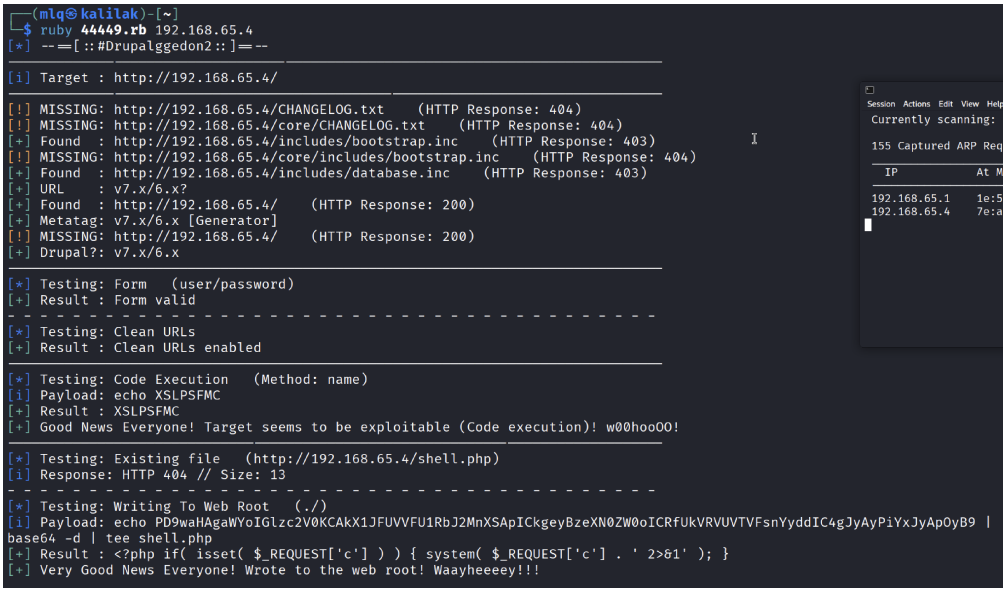

Figure 10: Rysunek 10: Uzyskanie dostępu CLI - interaktywna powłoka systemowa na maszynie DC-1

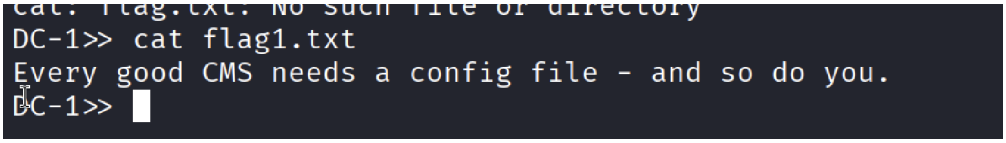

Figure 11: Rysunek 11: Zawartość flagi nr 1 znalezionej w katalogu /var/www