Vulnerability Assessment Report Template

Ime i prezime: Miloš Stojanović R2 29/2024
Tim: Tim 10
Datum: 26.10.2024.
Scan Tool: Nessus (10.8.3)
Test okruženje: Metasploitable3

---

1. Enumeracija CVE-a

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

- **Opis:**

  node-tar je paket za Node.js. node-tar pre verzije **6.2.1** nema ograničenja na broj podfoldera kreiranih za vreme kreiranja datoteke. Napadač koji generiše veliki broj podfoldera može da potroši RAM memoriju na sistemu koji pokreće node-tar, pa čak i da sruši Node.js klijent u roku od nekoliko sekundi od pokretanja koristeći putanju sa previše podfoldera.

---

2. CVSS skor

- **CVSS skor (numerička vrednost): 6.5**
- **Vektor:** CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H

  Attack vector: Network – Ranjivost dolazi od paketa sa koji se preuzima sa interneta. Napadač nije primoran da bude fizički blizu sistemu, već ga može napasti preko mreže, a najverovatnije preko interneta.
  Attack complexity: Low – Napadaču nije potrebno mnogo znanja za izvođenje napada. Takođe, sam napad nije previše kompleksan s obzirom na to da je samo potrebno postaviti veliki broj podfoldera pri kreiranju datoteke.
  Privileges Required: None – Napadaču nije potrebna nikakva veza sa sistemom za vršenje napada. Sam napad ne zahteva nikakve privilegije.
  User interaction: Required – Da bi se napad izvršio, potrebno je da korisnik pokrene raspakivanje datoteke.
  Scope: Unchanged – Opseg ranjivosti se ne menja.

Confidentiality: None – Napadom se ne narušava poverljivost podataka pošto ne dolazi do pristupa nedozvoljenim podacima.
Integrity: None – Napadač ne menja podatke niti dolazi u kontakt sa bilo kakvim podacima na sistemu.
Availability: High – Napadač može da zaguši sistem punjenjem RAM memorije i time napravi da sistem postane nedostupan.

- **Opravdanje:**

  Ovim napadom je moguće srušiti node client, a samim tim je moguće i iscrpiti resurse procesora i RAM memorije. U tom slučaju može doći do zamrzavanja sistema kao i prestajanja rada bitnih servisa. Olakšavajuća okolnost je to što je potrebna akcija korisnika da bi se izvršio napad pa postoji mogućnost da će se maliciozni kod primetiti pre pokretanja aplikacije. Nezgodno je to što, ukoliko je maliciozni kod dobro sakriven u aplikaciji ili korisnik nije dovoljno stručan da prepozna maliciozan kod, lako može doći do izvršenja napada. S obzirom na to da nije teško napisati kod koji koristi ovu ranjivost, eksploatabilnost je velika. Ova ranjivost ima potencijal da proizvede situaciju koju napadač može da iskoristi da napravi još veći problem sistemu. Ukoliko se, recimo, korišćenjem ove ranjivosti sruši neki bezbednosni sistem, to ostavlja prostor napadaču da napadne sistem koji je do tada bio zaštićen.

---

3. Dostupnost eksploita

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

  U ovom github issue je dat opis kako se moze iskoristiti ranjivost.

  https://github.com/isaacs/node-tar/security/advisories/GHSA-f5x3-32g6-xq36

- **Opis eksploita:**
  Potrebno je pri kreiranju datoteke specificirati putanju fajla u veliku dubinu. Na primer: root/folder1/folder2/.../folder10000000000/file.txt
  Ovakvim kreiranjem datoteke će se iscrpeti RAM memorija i može doći do zamrzavanja sistema.
- **Kod eksploita (ukoliko postoji):**

  Kod nije preuzet sa github linka. Napisan je prema opisu exploita. Bitno je da se pri kreiranju datoteke prođe kroz for petlju i napravi da putanja do fajla prolazi kroz veliki

broj podfoldera (odnosno u veliku dubinu).

```
function generateDeepNestedFolderStructure() {
  const dirPath = path.join(__dirname, 'deep-nested');
  let nestedDir = dirPath;

  for (let i = 0; i < 1000000000000; i++) {
    nestedDir = path.join(nestedDir, `folder${i}`);
    fs.mkdirSync(nestedDir, { recursive: true });
  }

  tar.c(
    {
      gzip: true,
      file: path.join(__dirname, 'deep-nested.tar.gz'),
      cwd: dirPath,
    },
    ['.']
  ).then(() => {
    console.log('Tar file created successfully!');
  });
}
```

---

4. Analiza uzroka (root cause)

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

  Ranjivost postoji od prve verzije node-tar paketa sve do verzije 6.2.1 kada je uveden fix. Problem je to što nije postojala provera za dubinu podfoldera pri kreiranju datoteke pa je time praktično bilo dozvoljeno kreiranje datoteke sa beskonačnom dubinom. Problem je rešen tako što je uvedena granica za dubinu. Podrazumevano je podešena na 1024, ali se može menjati i ukinuti (postavljanjem vrednosti na “Infinity”).

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

```
                                                  51 + const DEFAULT_MAX_DEPTH = 1024
51
52      // Unlinks on Windows are not atomic.     53      // Unlinks on Windows are not atomic.
53      //                                        54      //
        @@ -181,6 +182,12 @@ class Unpack extends Parser {
181         this.processGid = (this.preserveOwner ||  182         this.processGid = (this.preserveOwner ||
        this.setOwner) && process.getgid ?                this.setOwner) && process.getgid ?
182           process.getgid() : null                 183           process.getgid() : null
183                                                   184
                                                    185 +     // prevent excessively deep nesting of subfolders
                                                    186 +     // set to `Infinity` to remove this restriction
                                                    187 +     this.maxDepth = typeof opt.maxDepth === 'number'
                                                    188 +       ? opt.maxDepth
                                                    189 +       : DEFAULT_MAX_DEPTH
                                                    190 +
```

5. Preporuke za mitigaciju

- **Da li je dostupan Vendor Fix ili patch (Da/Ne): Da**
- **Mitigation Strategy:**

  Potrebno je odraditi instalaciju ili update paketa na verziju iznad 6.2.0. To se može uraditi komandom: **npm install tar@6.2.1**

- **Alternativni fix (ukoliko ne postoji vendorski):**

  Problem bi se rešio tako što se izmeni paket i uvede granica za dubinu i provera te granice prilikom kreiranja datoteke.

Vulnerability Assessment Report Template

Ime i prezime: Miloš Stojanović R2 29/2024
Tim: Tim 10
Datum: 30.10.2024.
Scan Tool: Nessus (10.8.3)
Test okruženje: Metasploitable3

---

1. Enumeracija CVE-a

- **CVE ID: CVE-2014-6271**

- **Opis:**

  Bash podržava export-ovanje shell funkcija kao environment variables. Ranjivost je to što bash nastavlja da parsira i izvršava shell komande nakon definisanja funkcije. Na primer:

  EXAMPLE=() { … }; some-command

  Ako se ovakva funkcija export-uje, kada se bude uvela kao environment variable u neki drugi bash proces, komanda posle definicije, tj. some-command će se izvršiti. Ovim se dozvoljava remote arbitrary code execution. Ranjivost je prisutna u GNU Bash do verzije 4.3.

---

2. CVSS skor

- **CVSS skor (numerička vrednost): 9.8**
- **Vektor:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

  Attack vector: Network – Napadač nije primoran da bude fizički blizu sistemu, već ga može napasti preko mreže.
  Attack complexity: Low – Napadaču nije potrebno mnogo znanja za izvođenje napada. Takođe, sam napad nije previše kompleksan, jer je samo potrebno posle definicije funkcije proslediti niz komandi koje treba izvršiti.

Privileges Required: None – Napadaču nije potrebna nikakva veza sa sistemom za vršenje napada. Sam napad ne zahteva nikakve privilegije (možda komande budu zahtevale). Korisnik ne mora biti autentifikovan niti autorizovan da bi izvršio napad.
User interaction: None – Da bi se napad izvršio, korisnik ne mora izvršiti nikakvu akciju.
Scope: Unchanged – Opseg ranjivosti se ne menja, odnosno, ograničen je na sistem na kome se vrši napad.
Confidentiality: High – Napadom je moguće ukrasti podatke pa se time narušava poverljivost podataka.
Integrity: High – Napadač može izvrsiti komandu kojom je moguće izmeniti ili obrisati podatke pa bi se time narušio i integritet podataka.
Availability: High – Zbog toga što je komanda koja se injektuje proizvoljna, može doći i do narušavanja dostupnosti sistema.

- **Opravdanje:**

  Ovim napadom je moguće izvršiti proizvoljni kod na nekom sistemu. Podešavanje environment varijabli nije teško uraditi, a i najčešće ne postoji mehanizam kojim bi se to sprečilo pa je zbog toga ovaj napad lako izvesti. Sam opus stvari koje može uraditi napadač injektovanjem komande je veliki, od krađe podataka do narušavanja dostupnosti sistema, pa zato ovaj napad ima visoku ocenu. Jos jedna stavka koja pridodaje značaju napada je to što nije potrebna nikakva akcija korisnika da bi se on izvršio, što znači da se napad može izvršiti u skoro svakom trenutku i da bi možda bio težak za uočavanje.

---

3. Dostupnost eksploita

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

  https://www.exploit-db.com/exploits/34766

- **Opis eksploita:**
  Exploit iskorišćava to što CGI (common gateway interface, koji služi kao veza interfejs između web servera i eksternih programa (CGI skripti)) smešta HTTP zaglavlja u environment variables kada prosleđuje vrednosti u Bash. Kada se u nekom zaglavlju nalazi maliciozna komanda, ona dolazi do Bash-a kao environment varijabla i kada se ta varijabla bude koristila, ta komanda će se izvršiti.

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

```php
$context = stream_context_create(
    array(
        'http' => array(
            'method'  => 'GET',
            'header'  => 'User-Agent: () { :;}; /bin/bash -c "'.$cmd.'"'
        )
    )
);
```

---

4. Analiza uzroka (root cause)

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

  Ranjivost postoji od prve verzije Bash-a sve do verzije 4.3 kada je uveden fix. Greška je bila u tome sto je nakon definicije funkcije nastavljano izvršavanje komandi.

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

  https://ftp.gnu.org/pub/gnu/bash/bash-4.3-patches/bash43-025

---

5. Preporuke za mitigaciju

- **Da li je dostupan Vendor Fix ili patch (Da/Ne): Da**
- **Mitigation Strategy:**

  Treba instalirati Bash počevši sa verzijom 4.3. Proveru da li je sistem ranjiv na ovu ranjivost i update je moguće uraditi komandom:

  env x='() { :;}; echo vulnerable' bash -c 'echo hello' | grep vulnerable && (sudo apt-get update && sudo apt-get install bash) || echo 'not vulnerable'

  https://gist.github.com/charlespeach/b83939eb4495c6118767

- **Alternativni fix (ukoliko ne postoji vendorski):**

Vulnerability Assessment Report Template

Ime i prezime: Miloš Stojanović R2 29/2024
Tim: Tim 10
Datum: 30.10.2024.
Scan Tool: Nessus (10.8.3)
Test okruženje: Metasploitable3

---

1. Enumeracija CVE-a

- **CVE ID: CVE-2023-48795**

- **Opis:**

  Napadač može da ukloni proizvoljnu količinu paketa koji se šalju tokom handshake-a tako što prosledi određeni sequence number. Time može promeniti algoritam za autentifikaciju korisnika i podesiti da se koristi neki slabiji. OpenSSH je ranjiv zaključno sa verzijom 9.5. Da bi se napad izvršio, potreban je man-in-the-middle napadač kao i kombinacija ChaCha20-Poly1305 algoritma sa Encrypt-then-MAC.

---

2. CVSS skor

- **CVSS skor (numerička vrednost): 5.9**
- **Vektor:** CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N

  Attack vector: Network – Napadač nije primoran da bude fizički blizu sistemu, već ga može napasti preko mreže.
  Attack complexity: High – Potrebno je da napadač zna kako funkcioniše SSH protokol kao i OpenSSH. Pored toga je potrebno da zna da manipuliše paketima i da zna koji sequnce number treba da postavi.
  Privileges Required: None – Napadaču nije potrebna nikakva veza sa sistemom za vršenje napada. Sam napad ne zahteva nikakve privilegije.
  User interaction: None – Da bi se napad izvršio, korisnik ne mora izvršiti nikakvu akciju.
  Scope: Unchanged – Opseg ranjivosti se ne menja, odnosno, ograničen je na sistem na kome se vrši napad.
  Confidentiality: None – Napadom se ne mogu ukrasti podaci.

Integrity: High – Integritet podataka se narušava menjanjem algoritma za autentifikaciju korisnika.
Availability: None – Napad ne može dovesti do nedostupnosti sistema.

- **Opravdanje:**

  Da bi napad uspeo, napadač mora biti u man-in-the-middle poziciji pošto je potrebno da menja podatke između dva peer-a. To doprinosi malo nižoj oceni. Na ocenu utiče i to što je napad veće kompleksnosti kao i posebna kombinacija algoritama. Ono što doprinosi značaju napada je to što je ranjivost na nivou protokola pa postoji dosta implementacija koje su ranjive.

---

3. Dostupnost eksploita

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

  https://terrapin-attack.com/TerrapinAttack.pdf

- **Opis eksploita:**

  Prilikom handshake-a kada server vraća odgovor klijentu, man-in-the-middle menja sadržaj odgovora tako što iz paketa ukloni deo koji nosi podatke o algoritmima. Time klijent ne dobija dobre podatke i narušava se integritet podataka.

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

  https://github.com/RUB-NDS/Terrapin-Artifacts/

  https://github.com/RUB-NDS/Terrapin-Artifacts/blob/main/pocs/ext-downgrade/ext_downgrade_chacha20_poly1305.py

```
# Insert ignore message (to client)
client_socket.send(rogue_msg_ignore)
# Wait half a second here to avoid missing EXT_INFO
# Can be solved by counting bytes as well
sleep(0.5)
# KEX_REPLY / NEW_KEYS / EXT_INFO
server_response = server_socket.recv(35000)
# Strip EXT_INFO before forwarding server_response to client
# Length fields of KEX_REPLY and NEW_KEYS are still unencrypted
server_kex_reply_length = LENGTH_FIELD_LENGTH + int.from_bytes(server_response[:LENGTH_FIELD_LENGTH], byteorder='big')
server_newkeys_start = server_kex_reply_length
server_newkeys_length = LENGTH_FIELD_LENGTH + int.from_bytes(server_response[server_newkeys_start:server_newkeys_start + LENGTH_FIELD_LENGTH], byteorder='big')
server_extinfo_start = server_newkeys_start + server_newkeys_length
client_socket.send(server_response[:server_extinfo_start])
```

4. Analiza uzroka (root cause)

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

  Ranjivost postoji od prve verzije OpenSSH zaključno sa verzijom 9.5. Ranjivost je rešena u e7010dc komitu. Rešeno je tako što, ukoliko deo paketa ne odgovara konkretnom sequence number, handshake se prekida. Takođe, prvi paket mora da bude KEX_INIT inače se handshake prekida.

  https://github.com/PowerShell/openssh-portable/commit/e7010dc405279e32d26daf6b94134bf04761a4db

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

```
492 +         /* If in strict mode, any unexpected message
      is an error */
493 +         if ((ssh->kex->flags & KEX_INITIAL) && ssh-
      >kex->kex_strict) {
494 +                 ssh_packet_disconnect(ssh, "strict
      KEX violation: "
495 +                     "unexpected packet type %u (seqnr
      %u)", type, seq);
496 +         }
497 +         error_f("type %u seq %u", type, seq);
```

5. Preporuke za mitigaciju

- **Da li je dostupan Vendor Fix ili patch (Da/Ne): Da**
- **Mitigation Strategy:**

  Treba instalirati OpenSSH iznad verzije 9.5.

  https://github.com/PowerShell/Win32-OpenSSH/releases/tag/v9.5.0.0p1-Beta

- **Alternativni fix (ukoliko ne postoji vendorski):**

  Treba zabraniti korišćenje slabijih algoritama, npr. chacha20-poly1305@openssh.com.