Vulnerability Assessment Report Template

Ime i prezime: Marija Ilić
Tim: 11
Datum: 27.10.2024.
Scan Tool: Nessus (10.8.3)
Test okruženje: Metasploitable3

---

1. Enumeracija CVE-a

- **CVE ID:CVE-2016-6816**
- **Opis:**
  Apache Tomcat verzija je starija od 8.0.39 i podlozna je injektovanju podataka u HTTP odgovor. Kod koji analizira HTTP liniju zahteva u Apache Tomcat verzijama(9.0.0.M1 do 9.0.0.M11, 8.5.0 do 8.5.6, 8.0.0.RC1 do 8.0.38, 7.0.0 do 7.0.72 i 6.0.0 do 6.0.47) dozvoljava nevazece znakove. Ovo se moze iskoristiti zajedno sa proksijem koji dozvoljava nevazece znakove i da se ubace ti podaci u HTTP odgovor. Manipulisuci HTTP odgovorom napadacam bi mogao izvrsi napad i da dobije osetljive informacije korisnika iz zahteva.

---

2. CVSS skor

- **CVSS skor (numerička vrednost): 7.1**
- **Vektor: `CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:L`**
  AV:N- Attack Vector: Network. Napadac moze da iskoristi ovu ranjivost putem interneta ili lokalne mreze.
  AC:L - Attack Complexity: Low. Napad je lak za izvrsenje.
  PR:N - Privileges Required: None. Napadac ne mora da ima nikakva ovlascenja za iskoriscavanje ranjivosti.
  UI:R- User Interaction: Required. Napad zahteva interakciju korisnika.
  S:C- Scope: Changed. Napad moze uticati na druge komponente i sisteme ne samo na napadnuti sistem.
  C:L - Confidentiality Impact: Low. Napadac moze dobiti minimalan pristup poverljivim informacijama.

I:L - Integrity Impact: Low. Napadač može izmeniti podatke, ali ne u značajnoj meri.
A:L - Availability Impact - Low. Napadac moze onemoguciti neke usluge ali to nece uticati na dostupnost sistema.

- **Opravdanje:**
  Vektor nam govori da ranjivost može biti lako iskorišćena putem interneta ili lokalne mreze, ali zahteva aktivnost korisnika, dok su potencijalni uticaji na poverljivost, integritet i dostupnost minimalni.

---

3. Dostupnost eksploita

- **Postoji javno dostupan eksploit (Da/Ne):Da**
  ExploitDb link.
- **Opis eksploita:**
  Napadac moze da iskoristi to sto verzije ApacheTomcat-a 9.0.0.M1 do 9.0.0.M11, 8.5.0 do 8.5.6, 8.0.0.RC1 do 8.0.38, 7.0.0 do 7.0.72 i 6.0.0 do 6.0.47 dozvoljavaju nevazece znakove i zajedno sa proksijem koji to isto dozvoljava da ubace podatke u HTTP odgovor. Tako moze da ugrozi bezbednost veb aplikacije i dovodi od narusavanje privatnosti i integriteta podataka.
  Posledice:
  - Kesirani odgovori proksi servera mogu biti izmenjeni i to dovodi do prikazivanja laznog sadrzaja korisnika.
  - XSS napad- izvrsavanje skripti u korisinickom pregledacu moze omoguciti kradju podataka.
  - Napadac moze dobiti pristup poverljivim indormacijama iz zahteva korisnika.

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

```
GET /?{{%25}}cake\=1 HTTP/1.1
Host: justpentest.com
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64;
Trident/5.0)
Connection: close
Cookie:
NSC_MSN-IBNQ-VX-mcwtfswfs=ffffffff091c1daaaa525d5f4f58455e445a4a488888

 OR

GET
/?a'a%5c'b%22c%3e%3f%3e%25%7d%7d%25%25%3ec%3c[[%3f$%7b%7b%25%7d%7dcake%5c=1
HTTP/1.1
```

U prvom primeru mozemo da vidimo `GET /?{{%25}}cake\=1` i to je pokusaj koriscenja nevazecih znakova, {{%25}} se interpreter kao {{%}} sto izaziva da server ne validira parametre, \= moze dovesti da server pogresno parsira parametre. Host i User-Agent identifikuju server na koji se zahtev šalje i aplikaciju koja ga šalje (lažni User-Agent identifikuje pretraživač kao Internet Explorer 9).Cookie simulira postojecu sesija i pomaze napadacu da zaobidje autentifikaciju.

U drugom primeru vidimo da zahtev koristi znakove i kreira kompleksan i necitljiv niz. Cilj je da se zbuni servis ili proksi koji mogu interpretirati sekvence na razlicite nacine. Kada servis i proksi obrade znakove napadac moze da postigne da se podaci nalaye u odgovoru servera i da manipulise HTTP kesiranjem.

---

4. Analiza uzroka (root cause)

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

  Ranjivost je uvedena u Apache Tomcat verziji 8.0.0.RC1. Nije specificno naveden datum uvodjenja, ali je ranjivost prijavljena 11.oktobra 2016.godine. U verzijama 1713990 i 1743647 su se stvorile greske.

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

```
--- tomcat/tc8.0.x/trunk/java/org/apache/coyote/http11/AbstractInputBuffer.java 2015/11/12 09:33:08     1713990
+++ tomcat/tc8.0.x/trunk/java/org/apache/coyote/http11/AbstractInputBuffer.java 2016/11/02 12:18:08     1767653
@@ -30,62 +30,10 @@ import org.apache.tomcat.util.res.String

 public abstract class AbstractInputBuffer<S> implements InputBuffer{

-    protected static final boolean[] HTTP_TOKEN_CHAR = new boolean[128];
-
     /**
      * The string manager for this package.
      */
-    protected static final StringManager sm =
-        StringManager.getManager(Constants.Package);
-
-
-    static {
-        for (int i = 0; i < 128; i++) {
-            if (i < 32) {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == 127) {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '(') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == ')') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '<') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '>') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '@') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == ',') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == ';') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == ':') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '\\') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '\"') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '/') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '[') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == ']') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '?') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '=') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '{') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == '}') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else if (i == ' ') {
-                HTTP_TOKEN_CHAR[i] = false;
-            } else {
-                HTTP_TOKEN_CHAR[i] = true;
-            }
-        }
-    }
+    protected static final StringManager sm = StringManager.getManager(Constants.Package);


     /**
```

Ovo je prvi primer koda u klasi AbstractInputBuffer.Java, iz revizije 1713990 koristila je statcki niz boolean-a za validaciju HTTP tokena i to je neefikasno i ograničeno zbog potrebe za ručnim proverama svake karakteristike.

```
--- tomcat/tc8.0.x/trunk/java/org/apache/coyote/http11/AbstractNioInputBuffer.java      2016/05/13 10:29:03     1743647
+++ tomcat/tc8.0.x/trunk/java/org/apache/coyote/http11/AbstractNioInputBuffer.java      2016/11/02 12:18:08     1767653
@@ -21,6 +21,7 @@ import java.nio.charset.StandardCharsets

 import org.apache.coyote.Request;
 import org.apache.tomcat.util.buf.MessageBytes;
+import org.apache.tomcat.util.http.parser.HttpParser;

 public abstract class AbstractNioInputBuffer<S> extends AbstractInputBuffer<S> {

@@ -228,7 +229,7 @@ public abstract class AbstractNioInputBu
                 if (buf[pos] == Constants.SP || buf[pos] == Constants.HT) {
                     space = true;
                     request.method().setBytes(buf, parsingRequestLineStart, pos - parsingRequestLineStart);
-                } else if (!HTTP_TOKEN_CHAR[buf[pos]]) {
+                } else if (!HttpParser.isToken(buf[pos])) {
                     throw new IllegalArgumentException(sm.getString("iib.invalidmethod"));
                 }
                 pos++;
@@ -276,9 +277,10 @@ public abstract class AbstractNioInputBu
                     parsingRequestLineEol = true;
                     space = true;
                     end = pos;
-                } else if ((buf[pos] == Constants.QUESTION)
-                           && (parsingRequestLineQPos == -1)) {
+                } else if ((buf[pos] == Constants.QUESTION) && (parsingRequestLineQPos == -1)) {
                     parsingRequestLineQPos = pos;
+                } else if (HttpParser.isNotRequestTarget(buf[pos])) {
+                    throw new IllegalArgumentException(sm.getString("iib.invalidRequestTarget"));
                 }
                 pos++;
             }
@@ -315,7 +317,7 @@ public abstract class AbstractNioInputBu
         if (parsingRequestLinePhase == 6) {
             //
             // Reading the protocol
-            // Protocol is always US-ASCII
+            // Protocol is always "HTTP/" DIGIT "." DIGIT
             //
             while (!parsingRequestLineEol) {
                 // Read new bytes if needed
@@ -330,6 +332,8 @@ public abstract class AbstractNioInputBu
                     if (end == 0)
                         end = pos;
                     parsingRequestLineEol = true;
+                } else if (!HttpParser.isHttpProtocol(buf[pos])) {
+                    throw new IllegalArgumentException(sm.getString("iib.invalidHttpProtocol"));
                 }
                 pos++;
             }
@@ -470,7 +474,7 @@ public abstract class AbstractNioInputBu
                 headerData.realPos = pos;
                 headerData.lastSignificantChar = pos;
                 break;
-            } else if (chr < 0 || !HTTP_TOKEN_CHAR[chr]) {
+            } else if (!HttpParser.isToken(chr)) {
                 // If a non-token header is detected, skip the line and
                 // ignore the header
                 headerData.lastSignificantChar = pos;
```

Ovo je iz klase AbstractNioInputBuffer.java iz revizije 1743647, kod je zahtevao manualne provere za svaki karakter, što je otežavalo održavanje. U ovoj verziji nije bilo bolje rukovanje greškama, a kod je bio manje čitljiv.

---

5. Preporuke za mitigaciju

- **Da li je dostupan Vendor Fix ili patch (Da/Ne):Da**
- **Mitigation Strategy:**
  Preporucuje se da se azurira verzija na 8.0.39 ili novije, koja ukljucuje ispravke.

- **Alternativni fix (ukoliko ne postoji vendorski):** Ako ne postoji dostupna verzija onda se moze ograniciti pristup samo na funkcionalnosti koje su neophodne za rad aplikacije. Moze se aplikacija postaviti na poseban server ili virtuelnu masinu. Redovnim azuriranjem i uvodjenjem logova dobija se evidencija o pristupu i aktivnostima korisnika.