Vulnerability Assessment Report Template

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

CVE-2016-6816

---

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.

  Servis: www (Web server)

  Port: 8282

  Protokol: tcp

---

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. Napomena: Nije verifikovan.
- **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 sesiju 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 nalaze u odgovoru servera i da manipulise HTTP kesiranjem.

---

4. Analiza uzroka (root cause)

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

  Ranjivost je uvedena u Apache Tomcat verzijama izmedju 8.0.0.RC1 i 8.0.39. Nije specificno naveden datum uvodjenja, ali je ranjivost prijavljena 11.oktobra 2016.godine. U verzijama 1713990 i 1743647 su se stvorile greske. Razreseno je u verziji 1767653.

- **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.

Update na Linuxu:

VERSION=8.0.39

Wget
https://archive.apache.org/dist/tomcat/tomcat-8/v$VERSION/bin/apache-tomcat-$VERSION.tar.gz

sudo tar -xzvf apache-tomcat-$VERSION.tar.gz -C /opt/

sudo systemctl stop tomcat

sudo mv /opt/tomcat /opt/tomcat-backup

sudo ln -sfn /opt/apache-tomcat-$VERSION /opt/tomcat

sudo systemctl start tomcat

Update na Windows-u, koristeci Chocolatey:

choco upgrade tomcat --version=8.0.39

- **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.

CVE-2022-36760

---

1. Enumeracija CVE-a

- **CVE ID:CVE-2022-36760**
- **Opis:**
  Ova ranjivost se javlja u Apache Http Serveru u verzijama 2.4.54 i starijim u modulu mod_proxy_ajp, koja omogucava napad HTTP Request Smuggling. Omogucava napadacu da posalje zahteve do AJP servera na koji Apache prosledjuje zahteve.

  Servis: www (Web server)

  Port:8585

  Protokol: TCP

---

2. CVSS skor

- **CVSS skor (numerička vrednost): 9.0**
- **Vektor: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H**
  AV:N- Attack Vector: Network. Napadac moze da iskoristi ovu ranjivost putem interneta ili lokalne mreze.
  AC:H - Attack Complexity: High. Napad je slozen.
  PR:N - Privileges Required: None. Napadac ne mora da ima nikakva ovlascenja za iskoriscavanje ranjivosti.
  UI:N- User Interaction: None. Napad ne zahteva interakciju korisnika.
  S:C- Scope: Changed. Napad moze uticati na druge komponente i sisteme ne samo na napadnuti sistem.
  C:H - Confidentiality Impact:High. Napadac moze dobiti pristup poverljivim informacijama i ugroziti sigurnost.
  I:H - Integrity Impact: High. Napadač može izmeniti i ostetiti podatke.
  A:H - Availability Impact: High. Napadac utice na dostupnost i potencijalno onemogucava servis ili resurs

- **Opravdanje:**
  Skor od 9.0 svrstava ranjivost u kriticne. To znaci da ranjivost moze da bude iskoriscena od strane napadaca da izvede napade poput kradje podataka, unistavanje sistema .

3. Dostupnost eksploita

- **Postoji javno dostupan eksploit (Da/Ne):Ne, medjutim nasla sam na sajtu** kako bi napad mogao da izgleda
- **Opis eksploita:**
  HTTP Request Smuggling napadi mogu imati ozbiljne posledice za web aplikacije, proxy servere i njihove korisnike. Kada napadač iskoristi ovu ranjivost, mogu nastati sledeće posledice:
    1. **Cache Poisoning**: Napadač može poslati zlćudni zahtev koji se kešira na proxy serveru. Kada drugi korisnici zatraže tu stranicu, umesto očekivanog sadržaja dobijaju zlonamerne podatke. Ovo može biti izuzetno štetno, jer može uključivati phishing stranice ili malware.
    2. **Bypassing Security Mechanisms**: Kroz loše oblikovane zahteve, napadač može zaobići bezbednosne filtere firewall-a, omogućavajući im da pošalju zlonamerne zahteve ili da izvrše neautorizovane akcije na serveru.
    3. **Malicious Payload Delivery**: Napadač može ubaciti los kod u odgovore servera koji se šalju korisnicima, potencijalno inficirajući njihove uređaje ili kradući njihove podatke.
    4. **Manipulacija sa Resursima**: Korišćenjem ovog napada, napadači mogu da manipulisu resursima, menjajući sadržaj stranica ili preusmeravajući korisnike na nebezbedne stranice.

- **Kod eksploita (ukoliko postoji):**
  https://cwe.mitre.org/data/definitions/444.html ovde mozemo da vidimo primer koda pod “Demonstrative Examples”, kako bi mogao napad da izgleda.

  Prvi primer prvo pokazuje lose oblikovan Http yahtev koji se salje na sajt koji ukljucuje proxy server i veb server, sa ciljem da se zasiti kes i poveze jedna veb stranica sa drugom zlonamernom stranicom.

```
POST http://www.website.com/foobar.html HTTP/1.1
Host: www.website.com
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
Content-Length: 54

GET /poison.html HTTP/1.1
Host: www.website.com
Bla: GET http://www.website.com/page_to_poison.html HTTP/1.1
Host: www.website.com
Connection: Keep-Alive
```

Kada se ovaj zahtev pošalje proxy serveru, proxy server obrađuje prva četiri reda POST zahteva i susreće dva "Content-Length" zaglavlja. Proxy server ignoriše prvo zaglavlje, tako da pretpostavlja da zahtev ima telo dužine 54 bajta. Stoga, tretira podatke u sledeća tri reda kao telo prvog zahteva. Ovo je rezultat:

```
GET /poison.html HTTP/1.1
Host: www.website.com
Bla:
```

Proxy zatim obrađuje preostale bajtove, koje tretira kao drugi zahtev klijenta.

```
GET http://www.website.com/page_to_poison.html HTTP/1.1
Host: www.website.com
Connection: Keep-Alive
```

Originalni zahtev se prosleđuje od strane proxy servera veb serveru. Za razliku od proxy servera, veb server koristi prvo "Content-Length" zaglavlje i smatra da prvi POST zahtev nema telo.

```
POST http://www.website.com/foobar.html HTTP/1.1
Host: www.website.com
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
Content-Length: 54 (ignored by server)
```

Pošto je veb server pretpostavio da je originalni POST zahtev dužine 0, obrađuje drugi zahtev koji dolazi, tj. za GET /poison.html:

```
GET /poison.html HTTP/1.1
Host: www.website.com
Bla: GET http://www.website.com/page_to_poison.html HTTP/1.1
Host: www.website.com
Connection: Keep-Alive
```

Zaglavlje "Bla:" se tretira kao regularno zaglavlje, tako da se ne obrađuje kao poseban GET zahtev.

Zahtevi koje veb server vidi su "POST /foobar.html" i "GET /poison.html", tako da šalje dva odgovora sa sadržajem stranice "foobar.html" i "poison.html". Proxy upoređuje ove odgovore sa dva zahteva za koje misli da ih je poslao klijent - "POST /foobar.html" i "GET /page_to_poison.html". Ako je odgovor kešabilan, proxy kešira sadržaj "poison.html" pod URL-om "page_to_poison.html", i keš je otrovan! Bilo koji klijent koji zatraži "page_to_poison.html" od proxy-a biće mu poslat "poison.html" sadržaj.

---

4. Analiza uzroka (root cause)

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

  Ranjivost je prijavljena bezbednosnom timu Apache-a 12. jula 2022. godine, a rešenje je objavljeno u verziji 2.4.55 na dan 17. januara 2023**.** Ova verzija je uključivala ispravke koje su obezbedile pravilniju obradu HTTP zahteva i dodatne provere za višestruka zaglavlja.

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

  Ne postoji primer koda. Ključni uzrok je neusklađena interpretacija višestrukih "Content-Length" zaglavlja. Kada proxy server primi zahtev s više zaglavlja, može ignorisati jedno od njih, što dovodi do pogrešnog tumačenja veličine tela zahteva. Ovo omogućava napadačima da podmetnu dodatne zahteve unutar jednog HTTP zahteva, što može dovesti do "smuggling" napada, kao što su keširanje zlonamernog sadržaja ili obilaženje sigurnosnih provere.

---

5. Preporuke za mitigaciju

- **Da li je dostupan Vendor Fix ili patch (Da/Ne):Da**
- **Mitigation Strategy:**
  Preporučuje se svim korisnicima Apache HTTP Server-a koji koriste verzije 2.4.54 i starije da nadograde na verziju 2.4.55 ili noviju kako bi eliminisali ovu ranjivost.
  Komande:

Linux:

1. sudo apt update
2. sudo apt upgrade apache2
3. apache2 -v

Windows:

1. Preuzmite najnoviju verziju Apache HTTP Server-a sa Apache Lounge.
2. Instalirajte novu verziju, a zatim ponovo pokrenite server.
3. Proverite verziju pomoću komandne linije:httpd -v

- **Alternativni fix (ukoliko ne postoji vendorski):** Ako ne postoji dostupna verzija onda se moze onemoguciti AJP ako nike potreban, da se koristi WAF(Web Application Firewall), ona moze da da blokira sumnjive zahteve, restrikcija pristupa i monitoring i azuriranje,

CVE-2010-3972

---

1. Enumeracija CVE-a

- **CVE ID:CVE-2010-3972**
- **Opis:**
  IIS FTP servis koji radi na udaljenom hostu sadrži ranjivost povezanih sa heap-based buffer overflow. U funkciji TELNET_STREAM_CONTEXT::OnSendData dolazi do problema u obradi korisničkog unosa, što može rezultirati prelivanjem bafera. Ova ranjivost omogućava neautentifikovanim, udaljenim napadačima da izvrše proizvoljan kod na ranjivom sistemu.

  Servis: FTP

  Port:21

  Protokol: TCP

---

2. CVSS skor

- **CVSS skor (numerička vrednost): 9.8**
- **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. 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:N- User Interaction: None. Napad ne zahteva interakciju korisnika.
  S:U- Scope: Unchanged. Napad ne utice na druge komponente i sisteme, vec samo na napadnuti sistem.
  C:H - Confidentiality Impact:High. Napadac moze dobiti pristup poverljivim informacijama i ugroziti sigurnost.
  I:H - Integrity Impact: High. Napadač može izmeniti i ostetiti podatke.
  A:H - Availability Impact: High. Napadac utice na dostupnost i potencijalno onemogucava servis ili resurs

- **Opravdanje:**
  Ovaj CVSS skor od 9.8 označava visoku kritičnost ranjivosti i odražava ozbiljnost potencijalnog napada. Ranjivost se smatra kritičnom, sa visokim rizikom od

iskorišćavanja koji može dovesti do ozbiljnih posledica po bezbednost i funkcionisanje sistema.

---

3. Dostupnost eksploita

- **Postoji javno dostupan eksploit (Da/Ne):Da, link.**
- **Opis eksploita:**
  Ovaj exploit se fokusira na ranjivost u FTP servisu Microsoft IIS 7.0, koja omogućava neovlašćeni pristup sistemu putem izvođenja komandi na daljinu. Ranljivost je povezana sa nepravilnom obradom FTP komandi od strane servera, što napadaču omogućava da izvrši arbitrarne komande na serveru. Napadač može preuzeti kontrolu nad serverom, što može rezultirati gubitkom podataka ili oštećenjem sistema. Eksploit omogućava napadaču da upravlja sistemom, instalira malver ili koristi server za dalje napade.
- **Kod eksploita (ukoliko postoji):**

Na linku se moze videti promenljiva buf koja je popunjena specificnim nizom bajtova, to je pokusaj preopterecenja memorije. I onda se to koristi u dole prikazanom kodu.

Funkcija usage: Ova funkcija prikazuje kako koristiti skriptu. Format za pokretanje je: ./msIIS7ftp.py <victim_ip> <victim_port>

Primer korišćenja bi bio: ./msIIS7ftp.py 192.168.1.22 21, gde 192.168.1.22 predstavlja IP adresu ciljanog servera, a 21 je FTP port.

Funkcija main:

Proverava broj argumenata komandne linije; ako nema tačno tri argumenta (ime skripte, IP adresa i port), prikazuje se uputstvo (usage). Ako su argumenti ispravni, pravi TCP socket (mrežnu vezu). Uzima IP adresu i port iz argumenata i pokušava da se poveže sa serverom. Zatim šalje podatke pomoću s.send(data + '\r\n'), gde bi data trebalo da bude promenljiva koja sadrži sadržaj kao što je buf. Zatvaranje konekcije: Nakon slanja podataka, prikazuje se poruka i socket veza se zatvara.

```
def usage():
        print "usage  : ./msiis7ftp.py <victim_ip>  <victim_port>"
        print "example: ./msiis7ftp.py 192.168.1.22 21"

def main():
    if len(sys.argv) != 3:
            usage()
            sys.exit()

    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    HOST = sys.argv[1]
    PORT = int(sys.argv[2])
    s.connect((HOST,PORT))
    data = s.recv(1024)
    print data
    print "[*] Sending Payload...\n"
    s.send(buf+'\r\n')
    print "[*] Closing Socket...\n"
    s.close()

if __name__ == "__main__":
    main()
```

---

4. Analiza uzroka (root cause)

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

  Ova ranjivost je identifikovana krajem 2010.godine, a otkrivena u Microsoft Internet Information Services (IIS) 7.5, specifično u FTP server komponenti na Windows 7 i Windows Server 2008 R2 operativnim sistemima. Greška omogućava napadaču da izvrši DoS (Denial of Service) napad, što može rezultirati padom FTP servera i gubitkom usluge za legitimne korisnike..

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

  Ne postoji dostupan primer koda zbog stetne prirode.

---

5. Preporuke za mitigaciju

- **Da li je dostupan Vendor Fix ili patch (Da/Ne):Da**
- **Mitigation Strategy:**
  Azurirana verzija koju je Microsoft objavio.

1. Otvori Settings (Postavke) tako što ćeš pritisnuti Windows taster + I.
2. Izaberi Update & Security (Ažuriranje i bezbednost).
3. Klikni na Check for updates (Proveri ažuriranja).
4. Ako postoje nova ažuriranja, klikni na Download and install (Preuzmi i instaliraj).

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