Vulnerability Assessment Report Template

Ime i prezime: Anastasija Savić
Tim: 3
Datum: 03.11.2024.
Scan Tool: Nessus (10.8.3)
Test okruženje: Metasploitable3

---

1. Enumeracija CVE-a

- **CVE ID:** CVE-2016-2183 (SSL Medium Strength Cipher Suites Supported)

- **Opis:** Kriptografski protokoli poput TCP, SSH, često koriste blokovske kriptografske algoritme poput 3DES-a, kako bi šifrovali poruke između klijenata i servera. Ovakvi algoritmi dele podatke na blokove fiksne dužine i zatim svaki blok posebno šifruju. 3DES algoritam koristi blokove dužine 64 bita, što ga čini podložnim „*birthday attack*“ napadima.

    - **Servis**: http
    - **Port**: 631
    - **Protokol**: tcp

---

2. CVSS skor

- **CVSS skor (numerička vrednost):** 7.5 (high)

- **Vektor:** CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
    - **AV(Attack Vector: Network)** - Ranjivost je dostupna preko mreže, što znači da napadač može da iskoristi ranjivost sa udaljene lokacije.
    - **AC(Attack Complexity: Low)** - Napadač može da iskoristi ranjivost relativno lako, bez specijalnih uslova ili dodatnih koraka, posebnih alata ili složene procedure.
    - **PR (Privileges Required: None)** - Napadač ne mora da ima posebna ovlašćenja ili privilegije da bi iskoristio ranjivost. Napadač može da napadne sistem bez potrebe za prijavom ili bilo kakvim predefinisanim pristupom.
    - **UI (User Interaction: None)** - Eksploatacija ranjivosti ne zahteva nikakvu interakciju korisnika.

- **S (Scope: Unchanged)** - Ranjivost utiče samo na komponente unutar istog bezbednosnog domena. Ovo znači da napad ne može preći granice sistema ili aplikacije.
- **C (Confidentiality: High)** - Ranjivost omogućava napadaču da pristupi ili otkrije poverljive informacije. Čime su privatnost i sigurnost podataka, veoma ugrožene.
- **I (Integrity: None)** - Nema značajnog uticaja na integritet podataka. Napadač ne može da menja ili briše informacije, što smanjuje štetu na integritet sistema.
- **A (Availability: None)** - Ranjivost ne utiče na dostupnost sistema. Sistem ostaje funkcionalan i ne doživljava pad performansi ili prekide u radu.

- **Opravdanje:** Kada se koristi protokol poput tls-a, sesije mogu da traju dugo, što omogućava napadaču da prikupi dovoljno enkriptovanih podataka, kako bi pronašao kolizije. Napadač može da iskoristi koliziju kako bi dešifrovao enkriptovane podatke i pristupio poverljivim informacijama, poput HTTP *cookies*, korisničkih podataka, sesijskih tokena. Obim ranjivosti je velik sa obzirom da se 3DES algoritam, kao i neki njemu slični, koriste i dalje u mnogim protokolima.

---

3. Dostupnost eksploita

- **Postoji javno dostupan eksploit (Da/Ne):** Da (https://github.com/azeemba/sour16?tab=readme-ov-file)

- **Opis eksploita:** Eksploit je primer *birthday attack*-a, koji koristi kriptografske algoritme sa malim veličinama blokova. Srž napada je u korišćenju ponovljenih šifrovanih blokova i prepoznavanju slučajeva gde se šifrovani blokovi ponavljaju. Na ovaj način se omogućava dešifrovanje skrivenog sadržaja, kao što je *cookie*. Eksploit se sastoji od skripti. **generate_packets.py** skripta omogućava generisanje enkriptovanih paketa i njihovo čuvanje u datoteku. Podržava -N flag za promenu broja generisanih paketa (u hiljadama). Takođe omogućava konfiguraciju vrednosti *cookies*-a ili veličine bloka. **sour16.py** skripta izvršava stvarni napad i zahteva fajl koji je generisan korišćenjem prethodno pomenute skripte. Pošto veličina bloka može da varira, skripta takođe treba da zna veličinu bloka koja je korišćena za enkripciju.

- **Kod eksploita (ukoliko postoji):** Slika predstavlja glavni deo koda eksploita. Metoda *_decrypt_block*, koja dekodira blokove *cookie* kroz prepoznavanje između šifrovaninh i poznatih blokova.

  Označen deo koda je ključ eksploita jer koristi koliziju kako bi se dešifrovao željeni deo *cookie*-a.

```python
def _decrypt_block(self, ciphertext, plaintext, block_index):
    cipher = ciphertext['cipher']
    encrypted_cookie_blocks = self.encrypted_cookie_blocks
    plain_cookie = self.decrypted_cookie_blocks

    cookie_block = encrypted_cookie_blocks[cipher[block_index]]

    if plain_cookie[cookie_block["index"]] is not None:
        return

    plain = plaintext[block_index]
    # print("Collision found between encryption of block: '{}' and a cookie block index {}."
    #       .format(plain.replace("\n", "\\n"), cookie_block["index"]))

    if block_index != 0:
        prev = cipher[block_index - 1]
    else:
        prev = ciphertext['iv']

    cookie_prev = cookie_block["prev"]
    cookie_plain = self._find_plaintext_from_collision(plain, prev, cookie_prev)
    plain_cookie[cookie_block["index"]] = cookie_plain
```

---

4. Analiza uzroka (root cause)

- **Uvođenje Greške (Commit/Verzija):** Ranjivost je uvedena zbog korišćenja 3DES algoritama i njemu sličnim algoritmima koji rade po istom principu**.** Sweet23 nije rezultat jednog *commit*-a ili promene u kodu, nego potiče iz korišćenja prethodno pomenutih algritama za šifrovanje.
- **Primer Koda (ako je primenljivo):** Na slici može da se vidi primer koda u kom je 3DES korišćen kao algoritam za šifrovanje. Umesto toga trebalo bi definisati neku sigurniju šifru.

```c
#include <openssl/evp.h>

EVP_CIPHER_CTX *ctx = EVP_CIPHER_CTX_new();
EVP_EncryptInit_ex(ctx, EVP_des_ede3_cbc(), NULL, key, iv);
```

---

5. Preporuke za mitigaciju

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

- **Mitigation Strategy:**

- **Alternativni fix (ukoliko ne postoji vendorski):** Da bi se ublažila ranjivost Sweet32, preporučuje se onemogućavanje ili uklanjanje 3DES šifara u TLS ili SSL konfiguraciji i korišćenje jačih algoritama za šifrovanje, poput AES-a. Ovo uključuje izmene u podešavanjima konfiguracije na pogođenim sistemima, kao što su web serveri, VPN prolazi ili drugi mrežni uređaji. Konkretno, preporučuju se sledeće mere za sprečavanje napada:

  1. Veb serveri i VPN-ovi treba da budu konfigurisani tako da preferiraju 128-bitne šifre.
  2. Veb pretraživači treba da nude 3DES samo kao "rezervnu" šifru, kako bi se izbeglo njegovo korišćenje sa serverima koji podržavaju AES, ali preferiraju 3DES.
  3. TLS biblioteke i aplikacije treba da ograniče dužinu TLS sesija sa 64-bitnim šiframa. Ovo se može postići ponovnim pregovaranjem TLS-a ili u nekim slučajevima zatvaranjem konekcije i započinjanjem nove (tj. ograničavanjem HTTP/1.1 Keep-Alive, SPDY i HTTP/2 kada se koriste 3DES paketi šifrovanja). U Apache serveru se to može podesiti tako što se u konfiguracionom fajlu definiše:

     `SSLSessionCacheTimeout 300` (ograničava keširanje sesije na 300 sekundi)

     `SSLRenegBufferSize 1048576`(forsira se ponovno uspostavljanje sesije nakon prenosa 1MB podataka)

  4. Korisnici OpenVPN-a mogu da promene šifru sa podrazumevane Blowfish na AES, koristeći, na primer, šifru *AES-128-CBC* u konfiguraciji klijenta i servera. Ako nemaju kontrolu nad konfiguracijom servera, mogu ublažiti napad tako što će forsirati često ponovno postavljanje ključa komandom *reneg-bytes 64000000*.