Vulnerability Assessment Report Template

Ime i prezime: Stasja Durutovic
Tim: 1
Datum: 03.11.2024.
Scan Tool: Nessus 10.8.3 (#10) WINDOWS
Test okruženje: Metasploitable3

---

1. Enumeracija CVE-a

- **CVE ID: CVE-2004-2761**

- **Opis: SSL Certificate Signed Using Weak Hashing Algorithm**

  CVE-2004-2761 ukazuje na ranjivost u SSL sertifikatima koji koriste kriptografski slabe algoritme za heširanje, kao što su MD2, MD4, MD5 ili SHA-1. Ovi algoritmi su podložni napadima sudara (collision attacks), gde se dva različita unosa mogu mapirati na isti heš. To može omogućiti napadaču da generiše drugi sertifikat s istim digitalnim potpisom, imitirajući time legitimni servis.

  Port i protokol: najčešće se javlja na portovima koji koriste SSL/TLS protokol (443, 993, 995)

---

2. CVSS skor

- **CVSS skor (numerička vrednost): 5**.0
- **Vektor CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N**

  AV-Attack Vector : Network – zahteva lokalni pristup (napadač mora da ima pristup uređaju ili da bude prijavljen na sistem)

  AC –Attack Complexity: Low – eksploatacija je jednostavna I ne zahteva kompeksne tehnike

Privileges Required: **None** – nisu potrebne privilegije ili specijalne dozvole da bi napadač iskoristio ranjivost

UI–User Interaction:None - nije potrebna nikakva akcija korisnika

S- Scope: Unchanged – ne utiče na druge sisteme
C – **Confidentiality**: **None** – napadač ne može da pristupi poverljivim informacija

I-Integrity: High – eksploatacija može narušiti integritet podataka; napadač može da izmeni važne informacije

A-Availability: None – nema uticija na dostupnost, tako da ne uzrokuje pad sistema

- **Opravdanje:**

  Ova ranjivost je klasifikovana kao srednje opasna zbog pristupa preko mreže i niske složenosti napada, takođe i zbog toga što nema potrebe za privilegijama ili korisničkom interakcijom. Iako nema uticaja na poverljivost podataka, visok uticaj na integritet znači da napadači mogu ozbiljno da manipulišu ili oštete važne informacije, što predstavlja značajan rizik za organizaciju. Srednji skor ukazuje na potrebu za pažnjom i predstavlja ozbiljan problem.

---

3. Dostupnost eksploita

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

  https://www.exploit-db.com/exploits/24807

- **Opis eksploita:**

  MD5 je popularni algoritam koji se koristi za heširanje, ali se pokazao kao nesiguran. Ranjivost ovog algoritma omogućava napadačima da naprave dva različita ulaza koja imaju isti MD5 heš. Ovo znači da napadač može stvoriti maliciozni fajl koji ima isti heš kao neki bezazleni fajl. Na primer, napadač može uzeti neki legalan dokument, potpisati ga i kasnije ga zameniti “zloćudnim” dokumentom koji izgleda identično, ali ima drugačiji sadržaj. Ovo može omogućiti izvršavanje malicioznog koda ili lažno predstavljanje poruka. U javno dostupnom eksploitu, koristi se Perl skripta da pokaže ranjivost MD5

heširanja. Skripta pokazuje kako se mogu stvoriti dva različita fajla koji imaju isti MD5 heš, što napadaču omogućava da zamenjuje fajlove.

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

```
# Let:  Vec1 and Vec2 equal our two files ("vectors") with the
same hash

my $vec1 = h2b("
d1 31 dd 02 c5 e6 ee c4 69 3d 9a 06 98 af f9 5c
2f ca b5 87 12 46 7e ab 40 04 58 3e b8 fb 7f 89
55 ad 34 06 09 f4 b3 02 83 e4 88 83 25 71 41 5a
08 51 25 e8 f7 cd c9 9f d9 1d bd f2 80 37 3c 5b
d8 82 3e 31 56 34 8f 5b ae 6d ac d4 36 c9 19 c6
dd 53 e2 b4 87 da 03 fd 02 39 63 06 d2 48 cd a0
e9 9f 33 42 0f 57 7e e8 ce 54 b6 70 80 a8 0d 1e
c6 98 21 bc b6 a8 83 93 96 f9 65 2b 6f f7 2a 70
");

my $vec2 = h2b("
d1 31 dd 02 c5 e6 ee c4 69 3d 9a 06 98 af f9 5c
2f ca b5 07 12 46 7e ab 40 04 58 3e b8 fb 7f 89
55 ad 34 06 09 f4 b3 02 83 e4 88 83 25 f1 41 5a
08 51 25 e8 f7 cd c9 9f d9 1d bd 72 80 37 3c 5b
d8 82 3e 31 56 34 8f 5b ae 6d ac d4 36 c9 19 c6
dd 53 e2 34 87 da 03 fd 02 39 63 06 d2 48 cd a0
e9 9f 33 42 0f 57 7e e8 ce 54 b6 70 80 28 0d 1e
c6 98 21 bc b6 a8 83 93 96 f9 65 ab 6f f7 2a 70
");
```

Srž eksploita su dva vektora koji se koriste za kreiranje dva različita fajla, vec1 i vec2. Oni su definisani tako da imaju isti MD5 heš. To omogućava napadaču da zameni jedan fajl (npr. bezazleni) drugim (npr. malicioznim) bez da se to otkrije.

---

4. Analiza uzroka (root cause)

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

Ova ranjivost se temelji na ranijim otkrićima o slabostima MD5 funkcije, koja je prvi put demonstrirana 2004. godine Ranjivost je primećena u RADIUS protokolu dizajniranom 1991. MD5 algoritam za heširanje je korišćen kao deo RADIUS paketa za generisanje *Response Authenticator* polja, a ranjivost je omogućena korišćenjem napada na koliziju MD5 (chosen-prefix collision attack). **Nema specifičnog commit ID** koji se može navesti kao direktan uzrok ranjivosti, jer je ranjivost vezana za upotrebu MD5 algoritma u RADIUS implementacijama, a ne za određenu verziju ili commit u kodu.

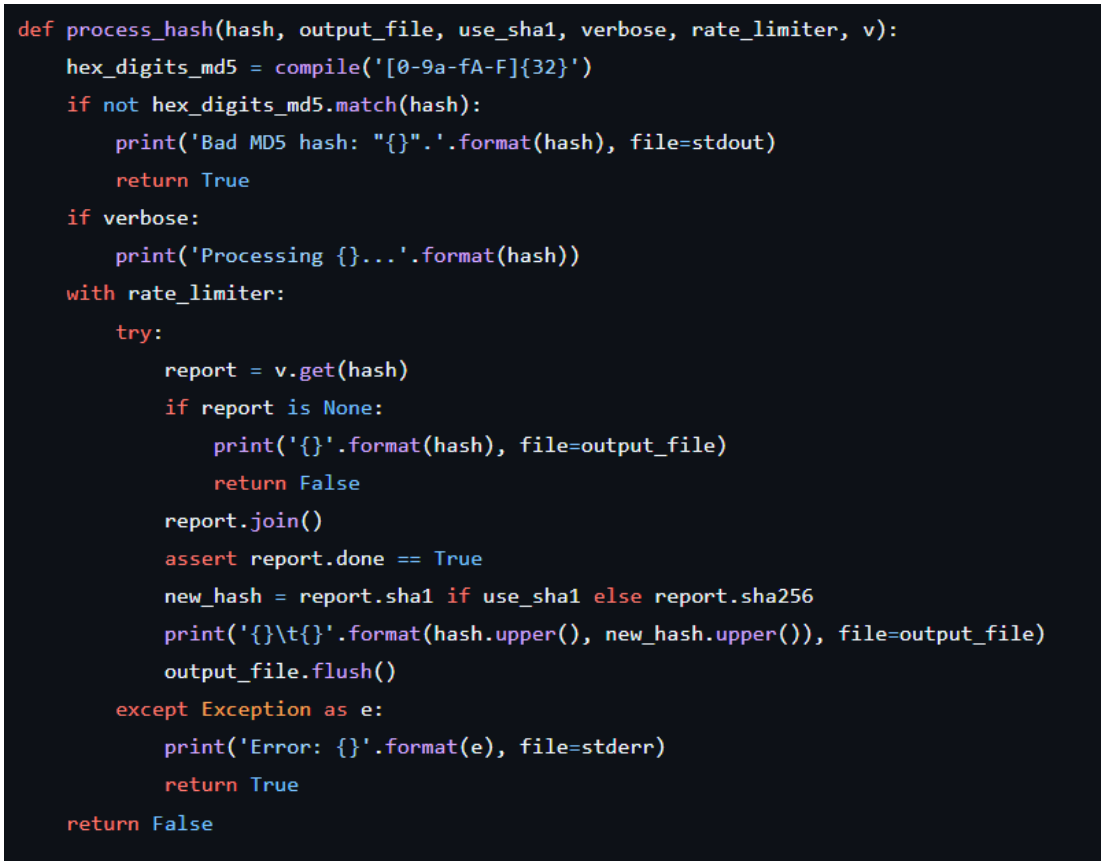
Primer Koda (ako je primenljivo): /

---

5. Preporuke za mitigaciju

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

1. **Prelazak na jače heš algoritme**: Organizacije bi trebale prioritetno migrirati sa MD5 na jače heš funkcije, kao što su SHA-256 ili SHA-1 za sertifikate. S obzirom na to da mnogi sertifikati klase enterprise već koriste SHA-1, važno je da svi novi sertifikati budu izdati sa SHA-1 ili višim standardima. Ova migracija je ključna za očuvanje integriteta i sigurnosti digitalnih potpisa.
   Npr: Sledeća Python skripta koristi API VirusTotal-a da potraži fajl sa odgovarajućim MD5 hešom, a zatim vraća odgovarajući SHA256 heš (ili SHA1, u zavisnosti od opcije). Skripta omogućava korisniku da unese MD5 heš, a zatim, putem VirusTotal-a, pronalazi i ispisuje odgovarajuću SHA256 ili SHA1 vrednost ako je fajl već otpremljen na VirusTotal.



```
def process_hash(hash, output_file, use_sha1, verbose, rate_limiter, v):
    hex_digits_md5 = compile('[0-9a-fA-F]{32}')
    if not hex_digits_md5.match(hash):
        print('Bad MD5 hash: "{}".'.format(hash), file=stdout)
        return True
    if verbose:
        print('Processing {}...'.format(hash))
    with rate_limiter:
        try:
            report = v.get(hash)
            if report is None:
                print('{}'.format(hash), file=output_file)
                return False
            report.join()
            assert report.done == True
            new_hash = report.sha1 if use_sha1 else report.sha256
            print('{}\t{}'.format(hash.upper(), new_hash.upper()), file=output_file)
            output_file.flush()
        except Exception as e:
            print('Error: {}'.format(e), file=stderr)
            return True
    return False
```

Ova funkcija prima MD5 heš kao ulaz i poziva VirusTotal API da potraži fajl sa tim MD5 hešom. Zatim koristi atribute report.sha256 ili report.sha1 da dobije odgovarajući SHA heš ako postoji.

2. **Pregled i ažuriranje politika sertifikata:** Mnogi sertifikacioni autoriteti (CAs) postepeno ukidaju MD5; treba osigurati da su svi novi sertifikati koji se nabavljaju u skladu s tim promenama. Svaka aplikacija ili server koji koristi MD5 potpisane sertifikate treba da se ažurira tako da podržava nove sertifikate. To može uključivati promene u konfiguraciji servera (npr. Apache, Nginx) da bi se osiguralo da koriste nove sertifikate.

   Preporučeni alati: OpelSSL (za upravljanje sertifikatima), Certbot (automatski instalira nove sertifikate) I Puppet ili Ansible.

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