1.1.1 TruffleHog Yüksek Entropili Dize Bulunması (TruffleHog HighEntropy Strings) (CWE-798)

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: Hassas bilgilere yetkisiz erişim, bilgi ifşası

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Kaynak kodlarda ya da yapılandırma dosyalarında gizli dizeler yer verilmesi “TruffleHog Yüksek Entropili Dize Bulunması (CWE-798)” açıklığı olarak ele alınır. TruffleHog yüksek entropili dizeleri kaynak kodlarda arayan bir aracın (tool’un) ismidir. Entropi ise bir rastgelelik ölçütüdür. Kaynak kodlarda gizli dizeler (API anahtarları, kriptografik token’lar (jetonlar), ssl private anahtarları, ssh private anahtarları ve diğer hassas hesap bilgileri) rastgele görünürler. Örn; Aq4Skj92lksn29sSdfk38... şeklinde. Dolayısıyla gizli dizeler sıradan metinlere göre yüksek entropilidirler. TruffleHog aracı da yüksek entropili dizeleri belirler ve şüpheli olarak işaretler. Çünkü hiçbir gizli dize kaynak kodda yer almamalıdır. Konfigürasyon dosyalarına veya sistem ortam değişkenlerine (environment variables) taşınmalıdır. Konfigurasyon dosyasında yer alacaksa da açık bir şekilde yer almamalıdır. “Şifreli” halde yer almalıdır.

Gizli dizeler uygulamaları sızıntılara karşı açık hale getirir. Eğer bir saldırgan kaynak koda erişim elde ederse gömülü gizli dizeleri çalabilecektir ve bunları ilgili varlığı suistimal etmede kullanabilecektir. Örneğin uygulamayı veya kullanıcıyı taklit ederek uzak sisteme (bir veritabanı olabilir, bir web servis olabilir,...) giriş yapma gibi. Saldırgan edindiği gizli dizeler ile uygulamayı veya kullanıcıyı taklit edebildiğinde (oymuş gibi davranabildiğinde) uygulamanın veya kullanıcının yapabileceği herhangi bir şeyi yapabilecektir. Bir saldırgan kaynak kodlara erişerek bu gizli dizeleri elde edebileceği gibi aynı şekilde derlenmiş uygulama binary’lerine tersine mühendislik yaparak da kolaylıkla gömülü gizli dizeleri ortaya çıkarabilir.

TruffleHog Yüksek Entropili Dize Bulunması açıklığını örneklemek maksadıyla güvensiz ve güvenli kod bloklarına yer verilmiştir.

Java - Güvensiz Kod Bloğu:

```java
public class ApiClient {

    // Zafiyet: Yüksek entropili dizeye kaynak kodda yer verilmiş.

    private static final String API_KEY = "AKIAIOSFODNN7EXAMPLE";

    public static void main(String[] args) {

        System.out.println("Using API key: " + API_KEY);
        // Anahtarı kullanarak API'ye bağlantı kurma kodları...

    }
}
```

Güvensiz kod örneğinde api anahtarı kaynak koda gömülmüştür. Bu yüksek entropili bir dizedir ve sızdırılması muhtemel senaryoların önünü açmaktadır. Güvensiz kabul edilmektedir.

Java - Güvenli Kod Bloğu:

```java
public class ApiClient {

    public static void main(String[] args) {

        String apiKey = System.getenv("MY_API_KEY");

        if (apiKey == null) {
            System.err.println("API key not set!");
            return;
        }

        System.out.println("Using API key from environment variable.");

        // API kodlarını devam ettir...

    }
}
```

Web Sunucu Bash Komut Satırı:

```
export MY_API_KEY=AKIAIOSFODNN7EXAMPLE
```

Güvenli kod örneğinde ise api anahtarı sistem ortam değişkenlerinden çekilmektedir. Kaynak koda gizli dizeleri gömmek yerine ortam değişkenlerine konulabileceği gösterilmiştir. Bu güvenli kullanımın alternatifi olarak yapılandırma dosyaları seçeneği de değerlendirilebilir.

Konfigurasyon Dosyası - Güvensiz Ayar:

```
# application.yml
apiKey: AKIAIOSFODNN7EXAMPLE
```

Bu güvensiz yapılandırmada gizli dize yapılandırma dosyasına şifrelenmeden yerleştirilmiştir. Gizli dizeler konfigürasyon dosyalarına belirlenecek bir anahtar ve şifreleme algoritması ile şifrelenerek yerleştirilmelidir. Şifreli haldeyken web sunucuda uygulama çalışma sırasında deşifreleme ile kullanılabilir. Sonuç olarak konfigürasyon dosyası seçeneği de güvenli bir uygulama için kullanılabilir.

Kurum uygulamada TruffleHog Yüksek Entropili Dize (CWE-798) açıklığı tespit edilmiştir.

::::::::::::::BULGU::::::::::::::

Açıklığın Önlemi:

Tavsiyeler şu şekildedir:

- Kaynak koda herhangi bir gizli veri koyulmamalıdır.
- Kullanıcı parolaları bir veri tabanında veya bir dizin hizmetinde depolanmalıdır ve güçlü şifreleme ile (örn; bcrypt, scrypt, PBKDF2, or Argon2) koruma altına alınmalıdır. Kullanıcı parolaları açık bir şekilde kaynak koda konulan değerler ile kıyaslanmamalıdır.
- Sistem parolaları bir konfigürasyon dosyasına veya veri tabanına depolanmalıdır ve güçlü bir şifreleme ile (örn; AES-256) koruma altına alınmalıdır. Şifreleme anahtarları güvenli bir şekilde yönetilmelidir ve açık bir şekilde konulmamalıdır.
- Diğer gizli dizeler ortam değişkenlerine konulmalıdır ve kaynak kodlarda ortam değişkenlerinden çekilmek suretiyle kullanılmalıdır.
- Alternatif olarak diğer gizli dizeler konfigürasyon dosyasına konulmalıdır ama belirlenen bir şifreleme anahtarı ve algoritması ile şifreli halde bulundurulmalıdır.

Referanslar:

1. https://cwe.mitre.org/data/definitions/798.html
2. https://github.com/trufflesecurity/trufflehog