Testing for Sensitive Information Sent via Unencrypted Channels (Şifrelenmemiş Kanallar Üzerinden Gönderilen Hassas Bilgilerin Testi)

Summary (Özet)

Hassas veriler ağ üzerinden iletildiğinde korunmalıdır. Veriler HTTPS üzerinden iletilirse veya başka bir şekilde şifrelenirse, koruma mekanizmasının sınırlamaları veya güvenlik açıkları olmamalıdır, daha geniş makalede belirtildiği gibi, Zayıf Taşıma Katmanı Güvenliği için Test ve diğer OWASP belgelerinde açıklanmaktadır:

- OWASP Top 10 2017 A3-Sensitive Data Exposure.
- OWASP AFS - Doğrulama V9.
- Taşıma Tabaka Koruma Çit Sac.

Veriler depolandığında korunması gerekiyorsa, bu verilerin bu veri iletim sırasında da korunması gerekir. Hassas veriler için bazı örnekler şunlardır:

- Kimlik doğrulamada kullanılan bilgiler (ör. Kimlik bilgileri, PIN'ler, Oturum tanımlayıcıları, Jetonlar, Kurabiyeler...)
- Yasalar, düzenlemeler veya belirli örgütsel politika ile korunan bilgiler (örneğin. Kredi Kartları, Müşteri verileri)

Uygulama hassas bilgileri şifrelenmemiş kanallar aracılığıyla iletirse - örneğin. HTTP - güvenlik riski olarak kabul edilir. Bazı örnekler, HTTP üzerinden düz metinde kimlik doğrulama kimlik bilgilerini
gönderen Temel kimlik doğrulaması, HTTP üzerinden gönderilen temel kimlik doğrulama kimlik bilgilerini veya düzenlemeler, yasalar, organizasyon politikası

veya uygulama iş mantığı nedeniyle hassas olarak kabul edilen diğer bilgilerin düz metin iletimi gönderir.

Kişisel Tanımlayıcı Bilgiler (PII) için örnekler şunlardır:

- Sosyal güvenlik numaraları
- Banka hesap numaraları
- Pasaport bilgileri
- Sağlık hizmetleri ile ilgili bilgiler
- Tıbbi sigorta bilgileri
- Öğrenci bilgileri
- Kredi ve banka kartı numaraları
- Sürücüler lisans ve Devlet kimliği bilgileri

Test Objectives (Test Hedefleri)

- Çeşitli kanallar üzerinden iletilen hassas bilgileri belirleyin.
- Kullanılan kanalların gizliliğini ve güvenliğini değerlendirin.

How To Test (Nasıl Test Edilir)

Korunması gereken çeşitli bilgi türleri, uygulama tarafından net metinle iletilebilir. Bu bilgilerin HTTPS yerine HTTP üzerinden iletilip iletilmediğini veya zayıf şifrelerin kullanılıp kullanılmadığını
kontrol etmek mümkündür. OWASP En İyi 10 2017 A3-Duyarlı Veri Maruziyeti veya Taşıma Katmanı Koruma Hile Sayfasının güvensiz iletimi hakkında daha fazla bilgi edinin

.

Example 1: Basic Authentication over HTTP (Örnek 1: HTTP üzerinden Temel Kimlik Doğrulama)

Tipik bir örnek, Temel Kimlik Doğrulamanın HTTP üzerinden kullanılmasıdır. Temel Kimlik Doğrulama kullanırken, kullanıcı kimlik bilgileri şifrelenmek yerine kodlanır ve HTTP başlığı olarak gönderilir. Aşağıdaki örnekte testçi bu sorunu test etmek

için curl kullanır. Uygulamanın HTTPS yerine Temel kimlik doğrulamayı nasıl kullandığını ve HTTP'yi nasıl kullandığını unutmayın

```
$ curl -kis http://example.com/restricted/
HTTP/1.1 401 Authorization Required
Date: Fri, 01 Aug 2013 00:00:00 GMT
WWW-Authenticate: Basic realm="Restricted Area"
Accept-Ranges: bytes Vary:
Accept-Encoding Content-Length: 162
Content-Type: text/html

<html><head><title>401 Authorization Required</title></head>
<body bgcolor=white> <h1>401 Authorization Required</h1>  Invalid login crede
ntials!  </body></html>
```

Example 2: Form-Based Authentication Performed over HTTP (Örnek 2: Form Tabanlı Kimlik Doğrulama HTTP üzerinden gerçekleştirildi)

Bir başka tipik örnek, kullanıcı kimlik doğrulama kimlik bilgilerini HTTP üzerinden ileten kimlik doğrulama formlarıdır. Aşağıdaki örnekte, HTTP'nin içinde kullanıldığını görebilir `action` Formun özniteliği. Bu konuyu bir müdahale vekili ile HTTP trafiğini inceleyerek görmek de mümkündür.

```
<form action="http://example.com/login">
   <label for="username">User:</label> <input type="text" id="username" na
   me="username" value=""/><br />
   <label for="password">Password:</label> <input type="password" id="passv
   ord" name="password" value=""/>
   <input type="submit" value="Login"/>
</form>
```

Example 3: Cookie Containing Session ID Sent over HTTP (Örnek 3: Çerez Oturum Kimliği Içeren HTTP üzerinden gönderildi)

Oturum ID Çerezi korumalı kanallar üzerinden iletilmelidir. Kurcanın güvenli bayrak seti yoksa, uygulamanın şifrelenmemiş olarak iletmesine izin verilir. Aşağıdaki çerezin ayarını Güvenli bayrak olmadan yapılır ve işlemdeki tüm giriş HTTP'de gerçekleştirilir ve HTTP'de değil HTTP'de yapılır.

```
https://secure.example.com/login

POST /login HTTP/1.1
Host: secure.example.com
[...]
Referer: https://secure.example.com/
Content-Type: application/x-www-form-urlencoded
Content-Length: 188

HTTP/1.1 302 Found
Date: Tue, 03 Dec 2013 21:18:55 GMT
Server: Apache
Set-Cookie: JSESSIONID=BD99F321233AF69593EDF52B123B5BDA; expires=Fri
n-2014 00:00:00 GMT; path=/; domain=example.com; httponly
Location: private/
Content-Length: 0
Content-Type: text/html
```

```
http://example.com/private

GET /private HTTP/1.1
Host: example.com
[...]
Referer: https://secure.example.com/login
Cookie: JSESSIONID=BD99F321233AF69593EDF52B123B5BDA;

HTTP/1.1 200 OK
Content-Type: text/html;charset=UTF-8
```

```
Content-Length: 730
Date: Tue, 25 Dec 2013 00:00:00 GMT
```

Example 4: Testing Password Sensitive Information in Source Code or Logs (Örnek 4: Kaynak Kodunda veya Günlüklerde Şifre Hassas Bilgilerin Test Edilmesi)

Hassas bilgileri aramak için aşağıdaki tekniklerden birini kullanın.

Şifre veya çıkıntı anahtarının kaynak kodunda veya yapılandırma dosyalarında sabit olup olmadığını kontrol etmek.

```
grep -r –E "Pass | password | pwd |user | guest| admin | encry | key | decrypt | sharekey " ./PathToSearch/
```

Günlüklerin veya kaynak kodunun telefon numarası, e-posta adresi, kimlik veya başka bir PII olup olmadığını kontrol etmek. PII formatına göre düzenli ifadeyi değiştirin.

```
grep -r " {2\}[0-9]\{6\} "  ./PathToSearch/
```

Tools (Araçlar)

- curl
- grep
- Identity Finder
- Wireshark
- TCPDUMP