Ders 1 - Insecure Logging (Güvensiz Log'lama)

Dersin Hedefi

Diva uygulamasındaki "1. INSECURE LOGGING" seçeneğine tıklayın. Bu seçenek ile açılan sayfadaki log'lama faaliyetlerinin android sistemde nereye yapıldığını ve nasıl bir formda (türde) kayıt aldıklarını tespit edin. Bunun sonucunda log kayıtlarına herhangi bir hassas veri düşmüş mü tespit edin. Ayrıca sayfada çalışan kodların ne şekilde zafiyetli kodlandığını gösterin.

Dersin Açıklaması

Uygulama geliştiricileri uygulamalarında herhangi bir nedenle hatalar meydana geldiğinde sorunun kaynağını anlamak adına uygulamalarında hata bilgilerini log'layacak log'lama kod satırları bulundururlar. Bu şekilde hatayı tespit etme ve çözme prosedürünü gerçekleştirirler. Android uygulamaları arasında Log sınıfını (class'ını) kullanan uygulamalar bu sınıfın metotları ile android sistemdeki log dosyalarına log kayıtları düşerler. Bu şekilde sonradan android uygulama hataları gerçekleştiğinde sorunu anlamak için gerekli hata bilgilerini log dosyalarından okuyarak çözüme yaklaşırlar.

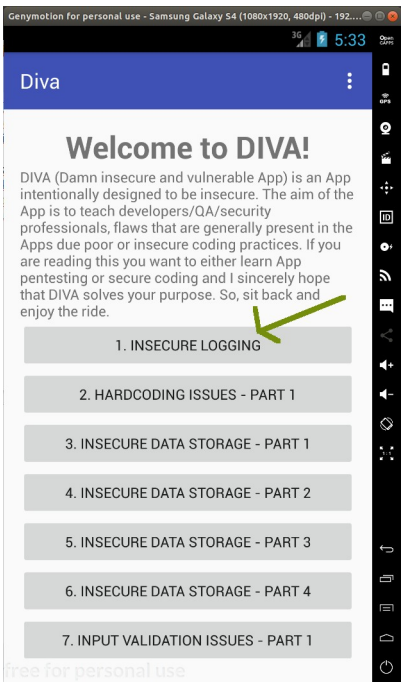
Android uygulamalarda geliştiriciler sayfalardaki kullanıcı girdi noktalarında girdi kaynaklı karşılaşılabilecek hatalar nedeniyle sayfayı sunan / kontrol eden java kaynak kodlarına Log sınıfı metotları koyabilmekteler. Örn; Log.v("UygulamaTest", "uygulamaSyf1" + gelenKullanıcıGirdisi1), Log.d("UygulamaTest", "UygulamaSyf2" + gelenKullanıcıGirdisi2), Log.i(...),…v.b. Bu kullanıcı taraflı girdilerinin log'lanması işlemi yaygın bir şekilde mevcuttur. Bu kullanım kullanıcı girdilerinin hassas nitelik taşıdığı durumlarda risk arz eder. Çünkü log'lardaki veriler başkalarınca okunabilme olasılığına sahiptir ve kullanıcı hassas bilgilerinin ifşasına sebebiyet verebilir.

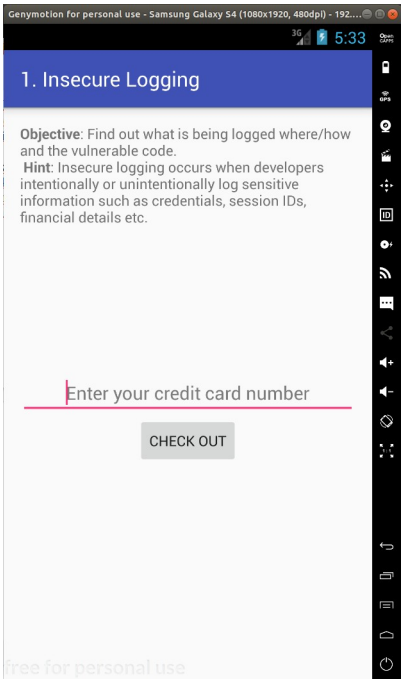
Android sistemlerde log dosyalarını okumak için resmi Android SDK araçları arasında yer alan logcat aracı kullanılabilir. Bu araç android sistemdeki log kayıtlarını ekranda anlık olarak gösterir ve gerekli hata incelemeleri / takibi yapılabilir.

Bu derste diva uygulaması Insecure Logging sayfasında kullanılan log'lama prosedürünün güvensiz kodlanışı nedeniyle ne tür bir açığa sebebiyet verdiği ve neden güvensiz kodlama örneği sergilediği gösterilecektir.

Dersin Çözümü

Ders ekranına bir göz atalım.

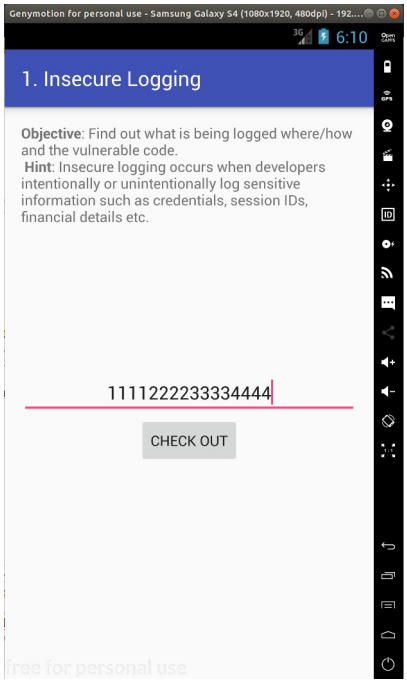




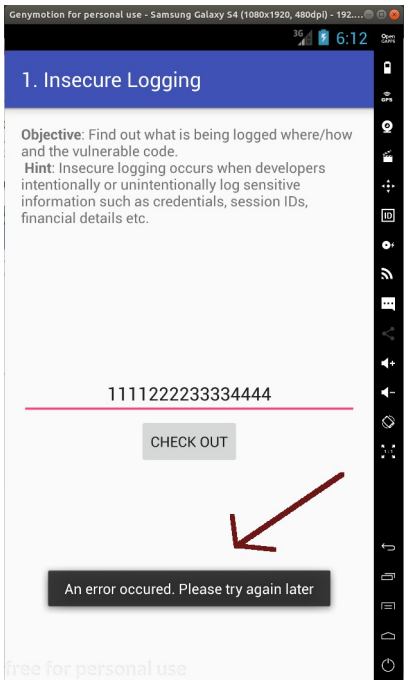
Ders ekranında kredi kart numarası girmemizi isteyen bir metin kutusu vardır. Bu metin kutusuna kredi kart numarası girilecektir ve “Check Out” ile hesabı öde butonuna basılacaktır. Bu şekilde bir alışveriş sonucu hesap ödeme işlemi senaryosu uygulanacaktır.

Uygulamada bu hesap ödeme işlemi aşamasında eğer hata yaşanırsa standart hata log’lama prosedürü gereği geliştirici kodlaması ile bir log kaydı hazırlanacaktır ve sistem log’larına bir kayıt eklenecektir. Bizim amacımız uygulamanın bu sayfasında yaşancak hata akabinde hata log’lamasının nereye yapıldığını tespit etmek ve log kaydının ne türden bir veri içerdiğini tespit etmektir.

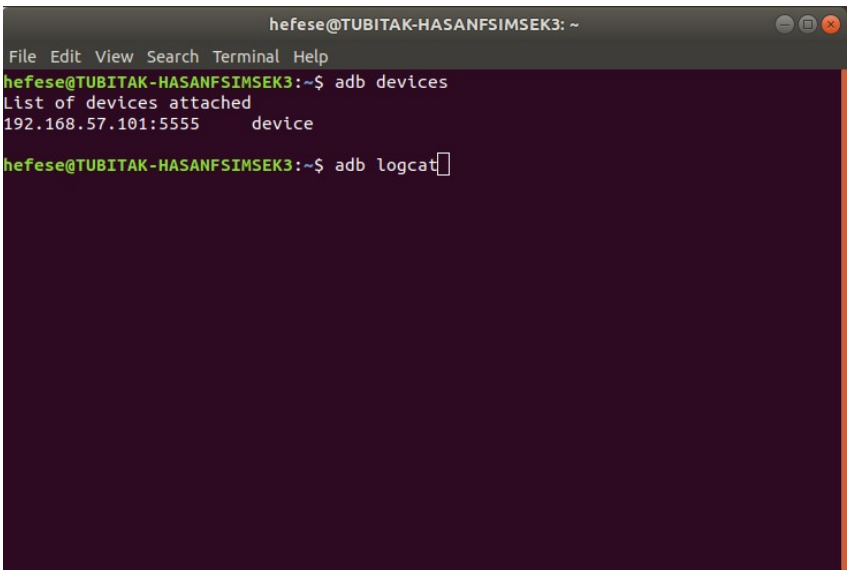
Şimdi ekrandaki metin kutusuna kredi kart numarasına benzer bir numara girelim.



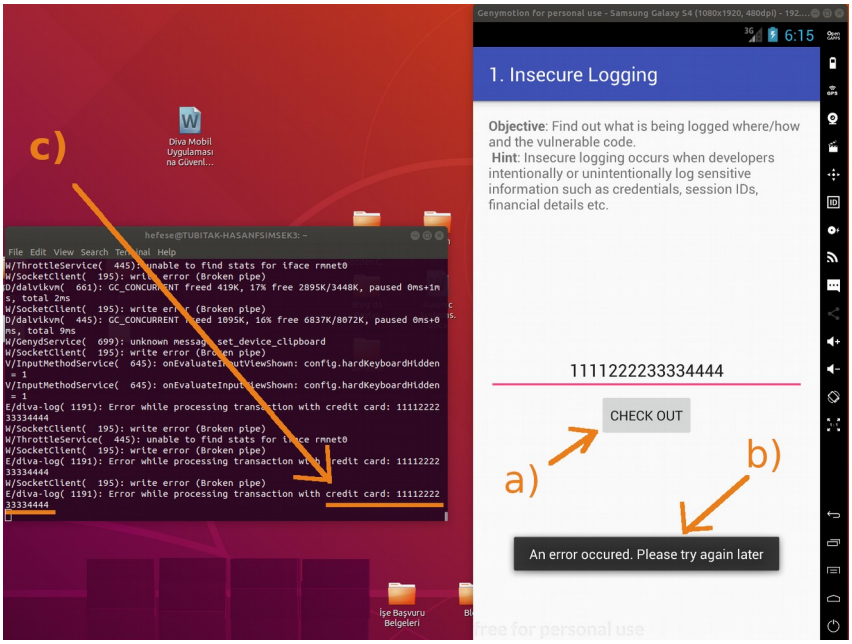
“Check Out” butonu ile hesabı öde diyelim.



Ekranda bir hata mesajı görülecektir. Uygulama kredi kart numaramızı geçerli bulmadığına (amiyane tabirle beğenmediğine) dair bir hata sunmaktadır. Şimdi bu uygulama sayfasının hata vermesi sonucu log kaydını android sistemde nereye eklediğini tespit edelim, sonra ise log kaydının ne tür bir veri içerdiğini görelim. Bunu bilgisayardan adb aracı ile mobil cihaza bağlanıp mobil cihazdaki log kayıtlarını logcat aracı ile görüntüleyerek yapabiliriz.



(Bilgisayardan Android Mobil Cihazdaki Log Kayıtlarına Bakılır)



(Uygulama Hata Ürettiğinde Log Kayıtlarına Uygulama Hata Bilgisi ve Hassas Bir Girdi Yansır)

Görüldüğü üzere mobil uygulamada “Check Out” butonuna bastığımızda gelen hata sonucunda adroid sistem log kayıtlarına uygulama hata bilgisi ve ilave olarak kullanıcı kredi kart numarası bilgisi gelmiştir. Bu bize uygulamanın güvensiz bir şekilde log’lama yaptığını gösterir. Çünkü android sistem log kayıtlarına uygulama hatası eklenirken aynı zamanda kullanıcı kişisel verisi de (kullanıcı hassas bir giridisi de) eklenmiştir. Log kayıtları açık bir şekilde okunabilen metin belgesi olduklarından bu sonraki zamanlarda yaşanabilecek mobil cihaza sızma aktiviteleri sonrası saldırganların log kayıtlarını incelemesi / ziyaret etmesiyle kullanıcı hassas verilerine erişebileceği anlamına gelmektedir.

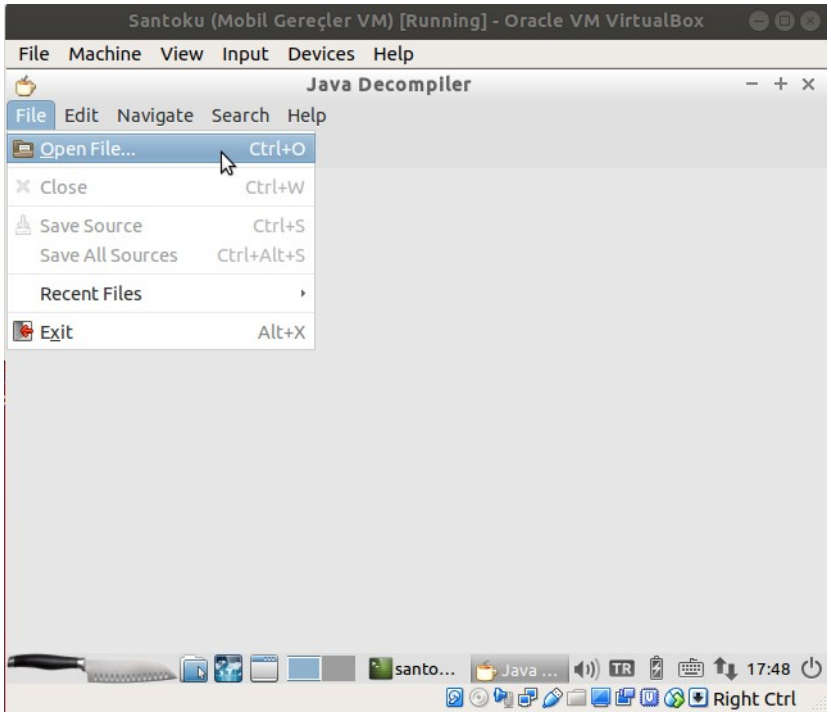
Dersin hedefi başlığında sonraki hedef diva uygulamasının görüntülemekte olduğumuz sayfasında kullanıyor olduğu güvensiz log kaydı tutma kodunu tespit etmekti. Bu işlem için bu makale dizisinin en başında yaptığımız uygulama binary dosyasını (.apk dosyasını) bilgisayara çekme,

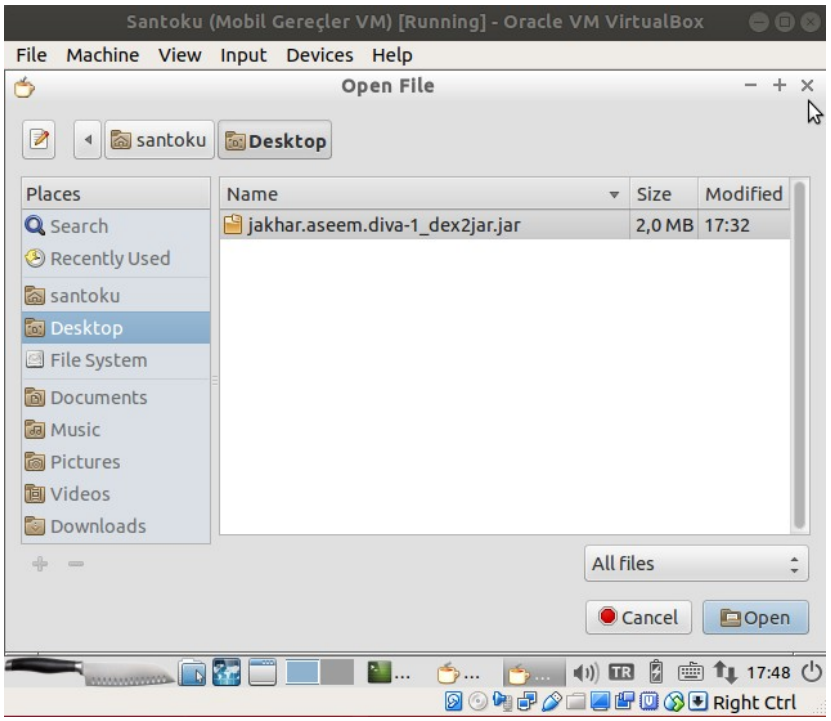
üzerinde tersine mühendislik yapma ve uygulamanın okunabilir java dosyalarını elde etme işini tekrarladığımızı varsayalım. Uygulamanın okunabilir java dosyalarını JD-GUI editörü ile inceleyerek bu uygulama sayfasını (“Insecure Logging” sayfasını) sunan / kontrol eden java dosyasını tespit edelim. Bahsedilen işlemlerin ayrıntısı için bkz. Başlangıç: Android Uygulamalarda Tersine Mühendislik ile Okunabilir Java Kaynak Kodları Elde Etme (Dex2Jar, JD-GUI, ApkTool)

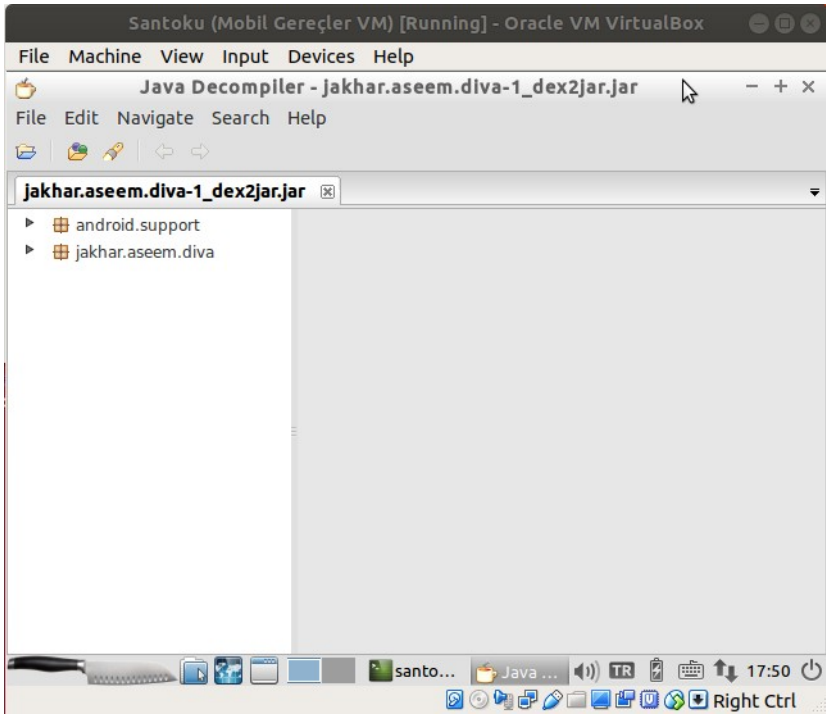
Santoku Linux Terminal:

> jd-gui

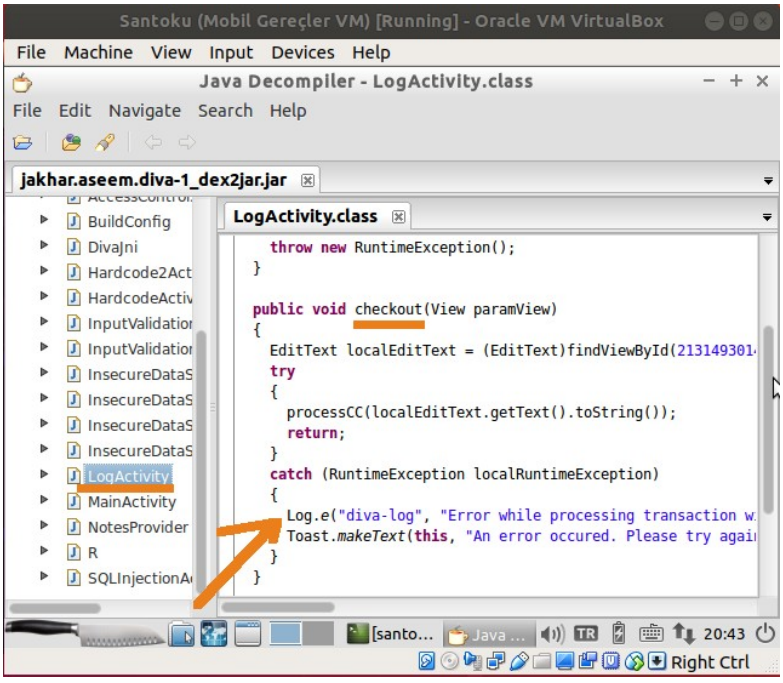
Çıktı:







LogActivity.class adlı java dosyası incelediğimizde uygulamanın görüntülüyor olduğumuz sayfasını çalıştırdığı görülebilir.



Java kaynak kod dosyaları incelendiğinde uygulamada görüntülüyor olduğumuz ekranın hangi java dosyası tarafından sunulduğu / kontrol edildiği yukarıdaki gibi görülebilir. Bu örnek için java dosyasını bulmak uygulama ekranındaki sayfa ismi ile benzerliği dolayısıyla kolay olmuştur ama gerçek bir senaryoda java dosyaları arasında inceleme yapmak ve doğru java dosyasını bulmak için okumalar yapmak gerekebilir.

İlgili .java dosyasındaki kodları inceleyecek olursak kodların nasıl çalıştığını detaylıca sorgulamak yerine kodlarda yer alan metotların isimlerine bakarak ne işler döndüğünü anlayabiliriz. checkout() metodu hesap ödeme işlemini gerçekleştirmek için vardır ve içerisindeki processCC() metodu uygulama sayfasındaki metin kutusundan gelen kredi kart (credit card (CC)) numarasını alıp işleme sokmak için vardır. process() metodu hata ürettiğinde (örneğin geçersiz bir kredi kart numarası aldığında) catch() adlı hata yakalama metoduna geçilir. catch() metodunda ise hata log'lama amacıyla bir Log sınıfı metodu kullanımı mevcuttur: Log.e(). Kullanıcı ekranında hata meydana geleceği zaman catch() metodu içerisine dallanılır ve içerdeki Log.e() metodu ile ekranda girilen kredi kart bilgisi android sistem log dosyalarına eklenir. İlgili Log.e() log kaydı tutma satırı tamamı şu şekildedir:

>> Güvensiz Log Kaydı Tutma Kod Satırı

```
Log.e("diva-log", "Error while processing transaction with credit card: " +
localEditText.getText().toString());
```

Görüldüğü üzere "Error …. with credit card" cümlesi sonuna kullanıcıdan gelen kredi kart bilgisi eklenmektedir. Hatırlarsanız bilgisayardan mobil cihaza bağlanıp logcat aracı ile mobil cihazdaki log kayıtlarına baktığımızda şu şekilde bir log kaydı görüntülemiştik:

```
...
...
E/diva-log(1911): Error while processing transaction with credit card: 1111222233334444
```

Dolayısıyla bu log tutma kod satırı güvensiz kodlanmıştır. Hata cümlesi sonuna kullanıcı hassas girdisi öylece eklenmemesi gerekirdi. Bu şekilde güvensiz log'lama dersinde android diva uygulamasının log kayıtlarını nereye yaptığı, ne şekilde yaptığı tespitini, ardından hatalı / güvensiz log kaydı tutan kod satırının tespitini yapmış olduk.

Sonuç

Geliştiriciler kullanıcı bir hatayla karşılaştığında ve çözemediğinde kullanıcının sorununu çözmek için kullanıcıdan hata bilgisini ve log dosyasını talep etmeyi hedeflerler. Bu sayede sorunu uzaktan anlayıp çözebilecektirler. Ama bu log'lama prosedürleri bu derste olduğu gibi hassas girdilerin olduğu noktalarda bu şekilde yürütülürlerse kullanıcı mobil cihazına sonradan sızacak saldırganların bu bilgileri elde etmesinin yolu açılmış olur.

Normal şartlarda örneğin kullanıcı cihazında uygulama içerisinde girilen hassas veriler uygulama sunucusuna şifreli gidip uygulama sunucusunda şifreli halde tutulurlar. Fakat eğer kullanıcı cihazında uygulamaya girilen hassas veriler kullanıcı cihazında yaşanan hata dolayısıyla kullanıcı cihazındaki log dosyalarına kaydedilirse bu hassas veriler log metin belgelerinde açık metin halinde yer alacaklarından risk arz ederler.

Kötü niyetli kimseler kullanıcıların mobil cihazlarına sızma faaliyetleri yürütecekleri zaman elleri altında kurban mobil cihazda / sistemde ziyaret edilecek dosyalar listesine sahiptirler. Bu dosyalar listesi içerisindeki dosyalardan bir tanesi de log dosyalarıdır. Örneğin mobil cihazdaki bir uygulamada eğer gereksiz derecede fazla log tutma faaliyeti yürütülüyorsa veya uygulama hassas kullanıcı verileri girme noktalarında log'lama faaliyeti yürütülüyorsa saldırganlar bu verilere erişip okuyabilecektirler.

Dolayısıyla geliştiricilerin hataların çözümü noktasında başvurdukları log'lama faaliyeti yürüten android Log class'ı ve metotları geliştiriciler tarafından uygulama, uygulama mağazasından yayına çıkmadan önce bol şekilde kullanılabilir. Fakat uygulama, uygulama mağazasından servis edileceği zaman çok dikkatli bir şekilde Log class'ı ve metotları uygulama kod satırlarında yer almalıdır. Uygulamadaki log kaydı tutma satırları artık ne fazla miktarda hata bilgisi kaydı tutmalıdır ve ne de kullanıcı hassas girdisi içeren hata bilgileri kaydı tutmalıdır. Bunun yerine özenle hazırlanmış ve öz log kaydı tutma prosedürü takip edilmelidir. Hassas girdi noktalarında olası meydana gelebilecek hatalar için log'lama mekanizması konulması gerekiyorsa hata konumunu bilgilendirici (hassas veriyi içermeyen) bir not / kayıt tutulabilir ve böylece geliştirici sorunun nereden patlak verdiğine bakarak çözüm yoluna gidebilir.

Sonuç olarak saldırganlar bu zafiyete sahip diva uygulamasının kullanıldığı mobil cihazlara şayet sızacak olurlarsa kullanıcılar arasında e-alışveriş yapmayı deneyip hata alanların kredi kart numaraları log kayıtlarına düşeceğinden saldırganlarca okunabilecektir / elde edilebilecektir.

Kaynaklar

https://developer.android.com/studio/command-line/logcat
https://developer.oculus.com/documentation/mobilesdk/latest/concepts/mobile-logcat/
https://developer.android.com/reference/android/util/Log
https://stackoverflow.com/questions/7959263/android-log-v-log-d-log-i-log-w-log-e-when-to-use-each-one
https://code.tutsplus.com/tutorials/android-essentials-application-logging--mobile-4578