CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

RUHR UNIVERSITÄT BOCHUM

Unchained Skies: A Deep Dive into Reverse Engineering and Exploitation of Drones

Nico Schiller

Moritz Schloegel

Who we are

Nico Schiller

- researcher @ CISPA
- interested in drones and their security
- fuzz all the things!

Moritz Schloegel

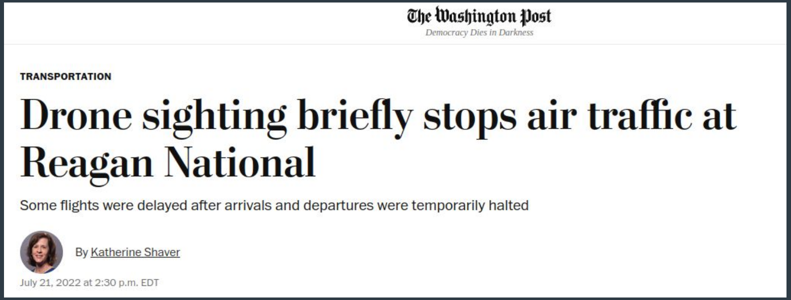
- also researcher @ CISPA
- interested in automated bug finding, mostly fuzzing
- obfuscation / deobfuscation (Next-gen VMs talk at REcon22)

Consumer Drones

Why Drones?

BUT

Where things can go wrong: Airports



The Washington Post

Democracy Dies in Darkness

TRANSPORTATION

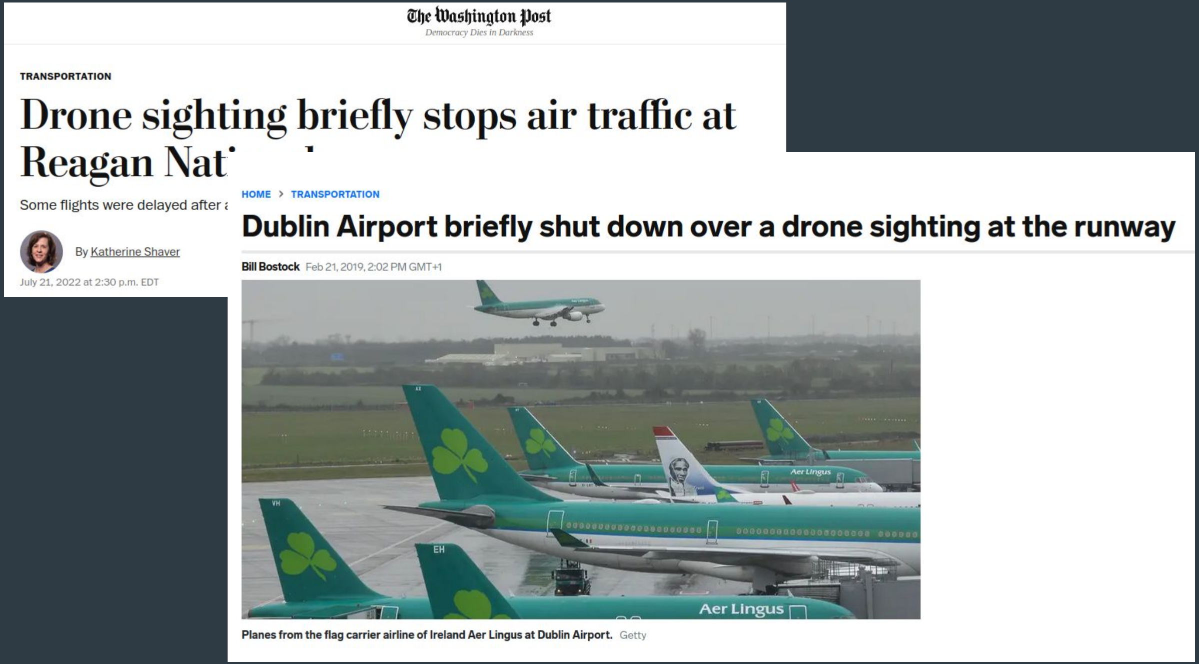
Drone sighting briefly stops air traffic at Reagan National

Some flights were delayed after arrivals and departures were temporarily halted

By Katherine Shaver

July 21, 2022 at 2:30 p.m. EDT

Where things can go wrong: Airports



The Washington Post
Democracy Dies in Darkness

TRANSPORTATION

Drone sighting briefly stops air traffic at Reagan Nat

Some flights were delayed after a

By Katherine Shaver
July 21, 2022 at 2:30 p.m. EDT

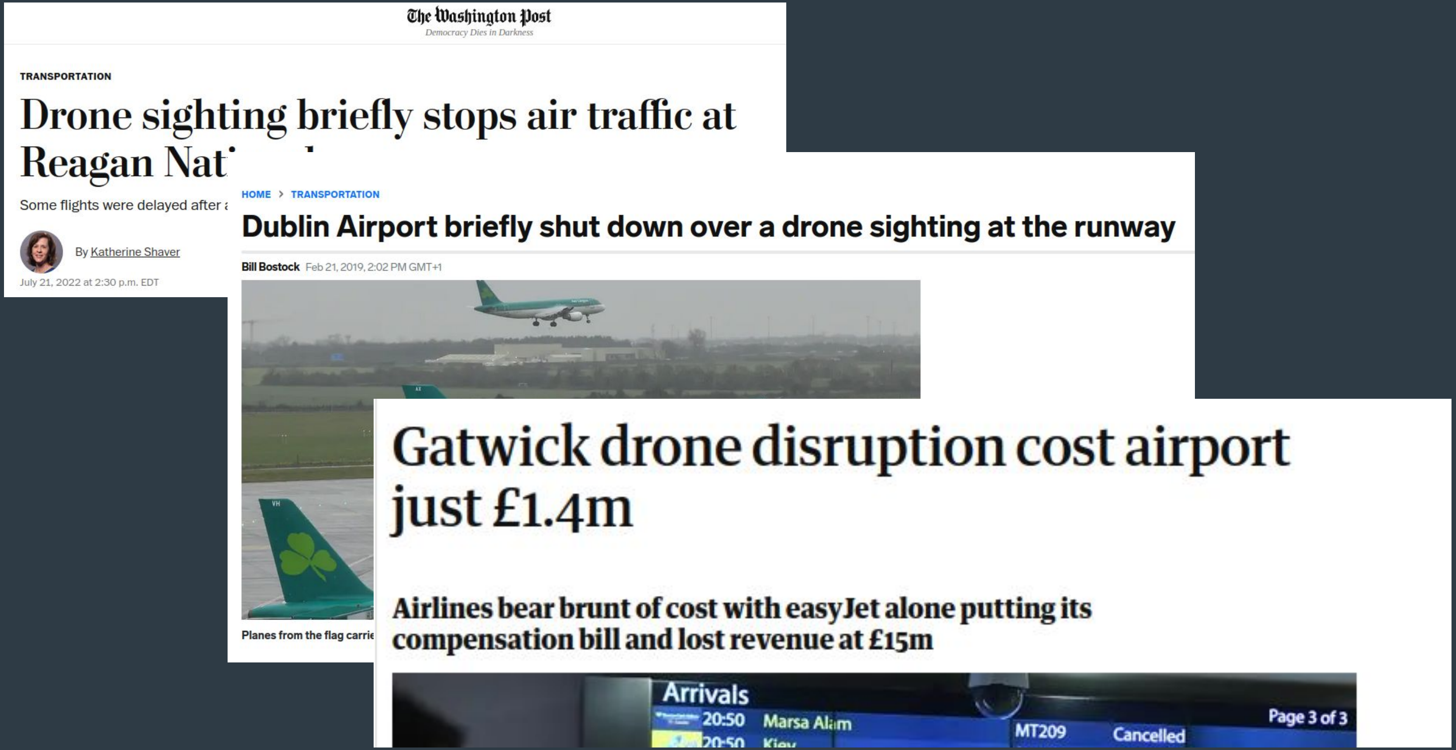
HOME > TRANSPORTATION

Dublin Airport briefly shut down over a drone sighting at the runway

Bill Bostock Feb 21, 2019, 2:02 PM GMT+1

Planes from the flag carrier airline of Ireland Aer Lingus at Dublin Airport. Getty

Where things can go wrong: Airports



The Washington Post
Democracy Dies in Darkness

TRANSPORTATION

Drone sighting briefly stops air traffic at Reagan Nat

Some flights were delayed after a

By Katherine Shaver

July 21, 2022 at 2:30 p.m. EDT

HOME > TRANSPORTATION

Dublin Airport briefly shut down over a drone sighting at the runway

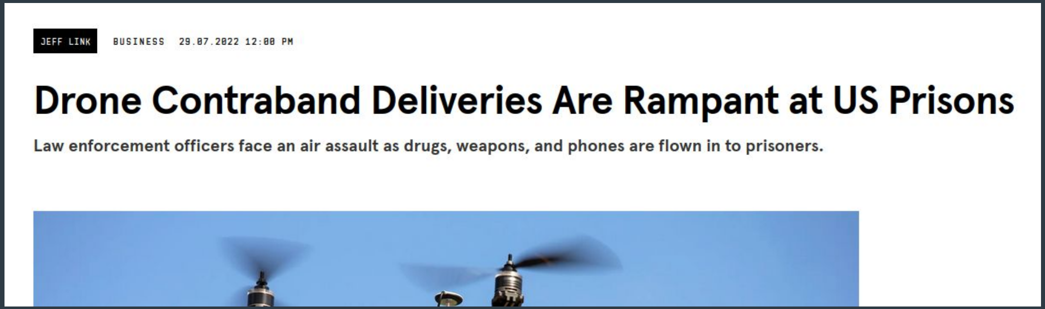
Bill Bostock Feb 21, 2019, 2:02 PM GMT+1

Planes from the flag carrie

Gatwick drone disruption cost airport just £1.4m

Airlines bear brunt of cost with easyJet alone putting its compensation bill and lost revenue at £15m

Where things can go wrong: Prisons

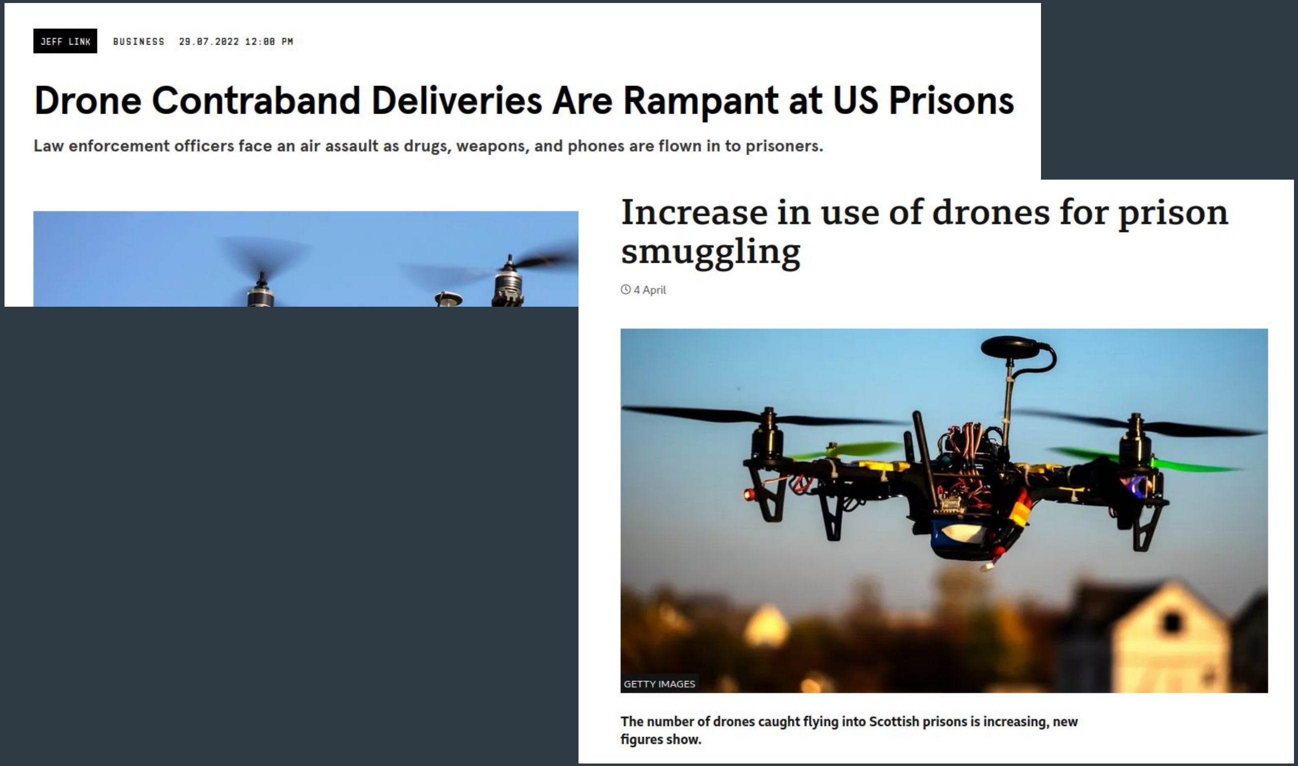


JEFF LINK BUSINESS 29.07.2022 12:00 PM

Drone Contraband Deliveries Are Rampant at US Prisons

Law enforcement officers face an air assault as drugs, weapons, and phones are flown in to prisoners.

Where things can go wrong: Prisons



JEFF LINK BUSINESS 29.07.2022 12:00 PM

Drone Contraband Deliveries Are Rampant at US Prisons

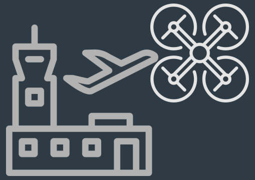
Law enforcement officers face an air assault as drugs, weapons, and phones are flown in to prisoners.

Increase in use of drones for prison smuggling

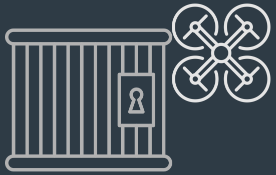
4 April

GETTY IMAGES

The number of drones caught flying into Scottish prisons is increasing, new figures show.



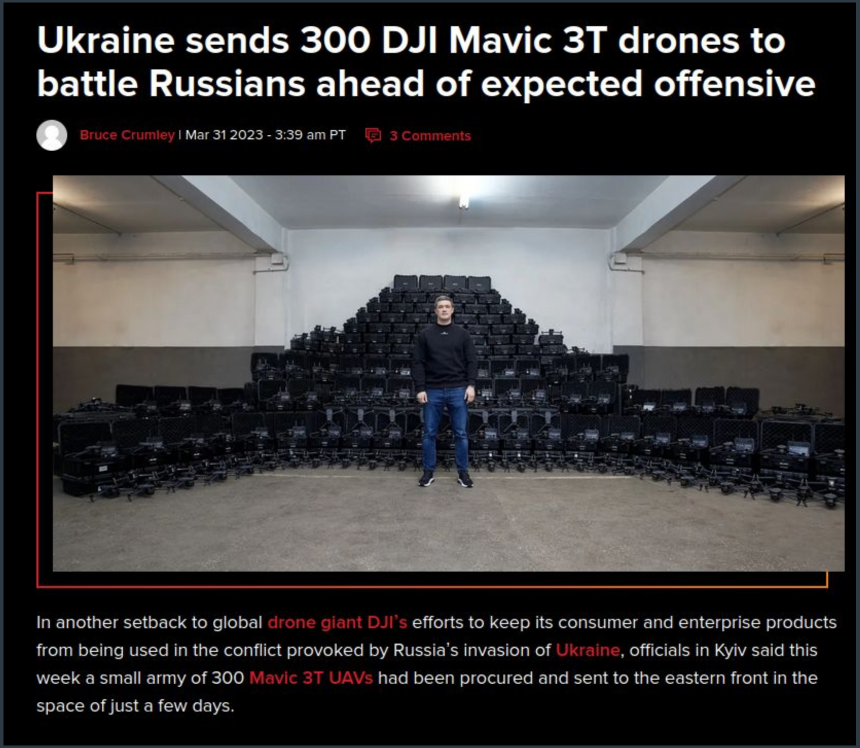
- Block airport operations
- Expensive shutdowns



- Bypass physical barriers
- Smuggling

Low entry barrier for air mobility in a traditionally heavily regulated sector!

Recent Scenario: Conflicts

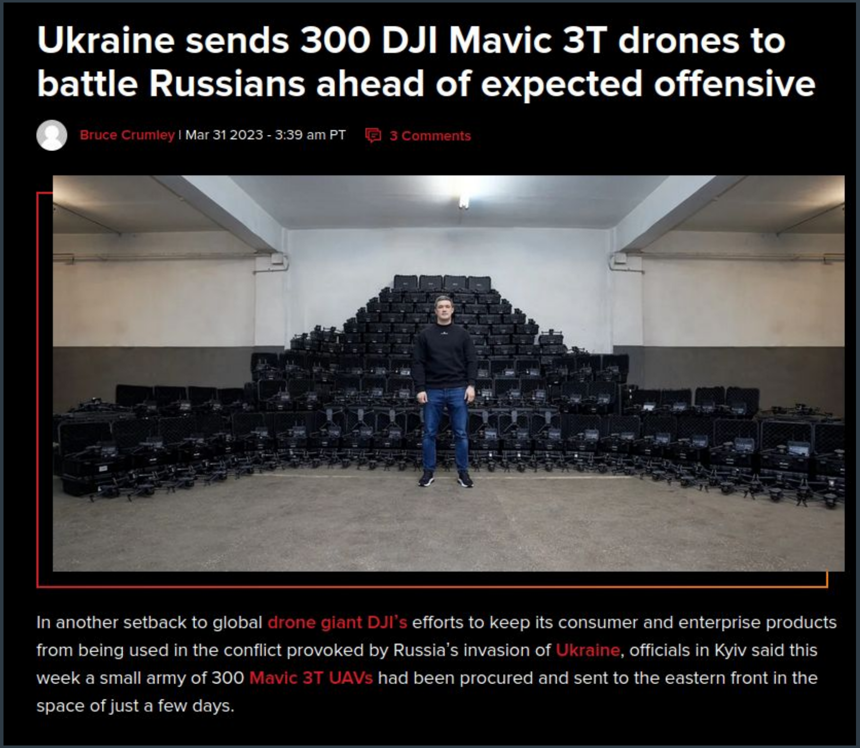


Ukraine sends 300 DJI Mavic 3T drones to battle Russians ahead of expected offensive

Bruce Crumley | Mar 31 2023 - 3:39 am PT 3 Comments

In another setback to global drone giant DJI's efforts to keep its consumer and enterprise products from being used in the conflict provoked by Russia's invasion of Ukraine, officials in Kyiv said this week a small army of 300 Mavic 3T UAVs had been procured and sent to the eastern front in the space of just a few days.

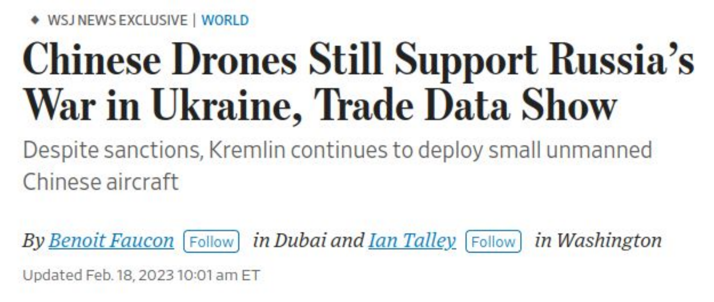
Recent Scenario: Conflicts



Ukraine sends 300 DJI Mavic 3T drones to battle Russians ahead of expected offensive

Bruce Crumley | Mar 31 2023 - 3:39 am PT 3 Comments

In another setback to global drone giant DJI's efforts to keep its consumer and enterprise products from being used in the conflict provoked by Russia's invasion of Ukraine, officials in Kyiv said this week a small army of 300 Mavic 3T UAVs had been procured and sent to the eastern front in the space of just a few days.



WSJ NEWS EXCLUSIVE | WORLD

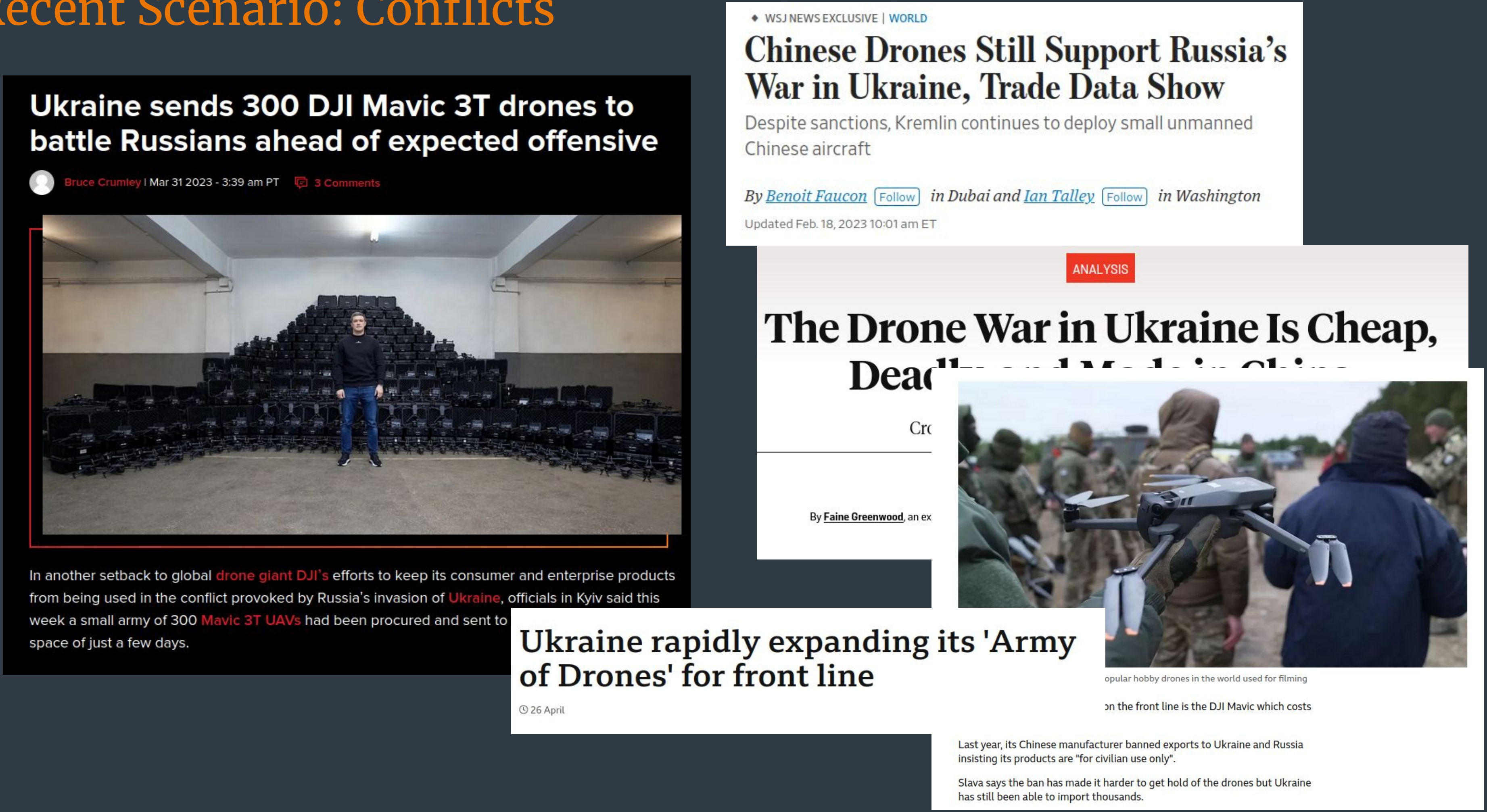
Chinese Drones Still Support Russia's War in Ukraine, Trade Data Show

Despite sanctions, Kremlin continues to deploy small unmanned Chinese aircraft

By Benoit Faucon Follow *in Dubai and Ian Talley* Follow *in Washington*

Updated Feb. 18, 2023 10:01 am ET

Recent Scenario: Conflicts



Ukraine sends 300 DJI Mavic 3T drones to battle Russians ahead of expected offensive

Bruce Crumley | Mar 31 2023 - 3:39 am PT 3 Comments

In another setback to global drone giant DJI's efforts to keep its consumer and enterprise products from being used in the conflict provoked by Russia's invasion of Ukraine, officials in Kyiv said this week a small army of 300 Mavic 3T UAVs had been procured and sent to

space of just a few days.

◆ WSJ NEWS EXCLUSIVE | WORLD

Chinese Drones Still Support Russia's War in Ukraine, Trade Data Show

Despite sanctions, Kremlin continues to deploy small unmanned Chinese aircraft

By Benoit Faucon Follow *in Dubai and Ian Talley* Follow *in Washington*

Updated Feb. 18, 2023 10:01 am ET

ANALYSIS

The Drone War in Ukraine Is Cheap, Dea

Cr

By Faine Greenwood, an ex

Ukraine rapidly expanding its 'Army of Drones' for front line

26 April

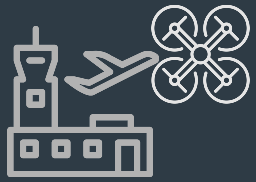
opular hobby drones in the world used for filming

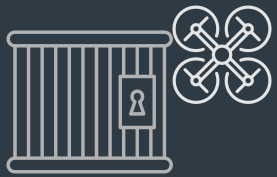
on the front line is the DJI Mavic which costs

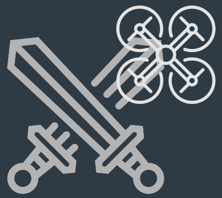
Last year, its Chinese manufacturer banned exports to Ukraine and Russia insisting its products are "for civilian use only".

Slava says the ban has made it harder to get hold of the drones but Ukraine has still been able to import thousands.

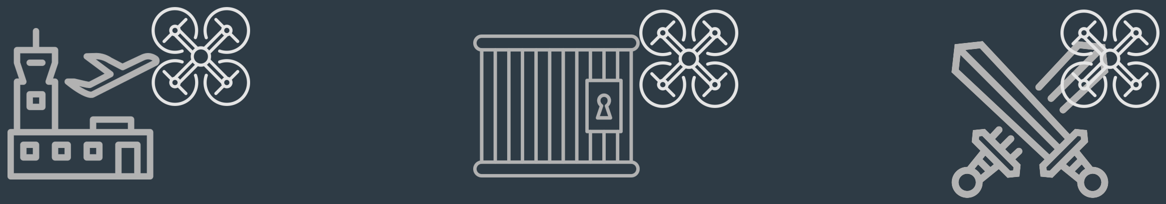
Vendors know these problems!





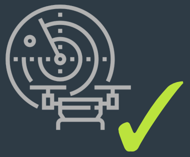


Vendors know these problems!

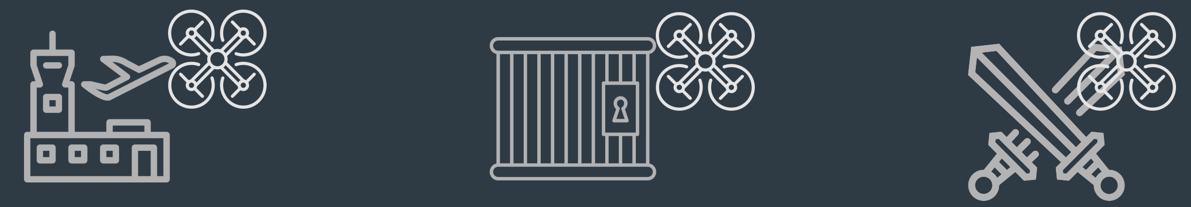


Position tracking

DJI Aeroscope



Vendors know these problems!



Position tracking ✓

DJI Aeroscope

Software limits ✓

Geofencing

Vendors know these problems!

Position tracking ✓

DJI Aeroscope

Software limits ✓

Geofencing

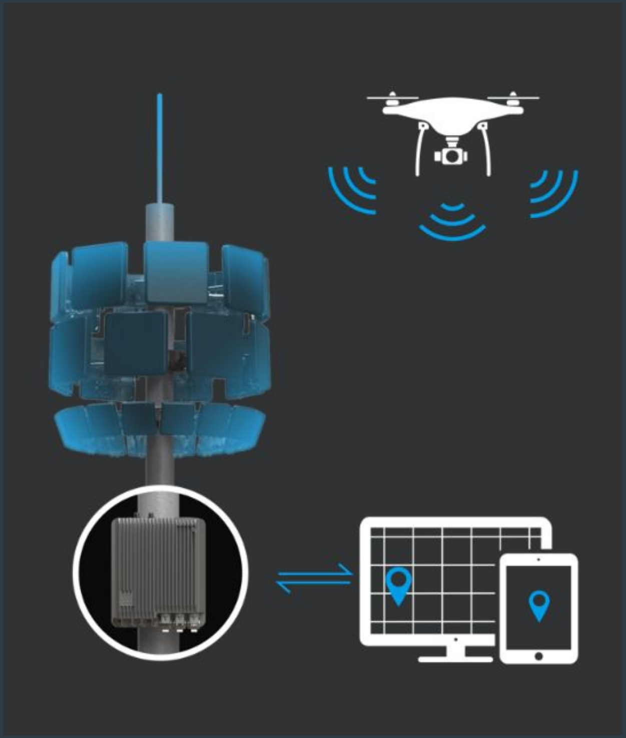
Hardware protection ✓

No debug interfaces

Tracking and Identification

- Drones broadcast information
  - Serial number
  - Position
- Tracking via DJI Aeroscope *(recently deprecated)*
- New regulations mandate tracking

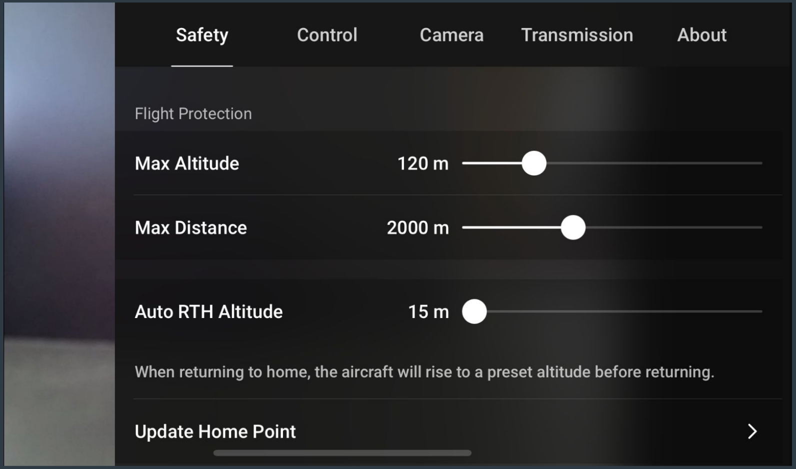
=> Quick identification and localization!



https://www.dji.com/de/aeroscope

Software Protection

- Height and range limitations
  - height: maximum 500m
  - but: safety warning above 120m
  - range: currently unlimited
- Speed limits
- No-Fly Zones



Flight Protection

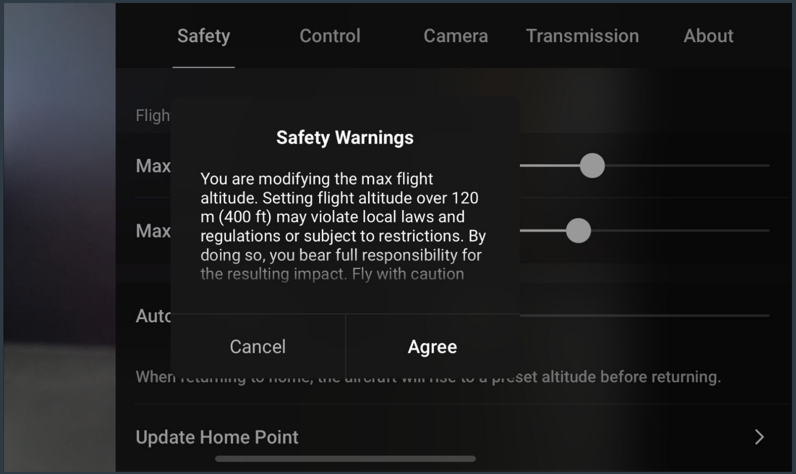
Max Altitude 120 m

Max Distance 2000 m

Auto RTH Altitude 15 m

When returning to home, the aircraft will rise to a preset altitude before returning.

Update Home Point



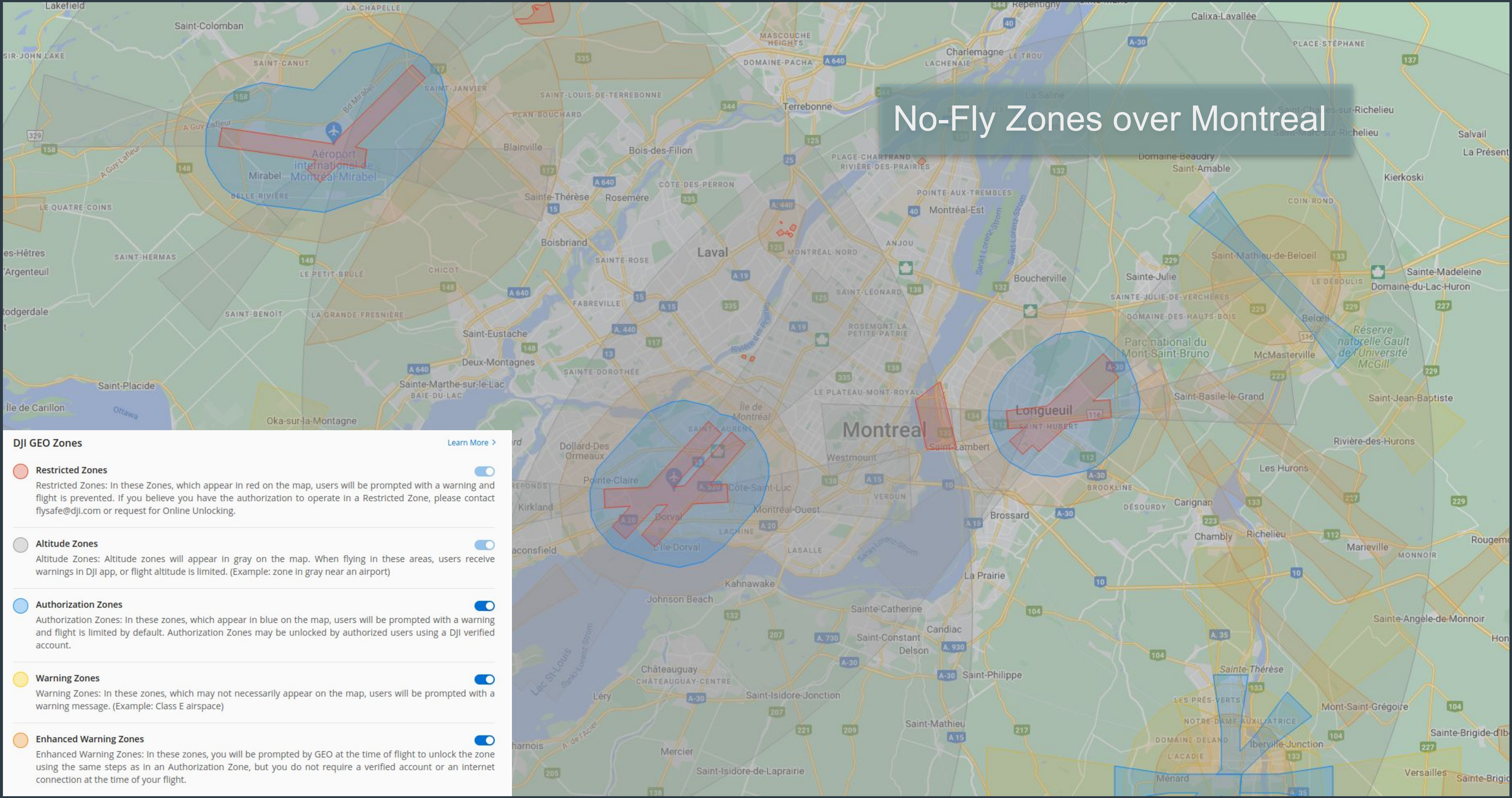
Safety Warnings

You are modifying the max flight altitude. Setting flight altitude over 120 m (400 ft) may violate local laws and regulations or subject to restrictions. By doing so, you bear full responsibility for the resulting impact. Fly with caution

Cancel Agree

When returning to home, the aircraft will rise to a preset altitude before returning.

No-Fly Zones over Montreal



DJI GEO Zones

Learn More >

Restricted Zones

Restricted Zones: In these Zones, which appear in red on the map, users will be prompted with a warning and flight is prevented. If you believe you have the authorization to operate in a Restricted Zone, please contact flysafe@dji.com or request for Online Unlocking.

Altitude Zones

Altitude Zones: Altitude zones will appear in gray on the map. When flying in these areas, users receive warnings in DJI app, or flight altitude is limited. (Example: zone in gray near an airport)

Authorization Zones

Authorization Zones: In these zones, which appear in blue on the map, users will be prompted with a warning and flight is limited by default. Authorization Zones may be unlocked by authorized users using a DJI verified account.

Warning Zones

Warning Zones: In these zones, which may not necessarily appear on the map, users will be prompted with a warning message. (Example: Class E airspace)

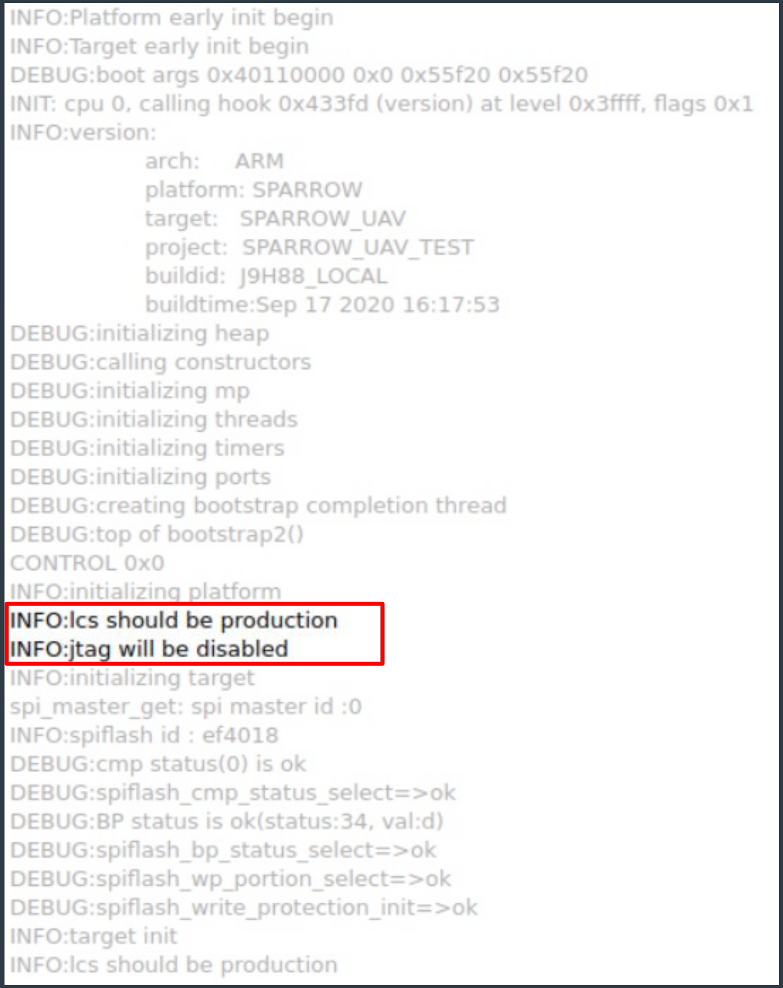
Enhanced Warning Zones

Enhanced Warning Zones: In these zones, you will be prompted by GEO at the time of flight to unlock the zone using the same steps as in an Authorization Zone, but you do not require a verified account or an internet connection at the time of your flight.

https://fly-safe.dji.com/nfz/nfz-query

Hardware Protection

- disabled debug interfaces
- firmware
  - closed source
  - encrypted
  - signed
- proprietary communication protocol



```
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x55f20 0x55f20
INIT: cpu 0, calling hook 0x433fd (version) at level 0x3ffff, flags 0x1
INFO:version:
                arch:     ARM
                platform: SPARROW
                target:   SPARROW_UAV
                project:  SPARROW_UAV_TEST
                buildid:  J9H88_LOCAL
                buildtime:Sep 17 2020 16:17:53
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
INFO:lcs should be production
```

Vendors know these problems!

Position tracking ✓
DJI Aeroscope

Software limits ✓
Geofencing

Hardware protection ✓
No debug interfaces

Let’s see if these countermeasures are good enough

Our focus: DJI drones

Photo: Copyright dji.com

Our focus: DJI drones

- Market share (94% Consumer)

Photo: Copyright dji.com

Our focus: DJI drones

- Market share (94% Consumer)
- Security-conscious
  - Whitepaper
  - Bug bounty program

Photo: Copyright dji.com

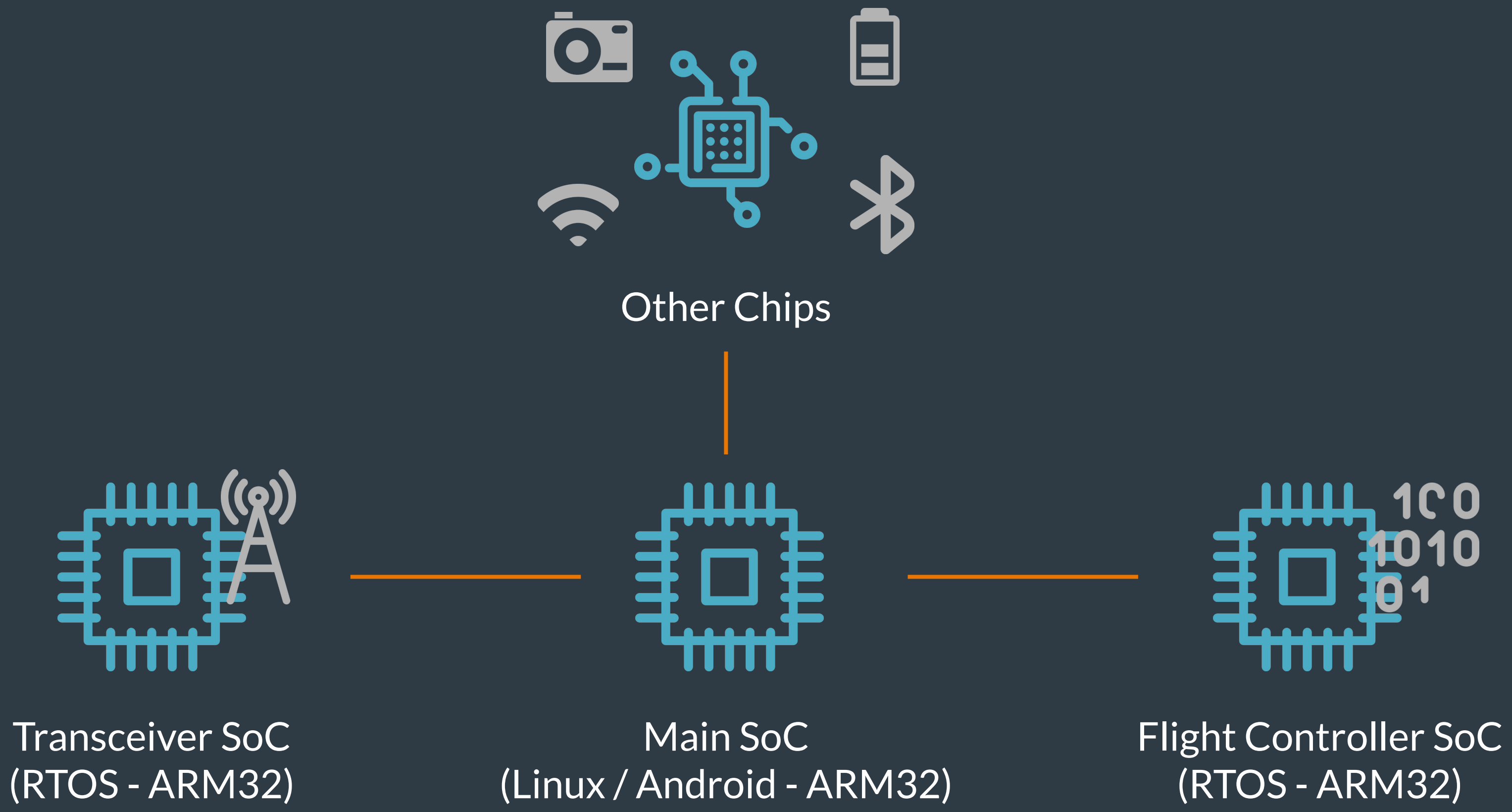
Our focus: DJI drones

- Market share (94% Consumer)
- Security-conscious
  - Whitepaper
  - Bug bounty program

Analyzed Drones: Mini 2, Mavic Air 2, Mavic 2

Photo: Copyright dji.com

Drone Hardware Overview



Other Chips

Transceiver SoC
(RTOS - ARM32)

Main SoC
(Linux / Android - ARM32)

Flight Controller SoC
(RTOS - ARM32)

Wireless Physical Layer

- Eavesdropping
- Signal analysis

- Tracking
- Protocol knowledge

Wireless Physical Layer

- Eavesdropping
- Signal analysis

- Tracking
- Protocol knowledge

Hardware

- PCB analysis
- Component lookup

- Debug interfaces
- Firmware dumping
- Memory dumping

Wireless Physical Layer

- Eavesdropping
- Signal analysis

- Tracking
- Protocol knowledge

Hardware

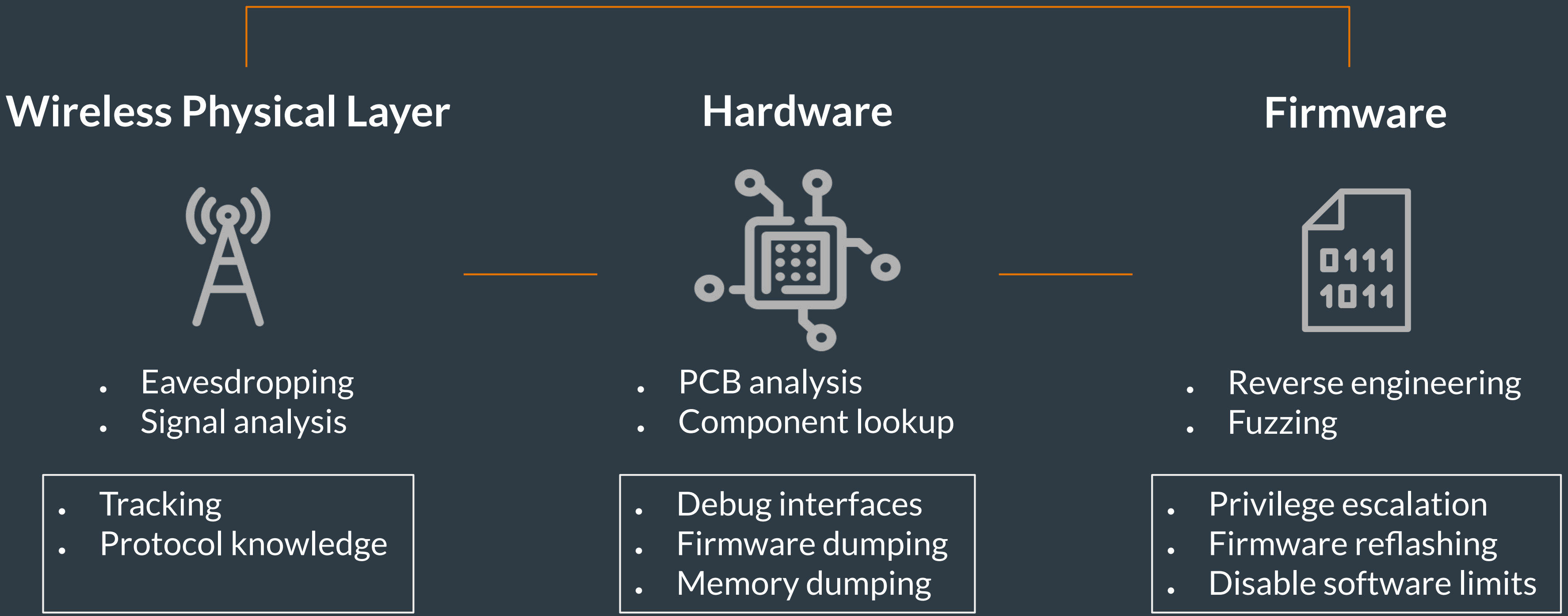
- PCB analysis
- Component lookup

- Debug interfaces
- Firmware dumping
- Memory dumping

Firmware

- Reverse engineering
- Fuzzing

- Privilege escalation
- Firmware reflashing
- Disable software limits



Wireless Physical Layer

- Eavesdropping
- Signal analysis

- Tracking
- Protocol knowledge

Hardware

- PCB analysis
- Component lookup

- Debug interfaces
- Firmware dumping
- Memory dumping

Firmware

- Reverse engineering
- Fuzzing

- Privilege escalation
- Firmware reflashing
- Disable software limits

Wireless Physical Layer

Reversing DJI DroneID

Static Analysis

Hands on the Drone

Dynamic Analysis

Fuzzing Drones for Pain and Profit

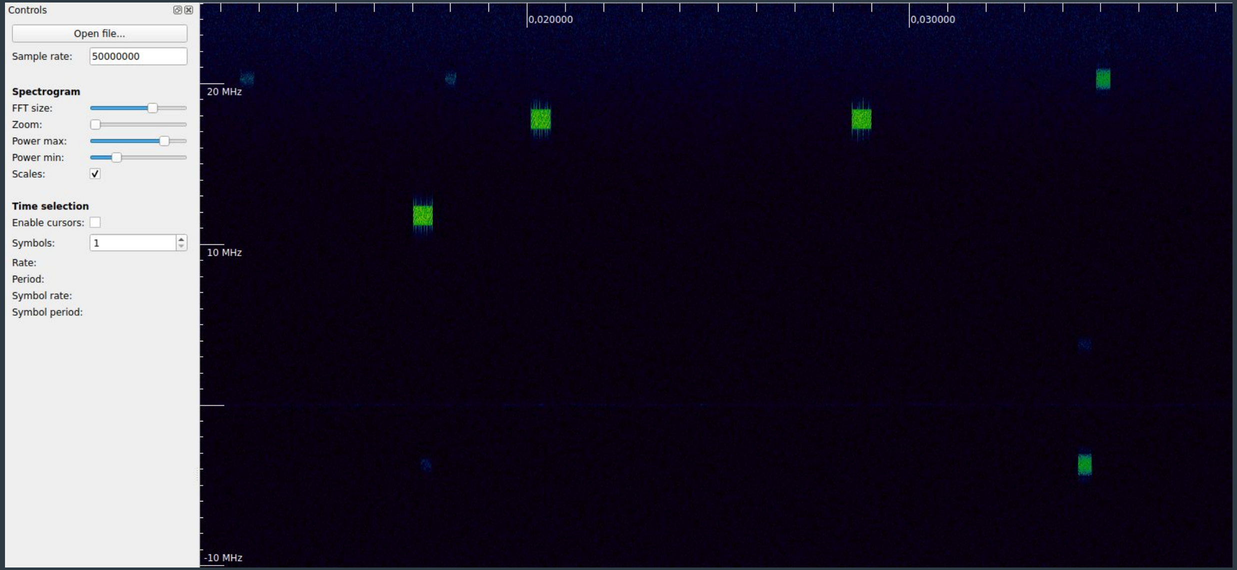
How to listen on the Wireless Physical Layer …

How to listen on the Wireless Physical Layer ...

Software Defined Radio
(SDR)

How to listen on the Wireless Physical Layer ...

Software Defined Radio
(SDR)


Controls

Open file...

Sample rate: 50000000

Spectrogram

FFT size:

Zoom:

Power max:

Power min:

Scales: ☑

Time selection

Enable cursors: □

Symbols: 1

Rate:

Period:

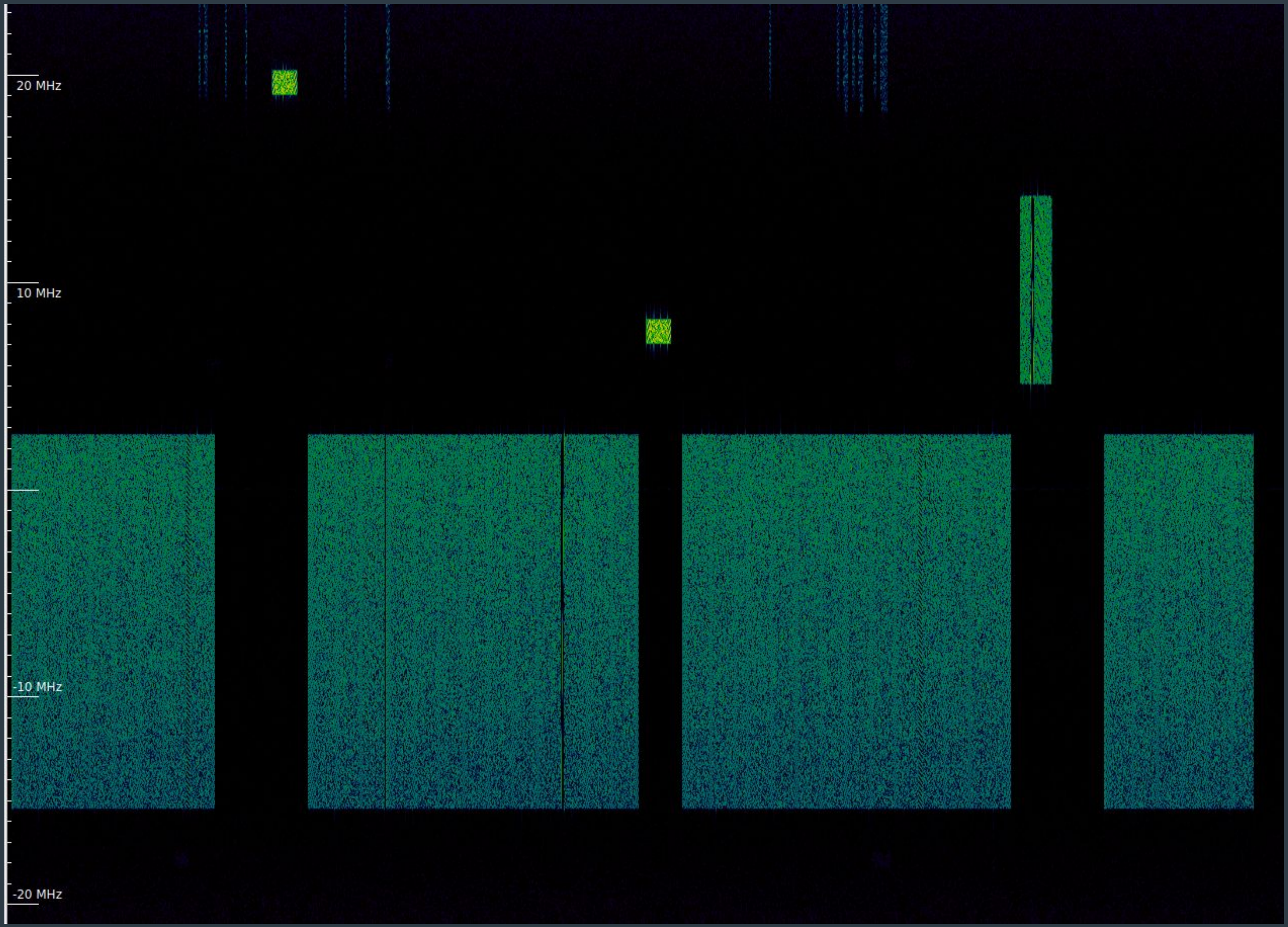
Symbol rate:

Symbol period:

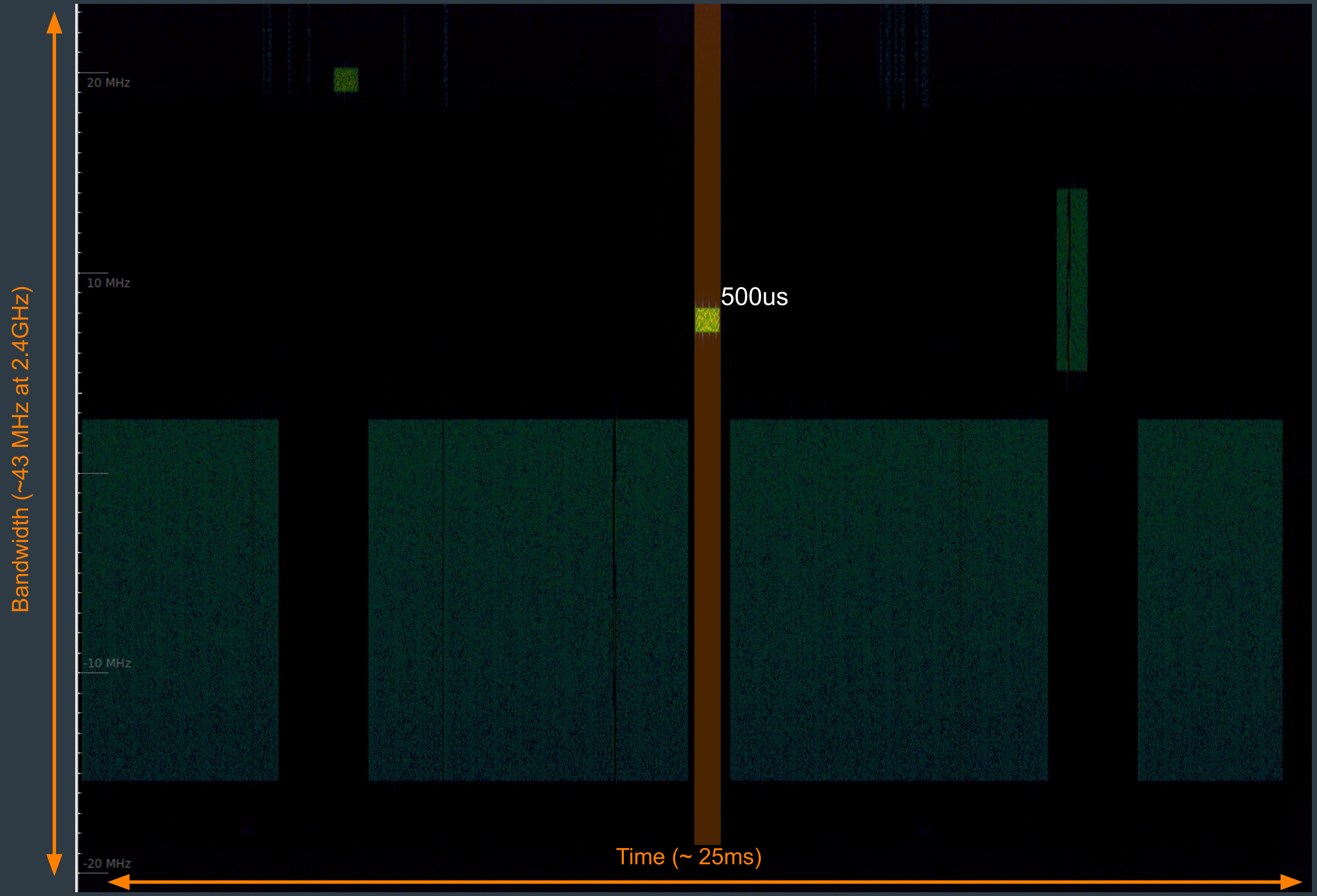
Signal Analyzer Software
(e.g., baudline, inspectrum)

Listening on the Wireless Physical Layer …

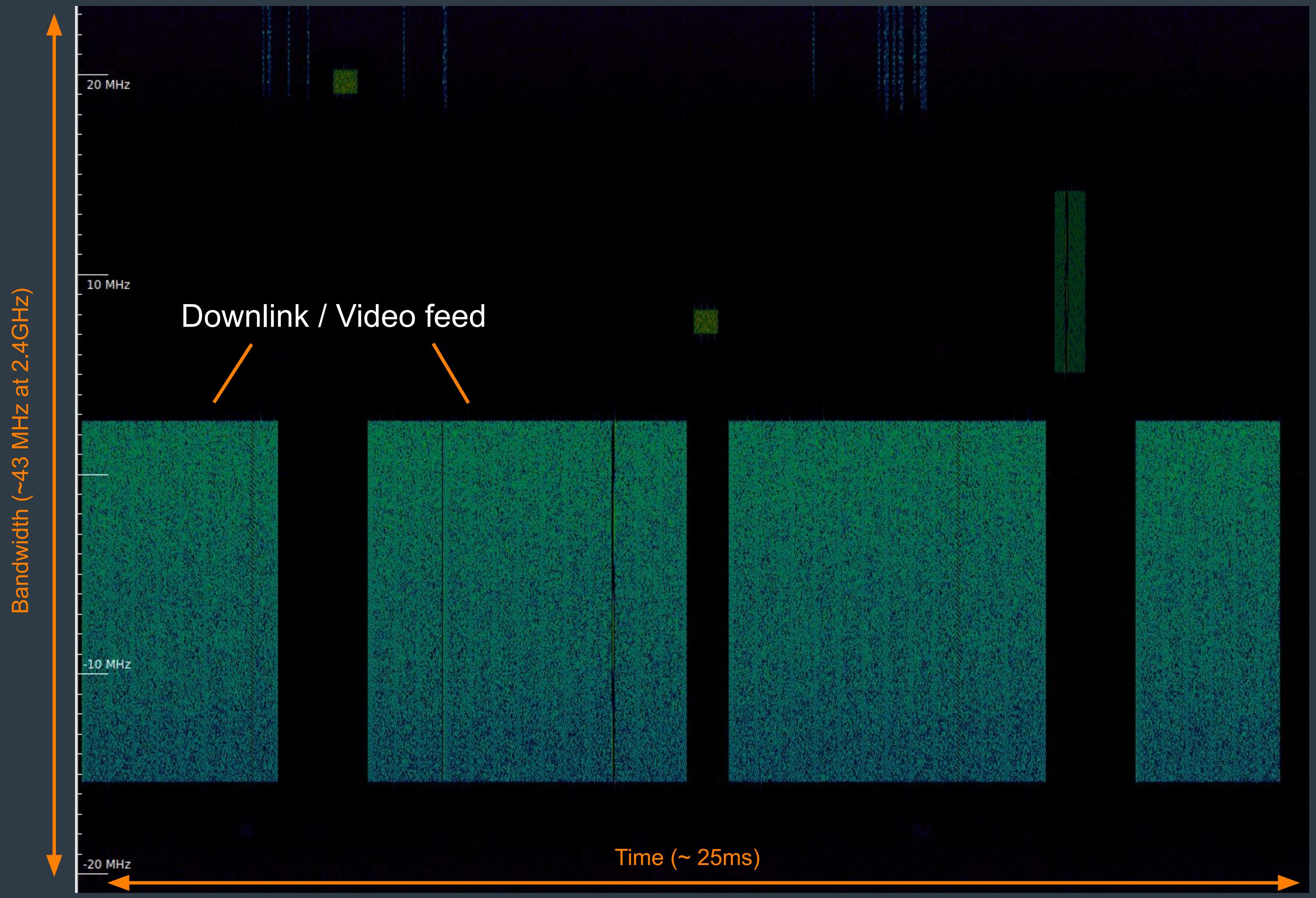
Listening on the Wireless Physical Layer ...



Listening on the Wireless Physical Layer ...

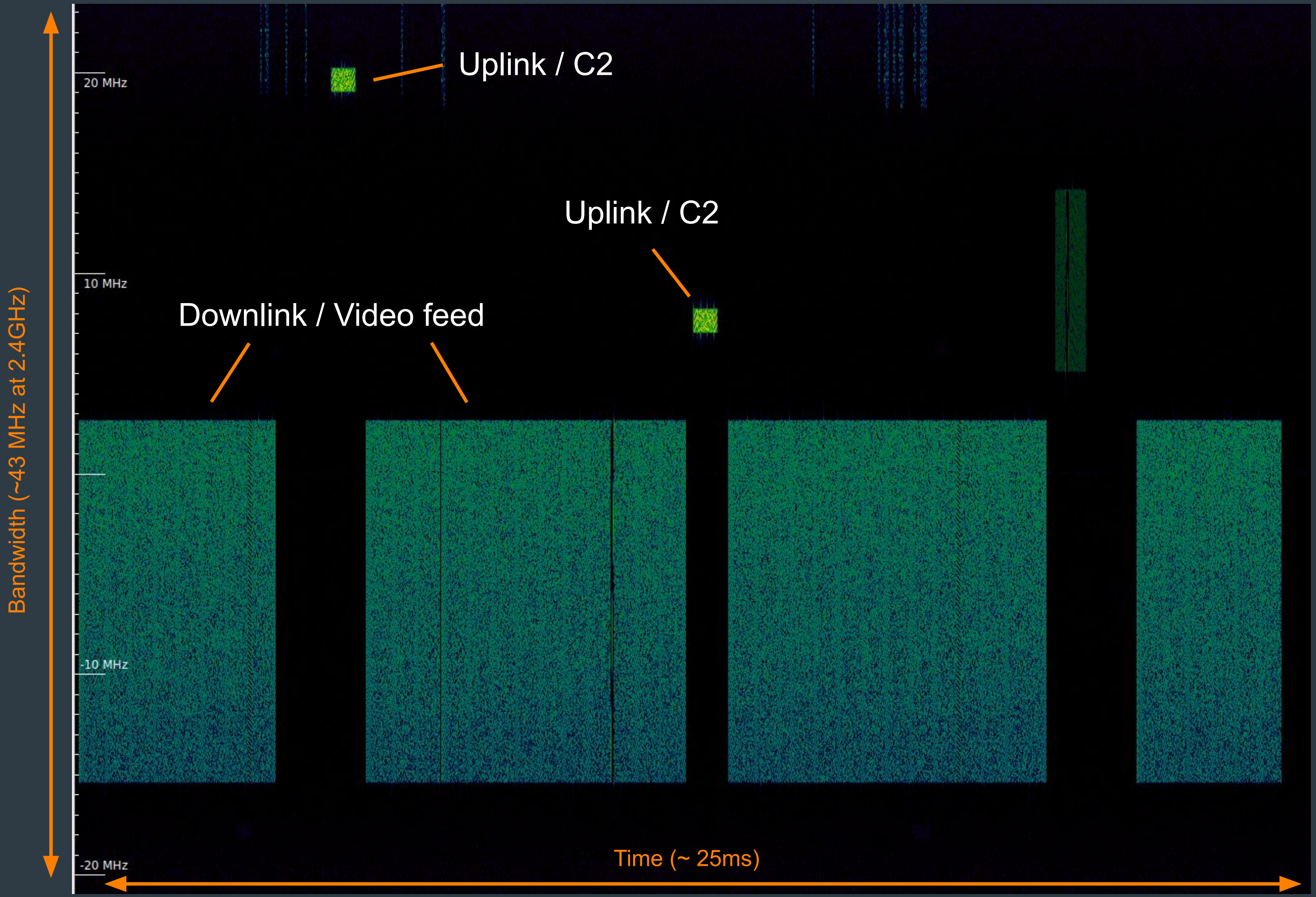


Listening on the Wireless Physical Layer ...

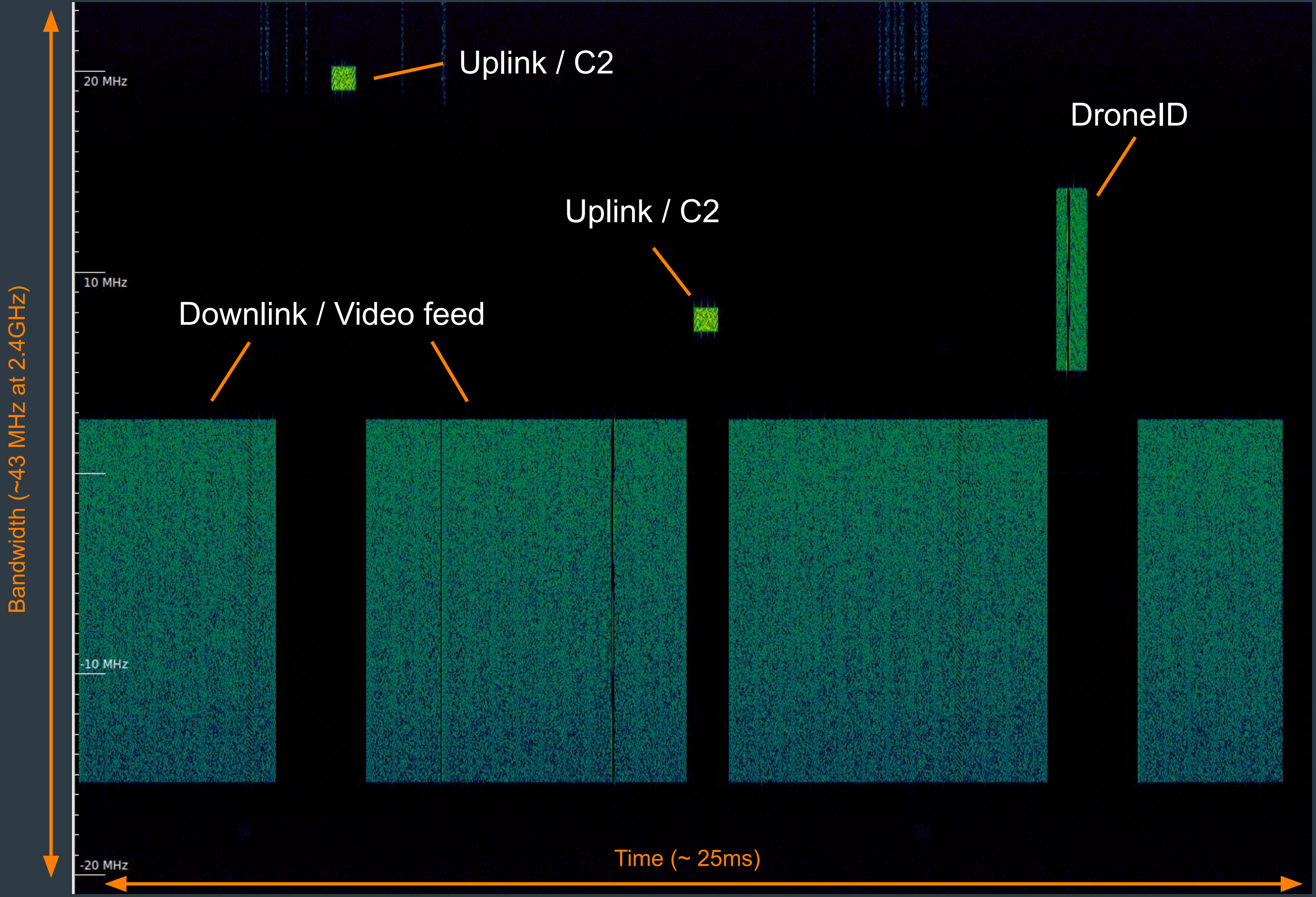


Downlink / Video feed

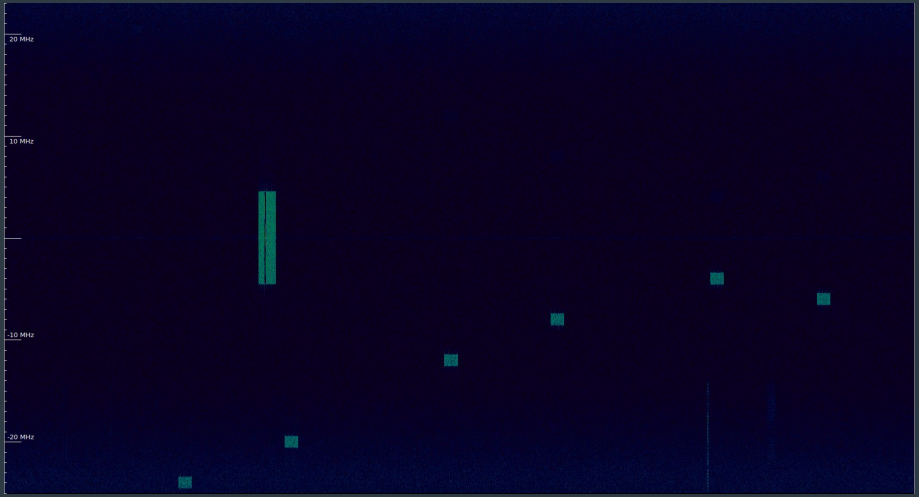
Listening on the Wireless Physical Layer ...



Listening on the Wireless Physical Layer ...

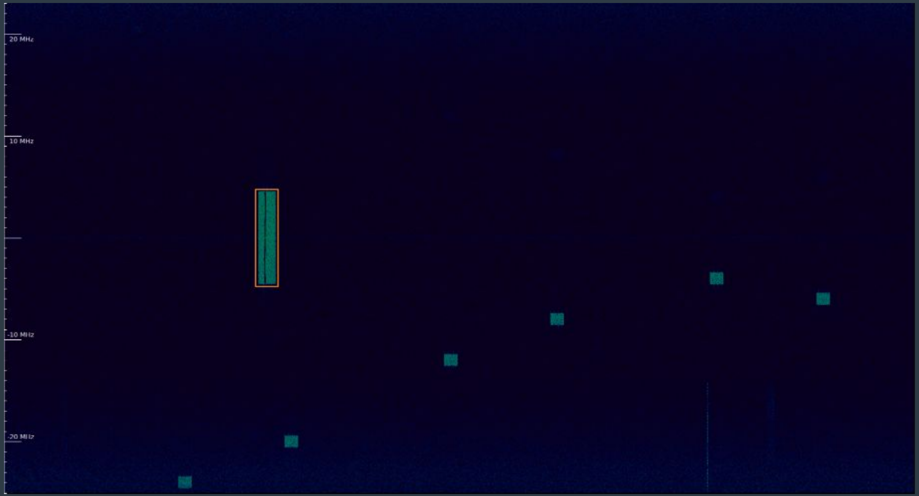


Reverse Engineering a Signal



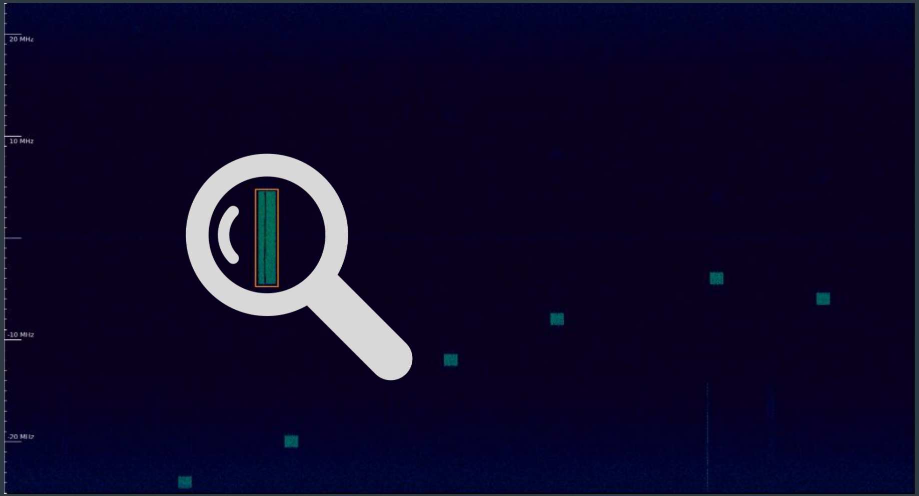
Capture Raw
Signal Data

Reverse Engineering a Signal



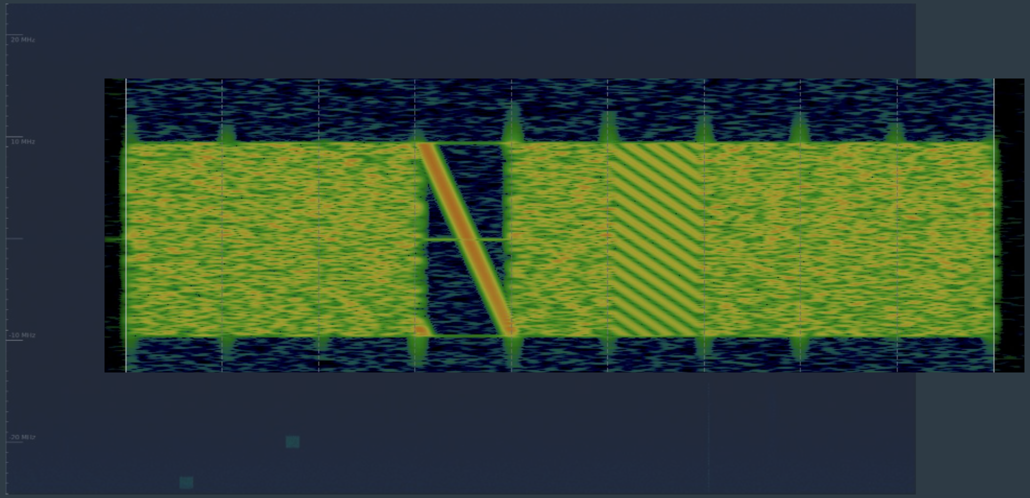
Capture Raw Signal Data → Packet Detection

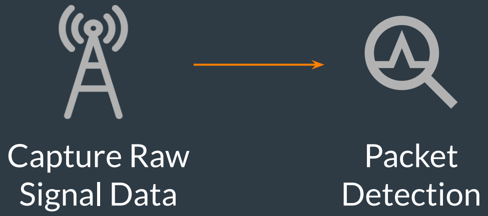
Reverse Engineering a Signal



Capture Raw Signal Data → Packet Detection

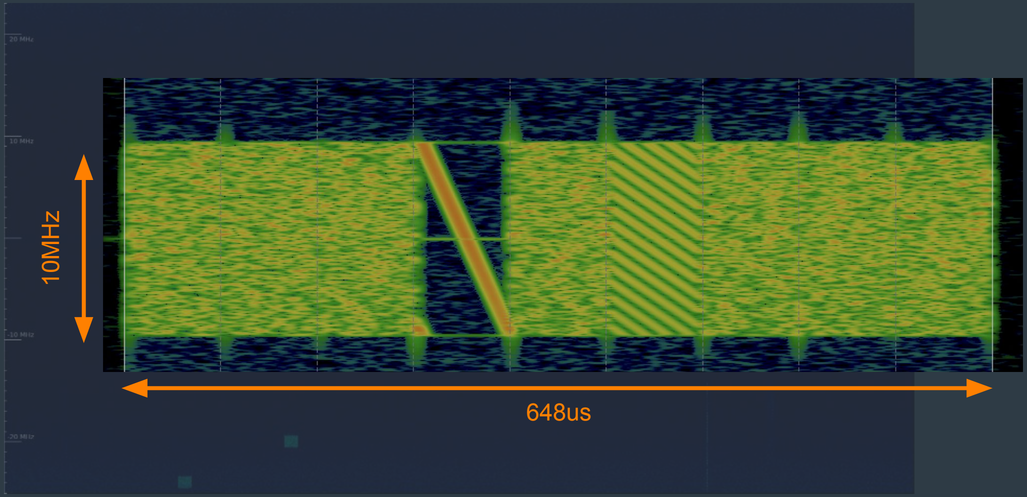
Reverse Engineering a Signal




Capture Raw Signal Data

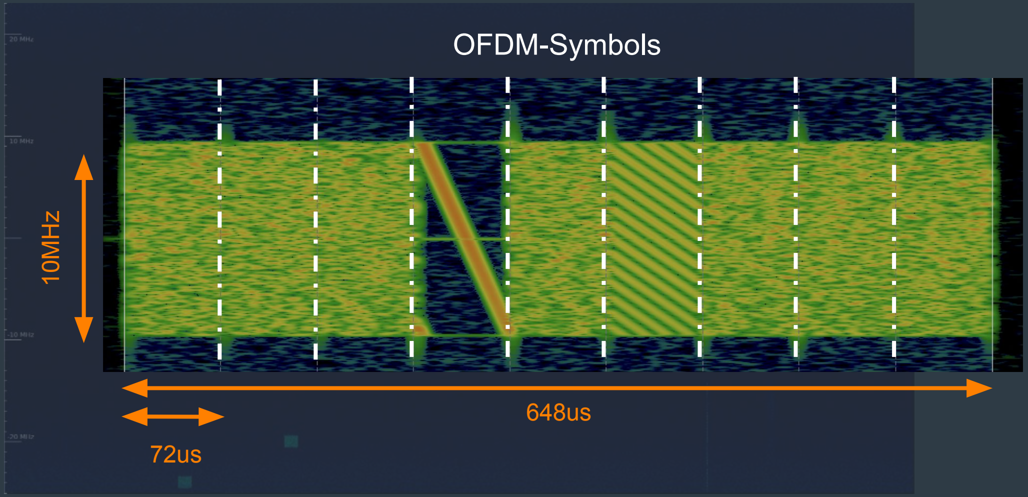
Packet Detection

Reverse Engineering a Signal



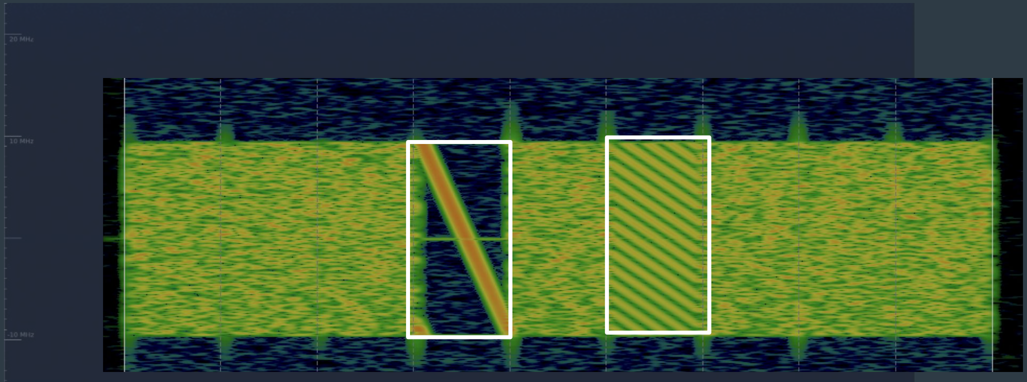
Capture Raw Signal Data → Packet Detection

Reverse Engineering a Signal

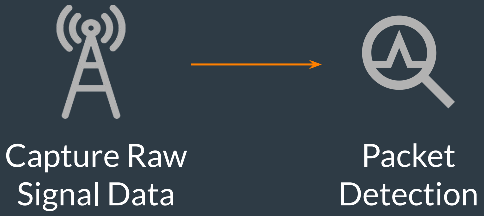


Capture Raw Signal Data → Packet Detection

Reverse Engineering a Signal

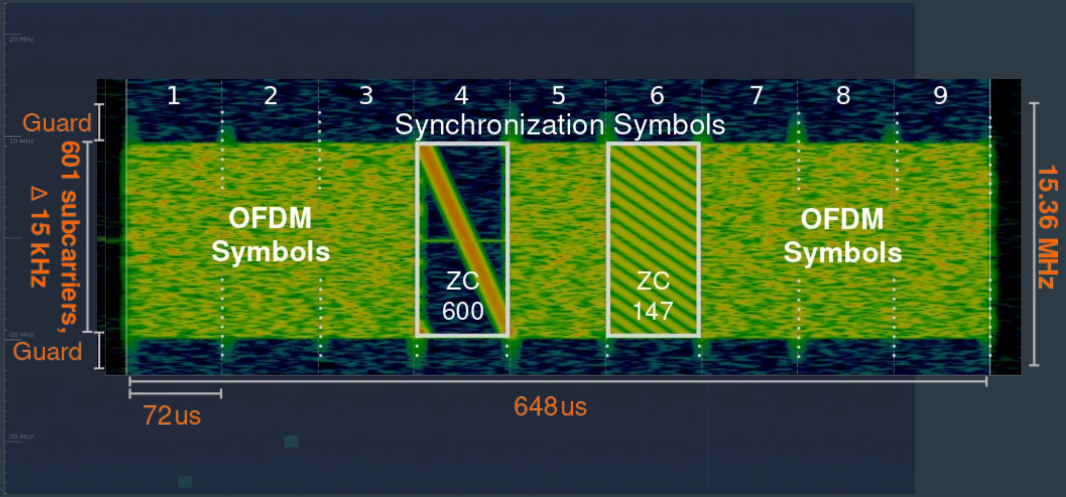


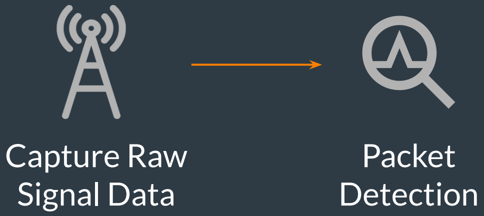
Fixed Synchronization Symbols
“Zadoff-Chu Sequences”


Capture Raw Signal Data

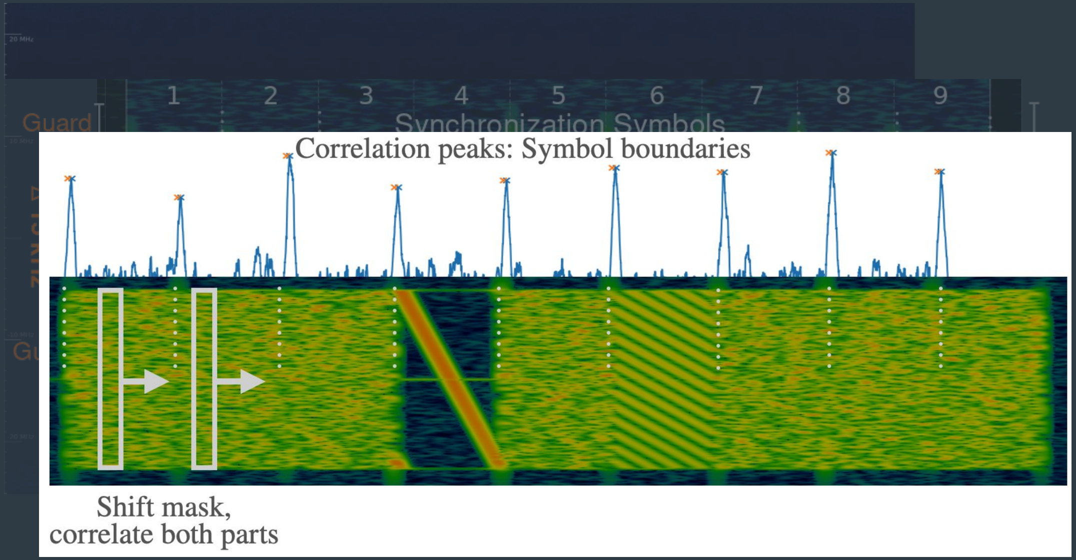
Packet Detection

Reverse Engineering a Signal

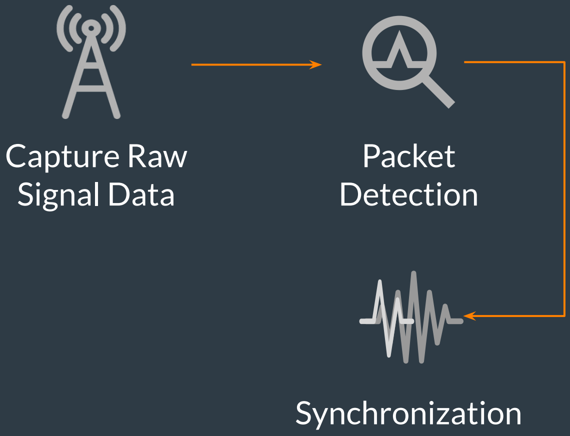




Reverse Engineering a Signal



Time synchronisation via cyclic prefixes

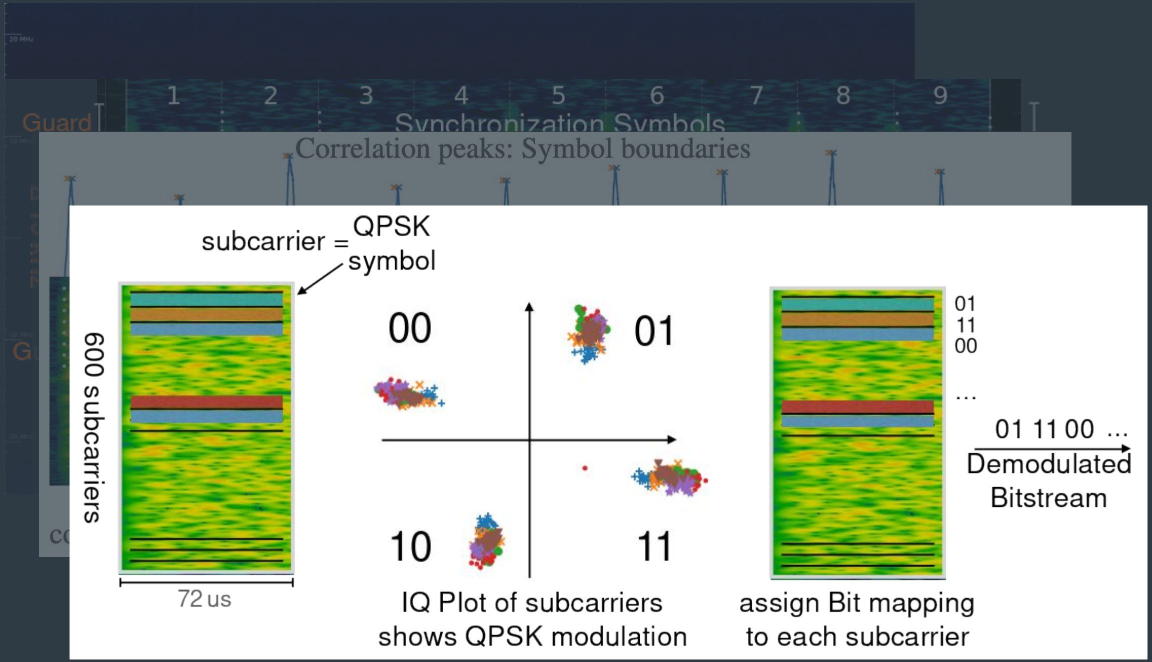


Capture Raw Signal Data

Packet Detection

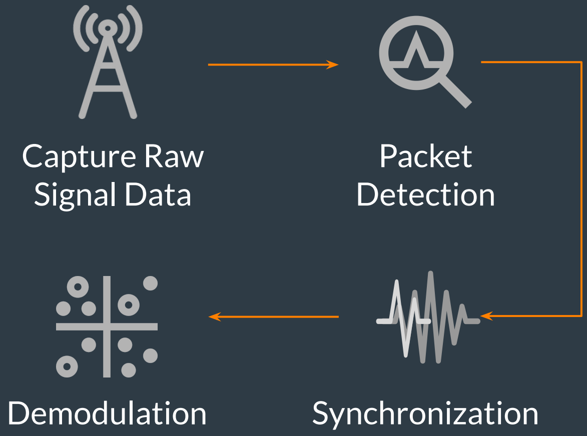
Synchronization

Reverse Engineering a Signal



IQ Plot of subcarriers shows QPSK modulation

assign Bit mapping to each subcarrier



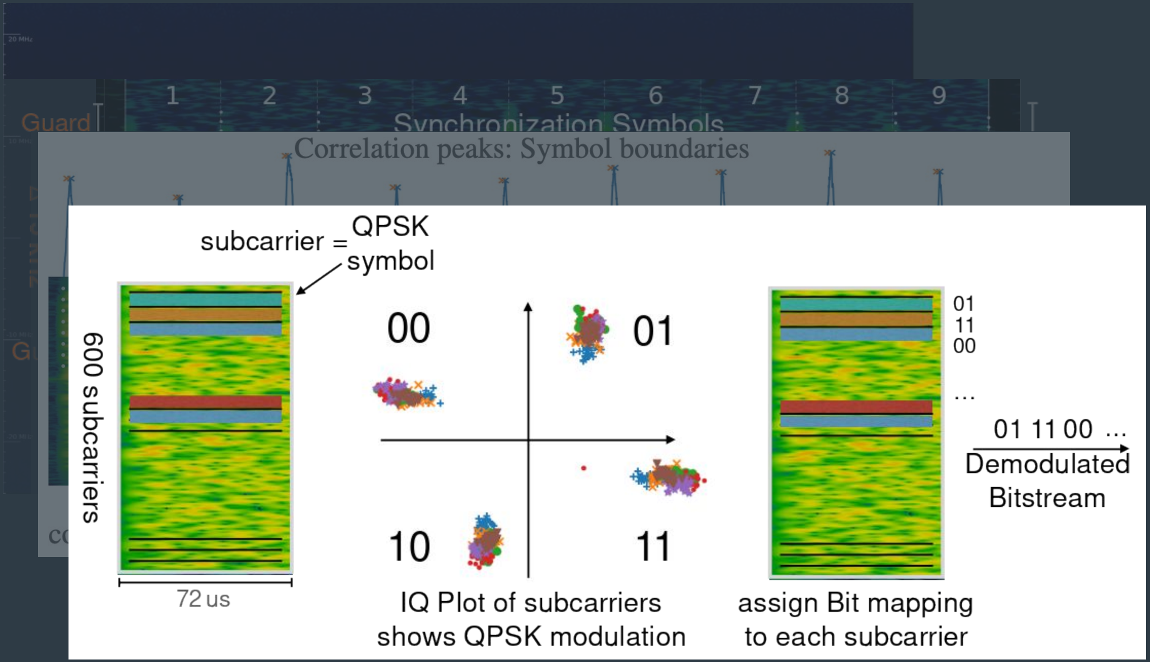
Capture Raw Signal Data

Packet Detection

Synchronization

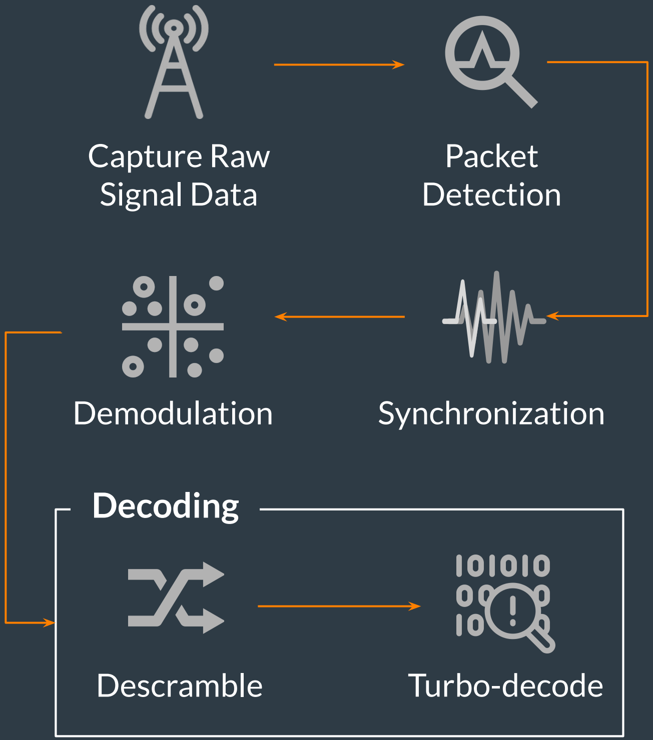
Demodulation

Reverse Engineering a Signal



IQ Plot of subcarriers shows QPSK modulation

assign Bit mapping to each subcarrier

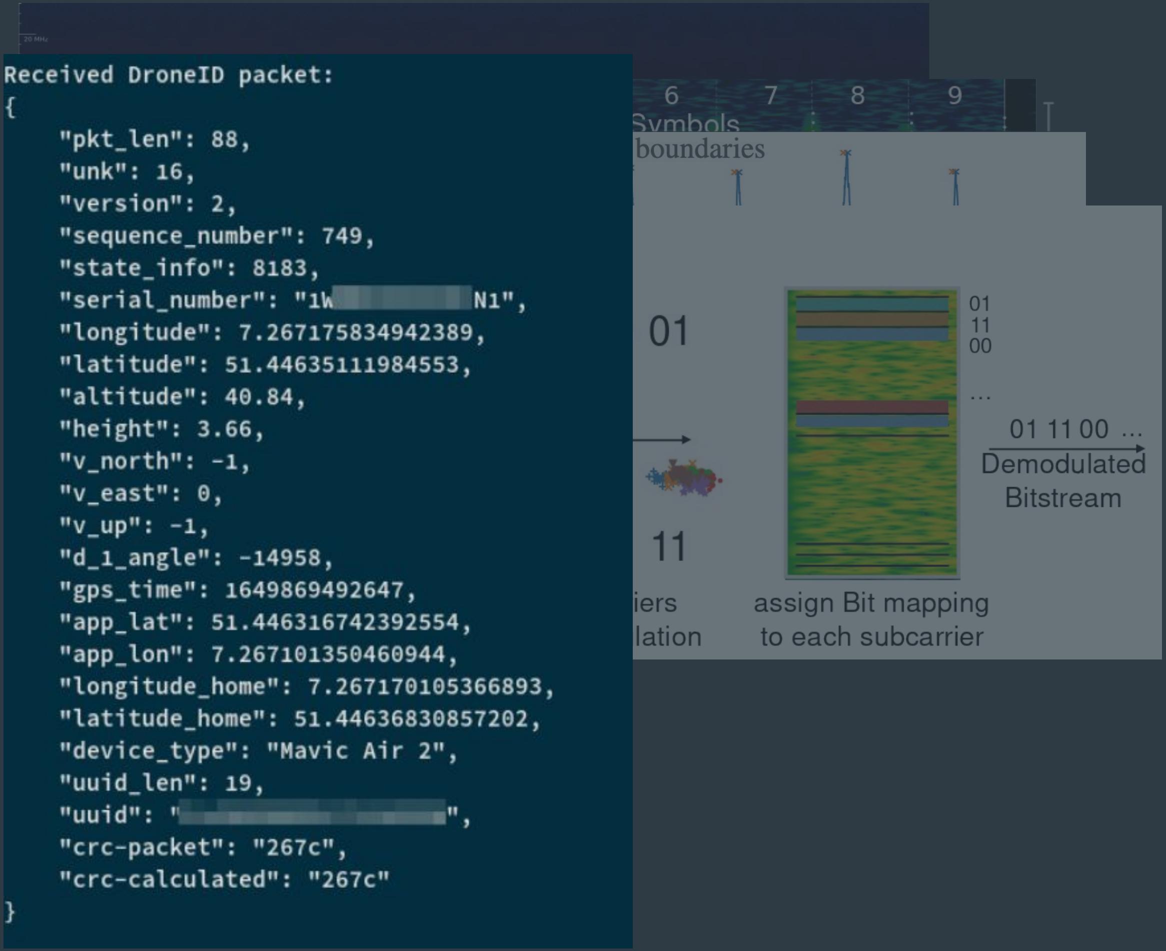


Capture Raw Signal Data → Packet Detection → Synchronization → Demodulation → Decoding

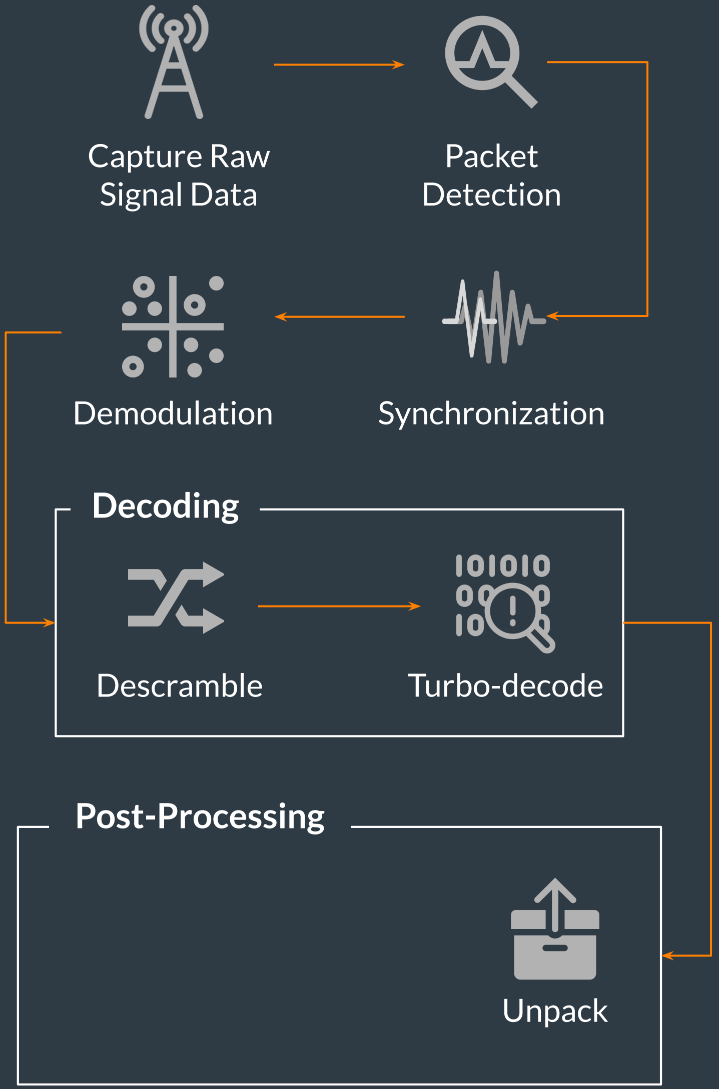
Decoding

Descramble → Turbo-decode

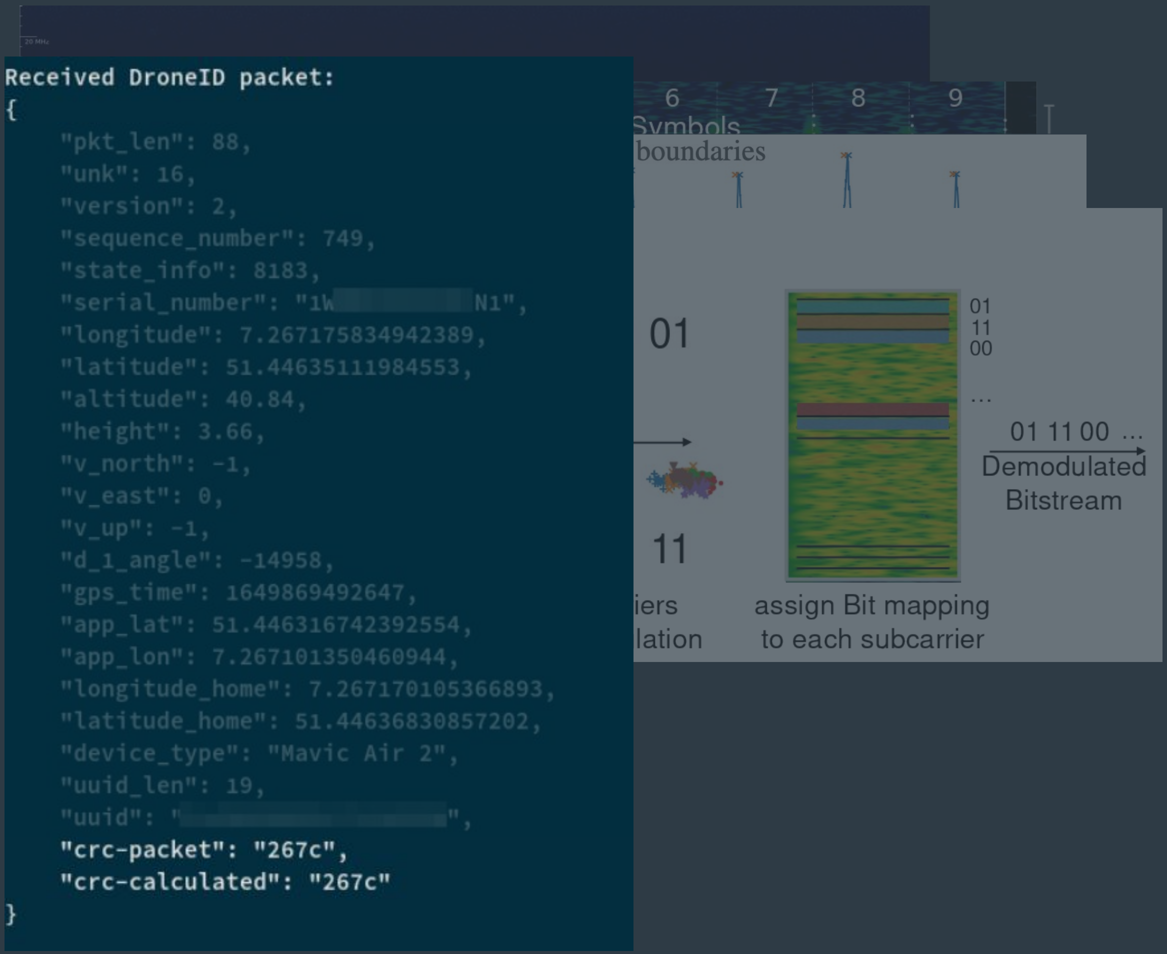
Reverse Engineering a Signal



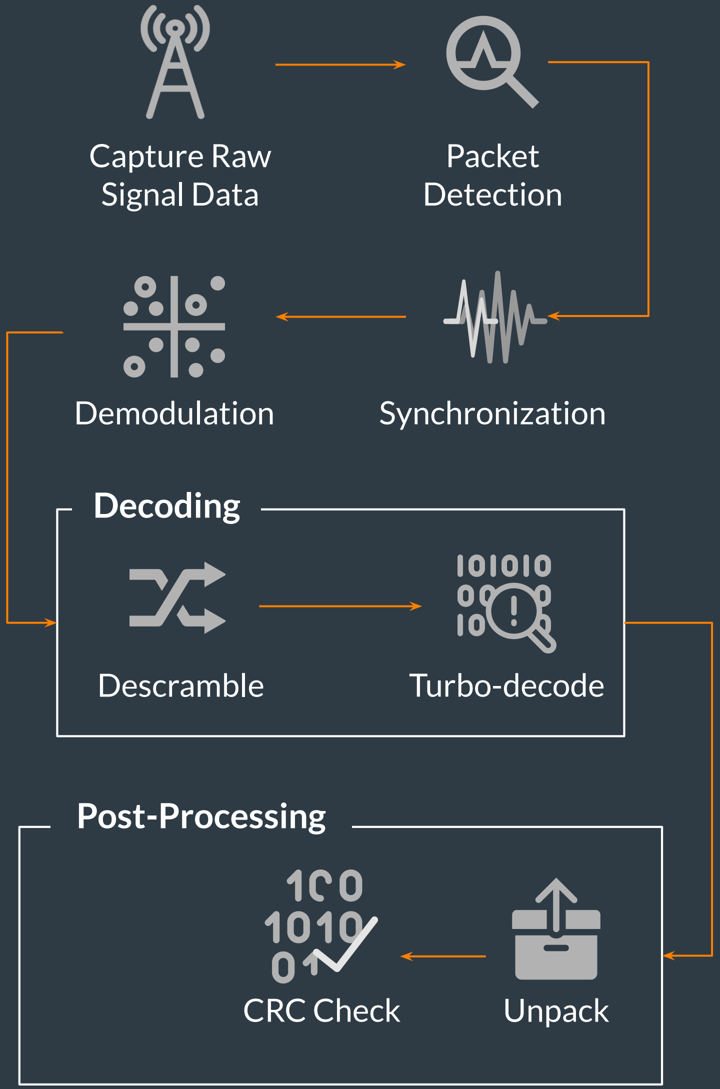
```
Received DroneID packet:
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_number": 749,
    "state_info": 8183,
    "serial_number": "1W[redacted]N1",
    "longitude": 7.267175834942389,
    "latitude": 51.44635111984553,
    "altitude": 40.84,
    "height": 3.66,
    "v_north": -1,
    "v_east": 0,
    "v_up": -1,
    "d_1_angle": -14958,
    "gps_time": 1649869492647,
    "app_lat": 51.446316742392554,
    "app_lon": 7.267101350460944,
    "longitude_home": 7.267170105366893,
    "latitude_home": 51.44636830857202,
    "device_type": "Mavic Air 2",
    "uuid_len": 19,
    "uuid": "[redacted]",
    "crc-packet": "267c",
    "crc-calculated": "267c"
}
```



Reverse Engineering a Signal



```
Received DroneID packet:
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_number": 749,
    "state_info": 8183,
    "serial_number": "1W[redacted]N1",
    "longitude": 7.267175834942389,
    "latitude": 51.44635111984553,
    "altitude": 40.84,
    "height": 3.66,
    "v_north": -1,
    "v_east": 0,
    "v_up": -1,
    "d_1_angle": -14958,
    "gps_time": 1649869492647,
    "app_lat": 51.446316742392554,
    "app_lon": 7.267101350460944,
    "longitude_home": 7.267170105366893,
    "latitude_home": 51.44636830857202,
    "device_type": "Mavic Air 2",
    "uuid_len": 19,
    "uuid": "[redacted]",
    "crc-packet": "267c",
    "crc-calculated": "267c"
}
```



Capture Raw Signal Data

Packet Detection

Synchronization

Demodulation

Decoding

Descramble

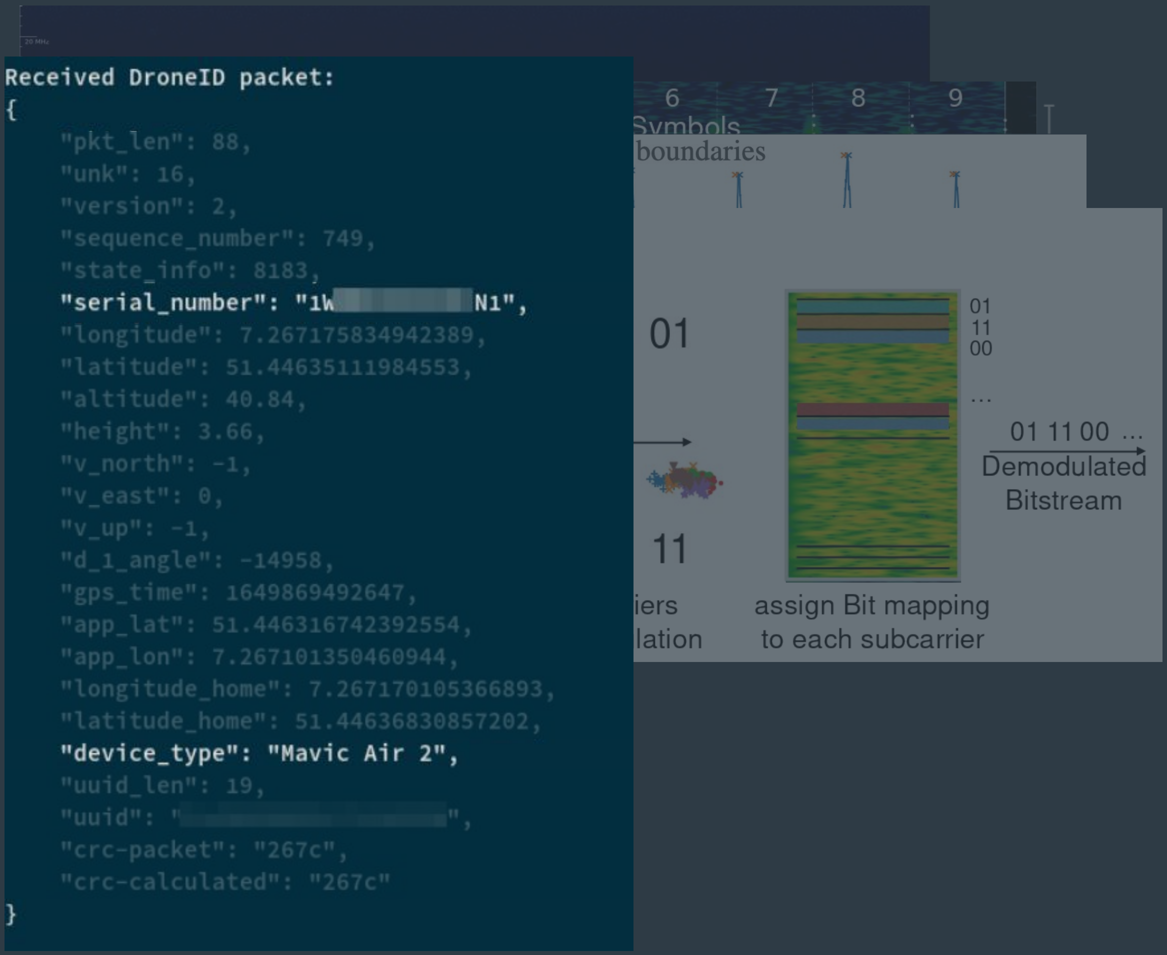
Turbo-decode

Post-Processing

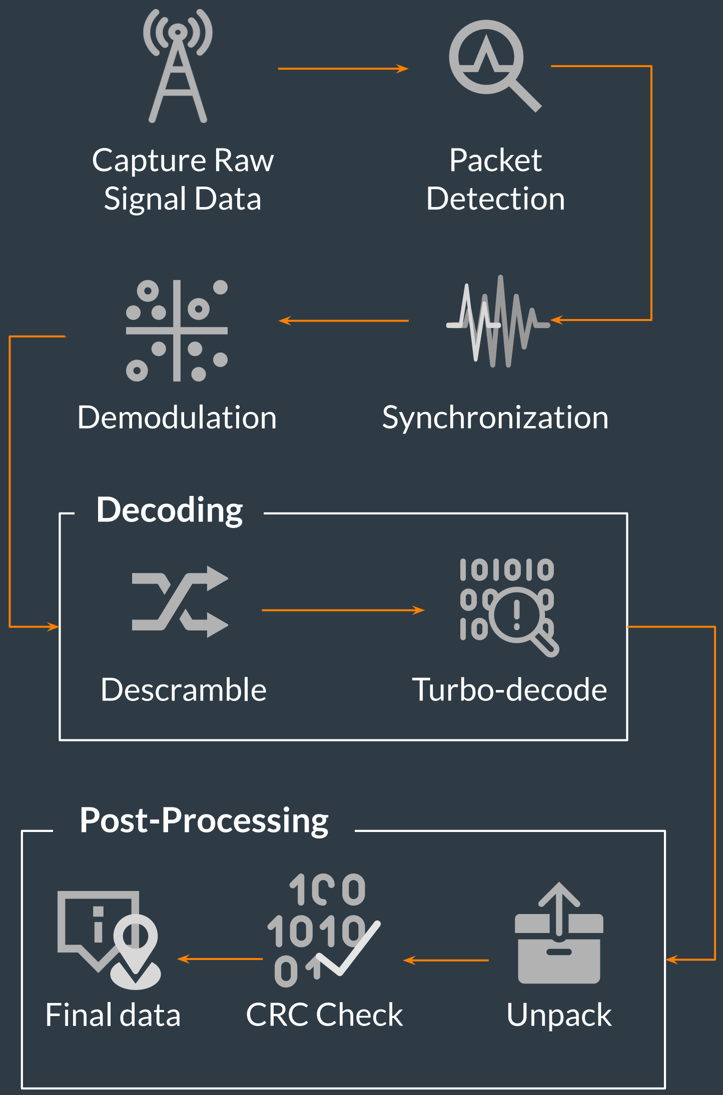
Unpack

CRC Check

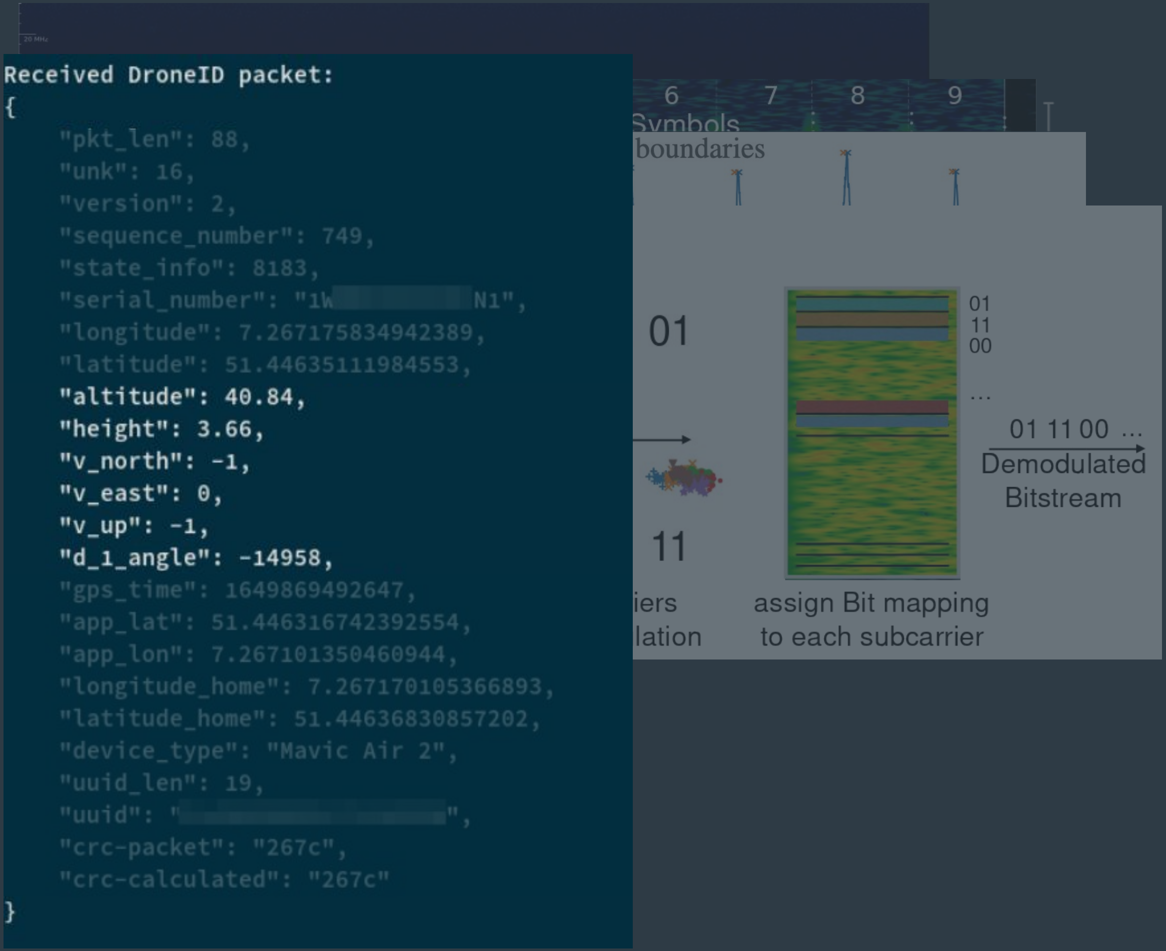
Reverse Engineering a Signal



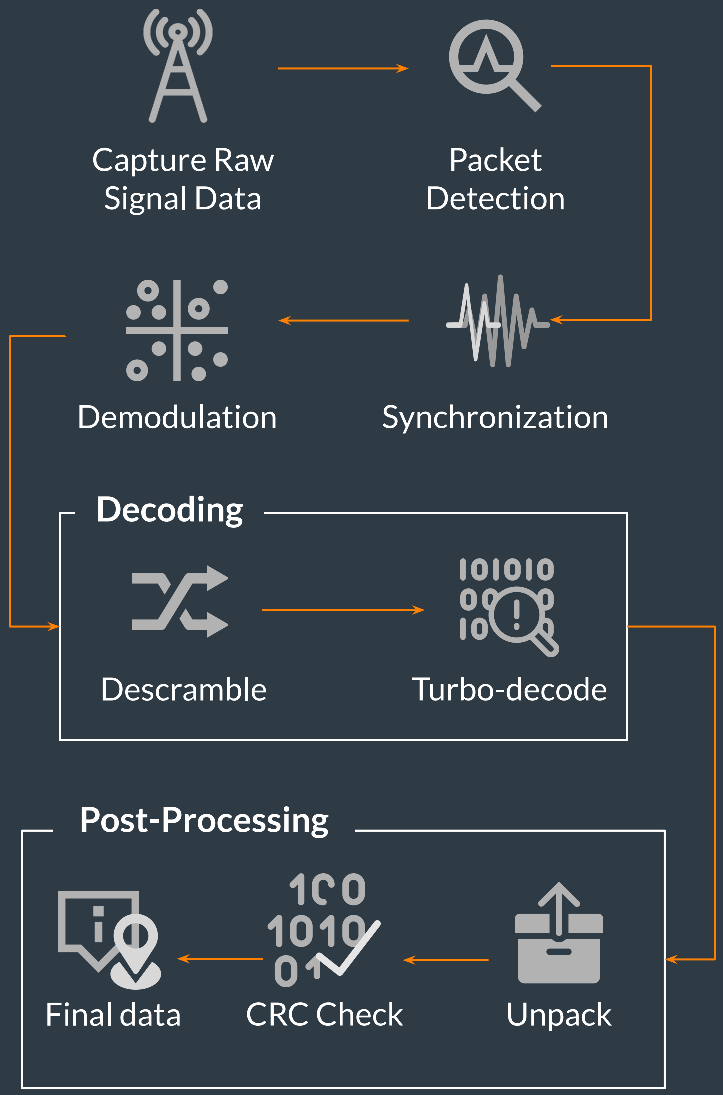
```
Received DroneID packet:
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_number": 749,
    "state_info": 8183,
    "serial_number": "1W[redacted]N1",
    "longitude": 7.267175834942389,
    "latitude": 51.44635111984553,
    "altitude": 40.84,
    "height": 3.66,
    "v_north": -1,
    "v_east": 0,
    "v_up": -1,
    "d_1_angle": -14958,
    "gps_time": 1649869492647,
    "app_lat": 51.446316742392554,
    "app_lon": 7.267101350460944,
    "longitude_home": 7.267170105366893,
    "latitude_home": 51.44636830857202,
    "device_type": "Mavic Air 2",
    "uuid_len": 19,
    "uuid": "[redacted]",
    "crc-packet": "267c",
    "crc-calculated": "267c"
}
```



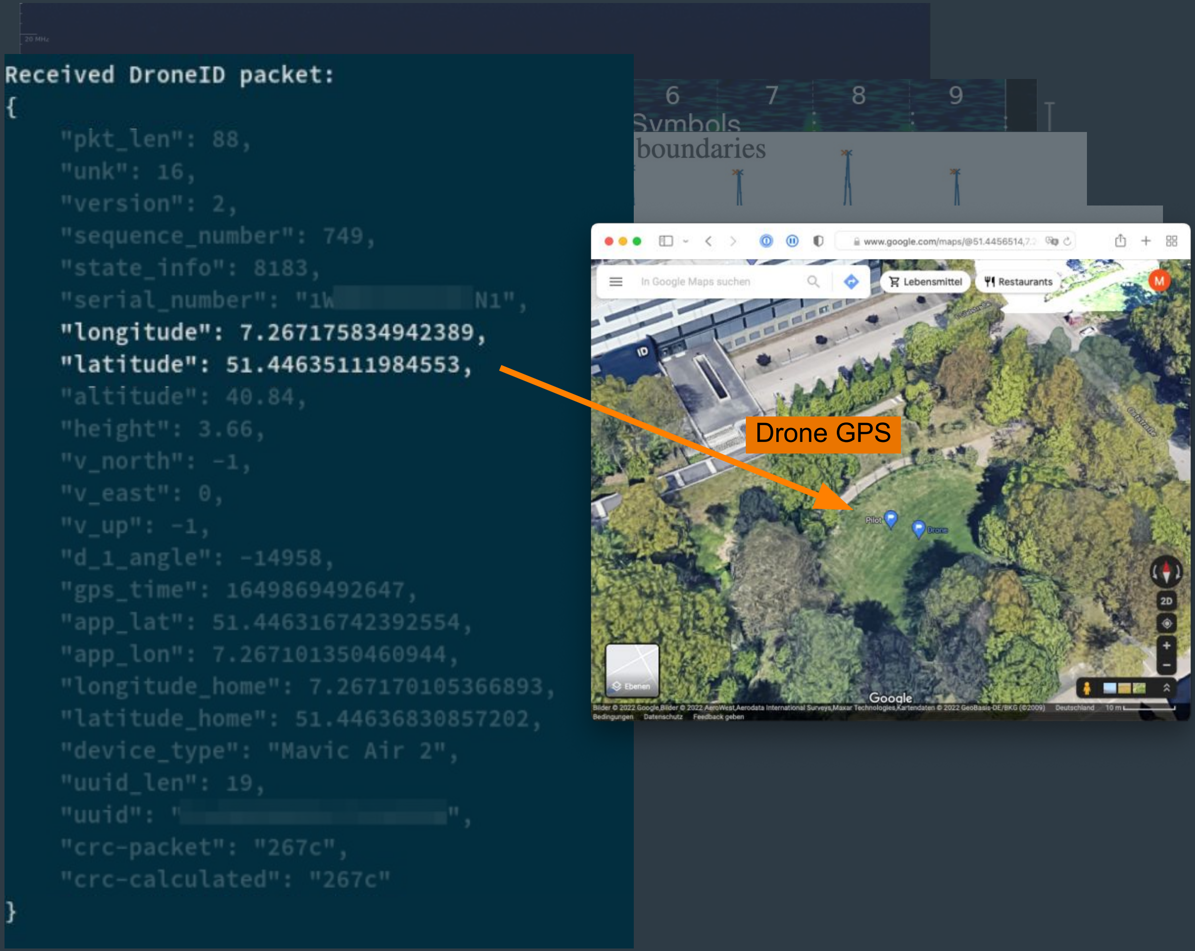
Reverse Engineering a Signal



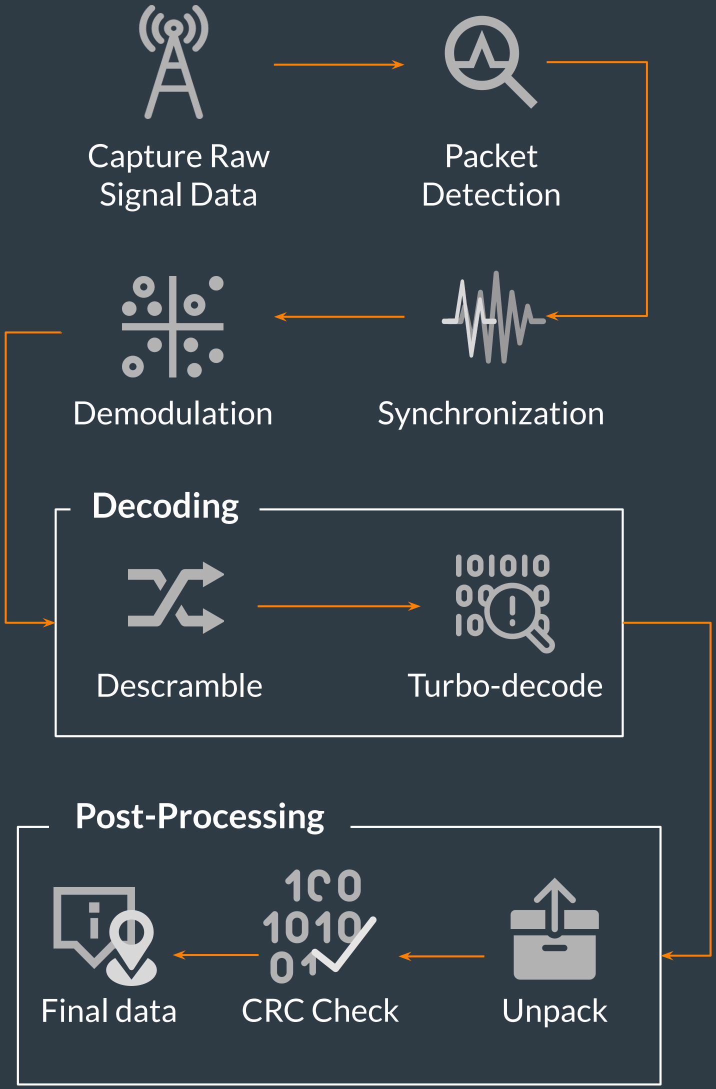
```
Received DroneID packet:
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_number": 749,
    "state_info": 8183,
    "serial_number": "1W[redacted]N1",
    "longitude": 7.267175834942389,
    "latitude": 51.44635111984553,
    "altitude": 40.84,
    "height": 3.66,
    "v_north": -1,
    "v_east": 0,
    "v_up": -1,
    "d_1_angle": -14958,
    "gps_time": 1649869492647,
    "app_lat": 51.446316742392554,
    "app_lon": 7.267101350460944,
    "longitude_home": 7.267170105366893,
    "latitude_home": 51.44636830857202,
    "device_type": "Mavic Air 2",
    "uuid_len": 19,
    "uuid": "[redacted]",
    "crc-packet": "267c",
    "crc-calculated": "267c"
}
```



Reverse Engineering a Signal



```
Received DroneID packet:
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_number": 749,
    "state_info": 8183,
    "serial_number": "1W[redacted]N1",
    "longitude": 7.267175834942389,
    "latitude": 51.44635111984553,
    "altitude": 40.84,
    "height": 3.66,
    "v_north": -1,
    "v_east": 0,
    "v_up": -1,
    "d_1_angle": -14958,
    "gps_time": 1649869492647,
    "app_lat": 51.446316742392554,
    "app_lon": 7.267101350460944,
    "longitude_home": 7.267170105366893,
    "latitude_home": 51.44636830857202,
    "device_type": "Mavic Air 2",
    "uuid_len": 19,
    "uuid": "[redacted]",
    "crc-packet": "267c",
    "crc-calculated": "267c"
}
```



Capture Raw Signal Data

Packet Detection

Synchronization

Demodulation

Decoding

Descramble

Turbo-decode

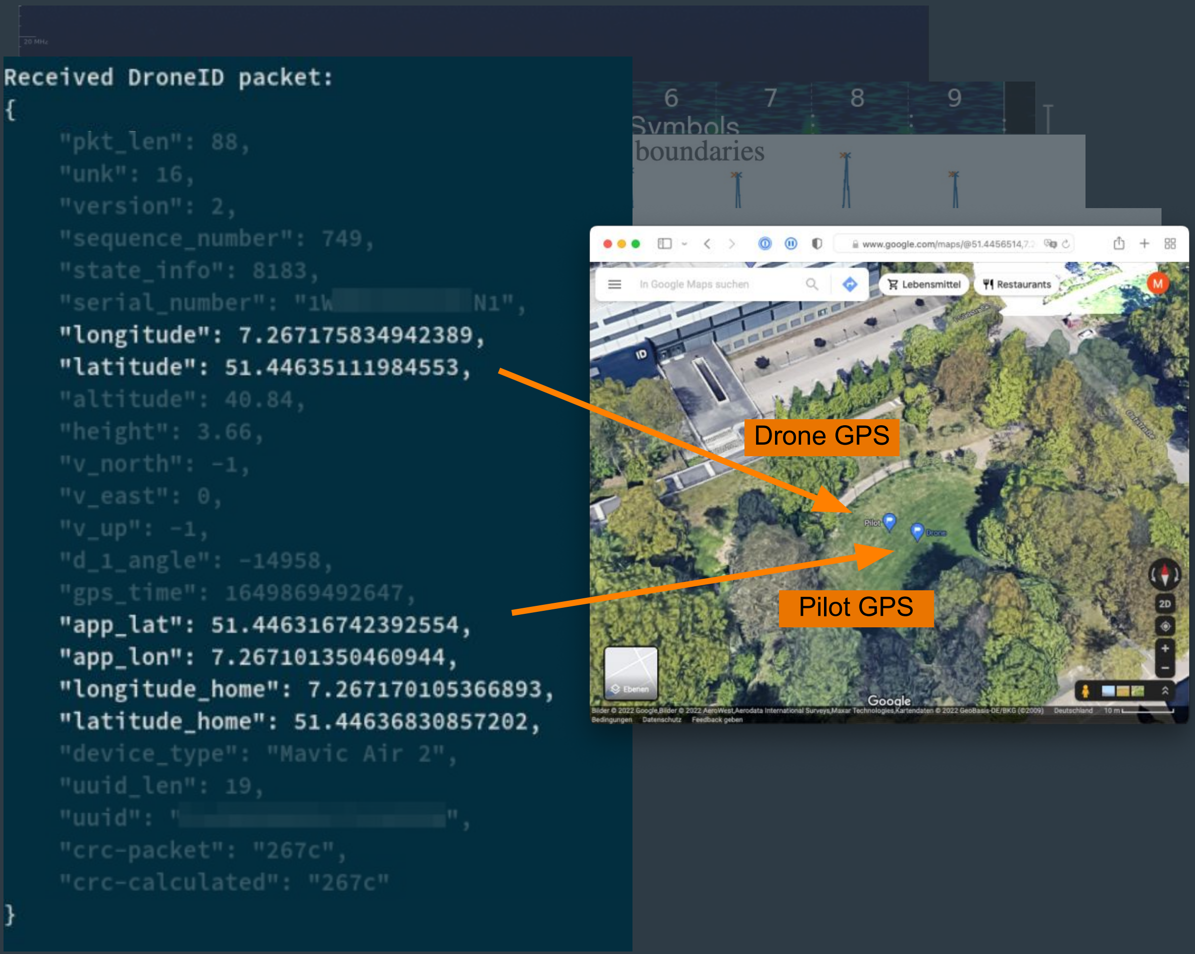
Post-Processing

Unpack

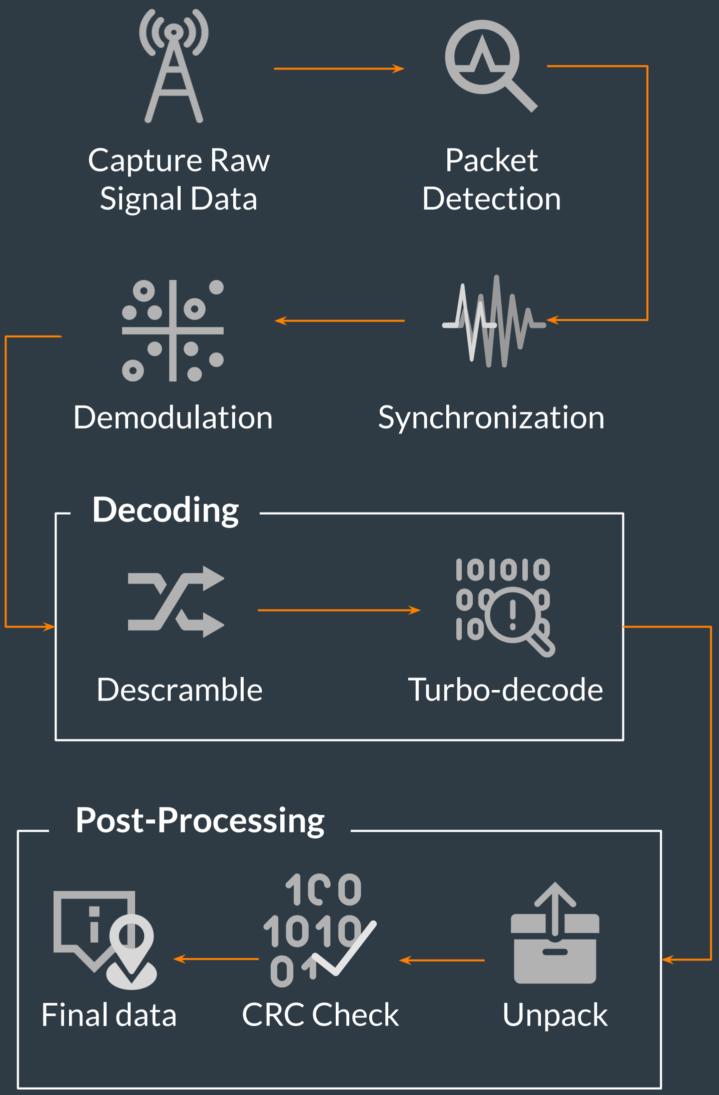
CRC Check

Final data

Reverse Engineering a Signal



```
Received DroneID packet:
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_number": 749,
    "state_info": 8183,
    "serial_number": "1W[redacted]N1",
    "longitude": 7.267175834942389,
    "latitude": 51.44635111984553,
    "altitude": 40.84,
    "height": 3.66,
    "v_north": -1,
    "v_east": 0,
    "v_up": -1,
    "d_1_angle": -14958,
    "gps_time": 1649869492647,
    "app_lat": 51.446316742392554,
    "app_lon": 7.267101350460944,
    "longitude_home": 7.267170105366893,
    "latitude_home": 51.44636830857202,
    "device_type": "Mavic Air 2",
    "uuid_len": 19,
    "uuid": "[redacted]",
    "crc-packet": "267c",
    "crc-calculated": "267c"
}
```



Capture Raw Signal Data

Packet Detection

Synchronization

Demodulation

Decoding

Descramble

Turbo-decode

Post-Processing

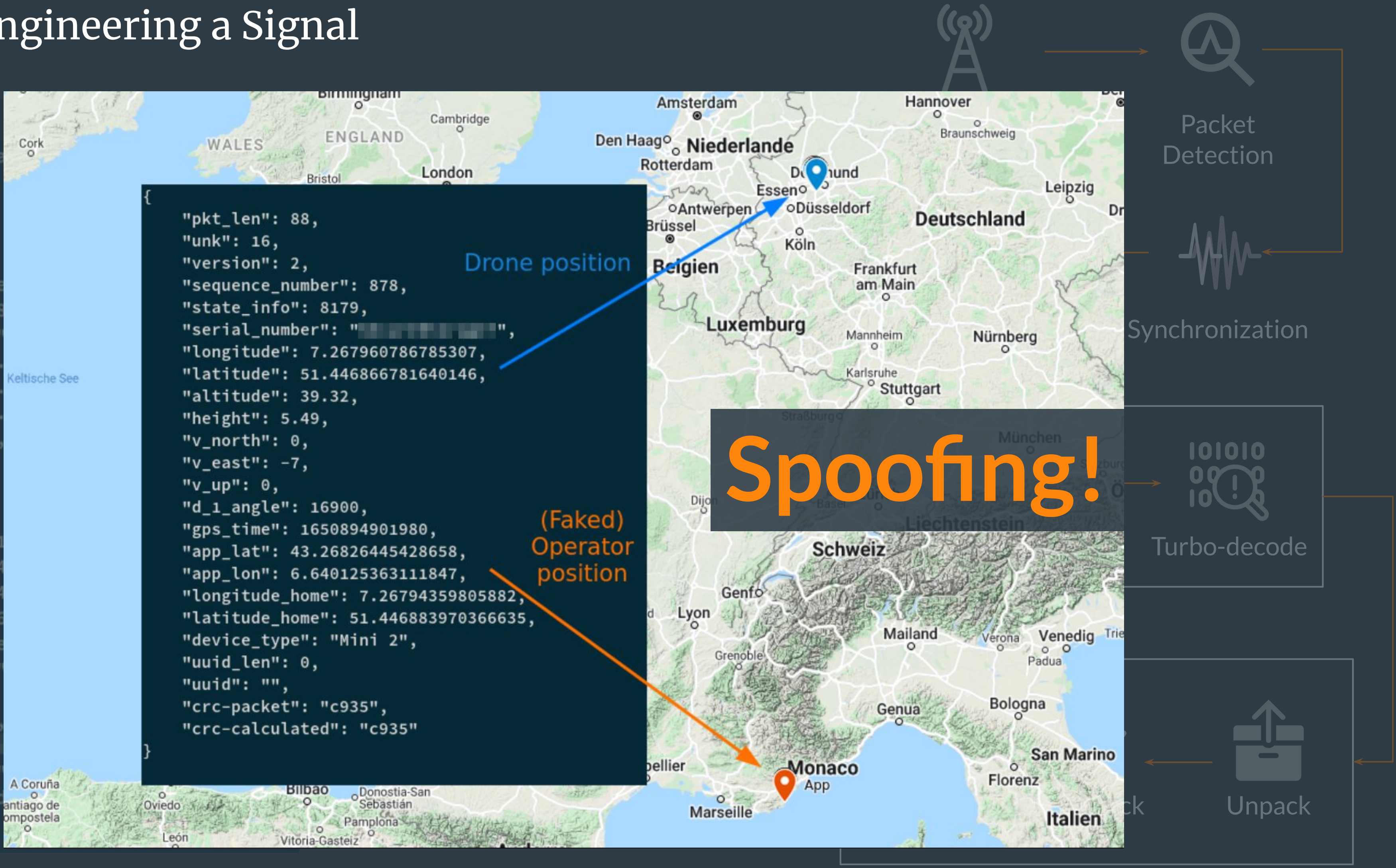
Unpack

CRC Check

Final data

Reverse Engineering a Signal

```
Received DroneID pa
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_numbe
    "state_info": 8
    "serial_number"
    "longitude": 7.
    "latitude": 51.
    "altitude": 40.
    "height": 3.66,
    "v_north": -1,
    "v_east": 0,
    "v_up": -1,
    "d_1_angle": -1
    "gps_time": 164
    "app_lat": 51.4
    "app_lon": 7.26
    "longitude_home
    "latitude_home"
    "device_type": 
    "uuid_len": 19,
    "uuid": "[redacted]
    "crc-packet": "
    "crc-calculated
}
```



```
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_number": 878,
    "state_info": 8179,
    "serial_number": "[redacted]",
    "longitude": 7.267960786785307,
    "latitude": 51.446866781640146,
    "altitude": 39.32,
    "height": 5.49,
    "v_north": 0,
    "v_east": -7,
    "v_up": 0,
    "d_1_angle": 16900,
    "gps_time": 1650894901980,
    "app_lat": 43.26826445428658,
    "app_lon": 6.640125363111847,
    "longitude_home": 7.26794359805882,
    "latitude_home": 51.446883970366635,
    "device_type": "Mini 2",
    "uuid_len": 0,
    "uuid": "",
    "crc-packet": "c935",
    "crc-calculated": "c935"
}
```

Drone position

(Faked) Operator position

Spoofing!

Summary: Wireless Physical Layer

- Much information is broadcast, including:
  - Drone location
  - Pilot location
  - Serial number

- Signal not encrypted

- But: Easy to spoof the pilot location

Wireless Physical Layer

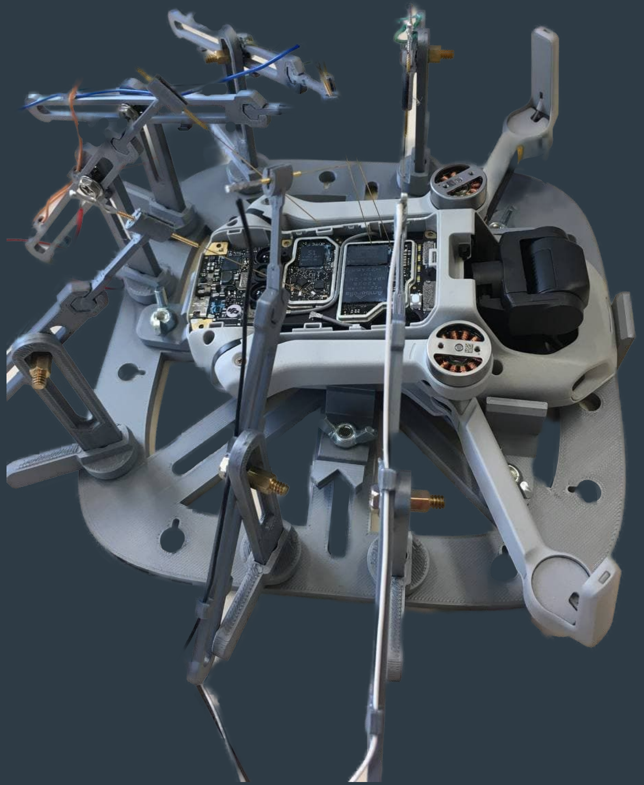
Reversing DJI DroneID

Static Analysis

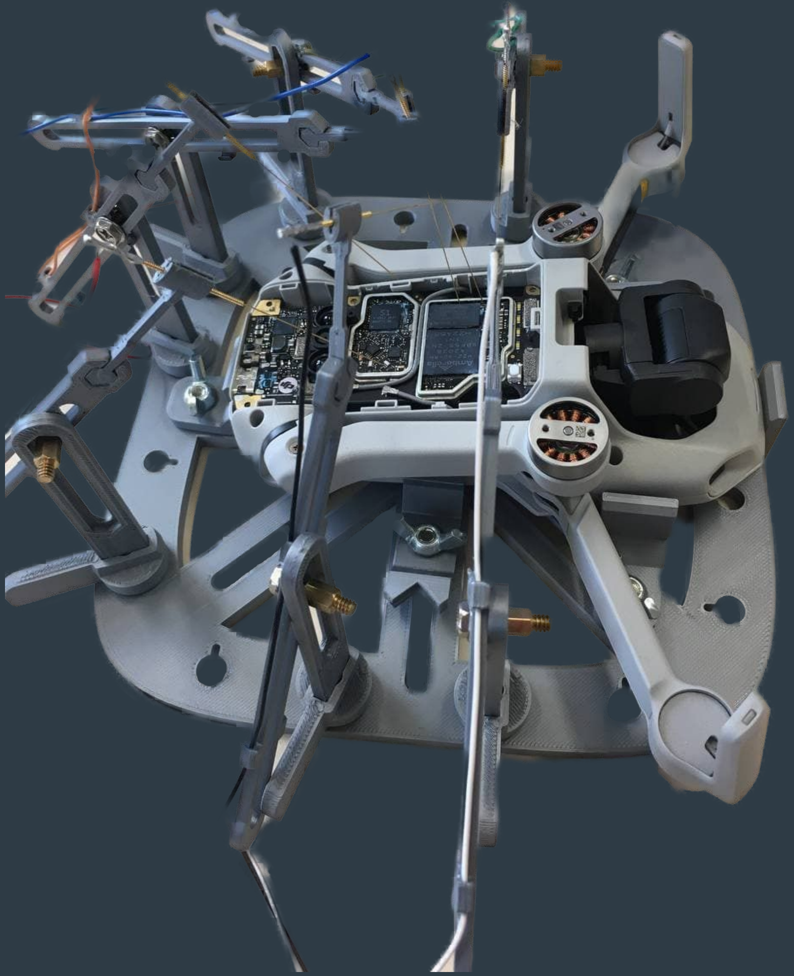
Hands on the Drone

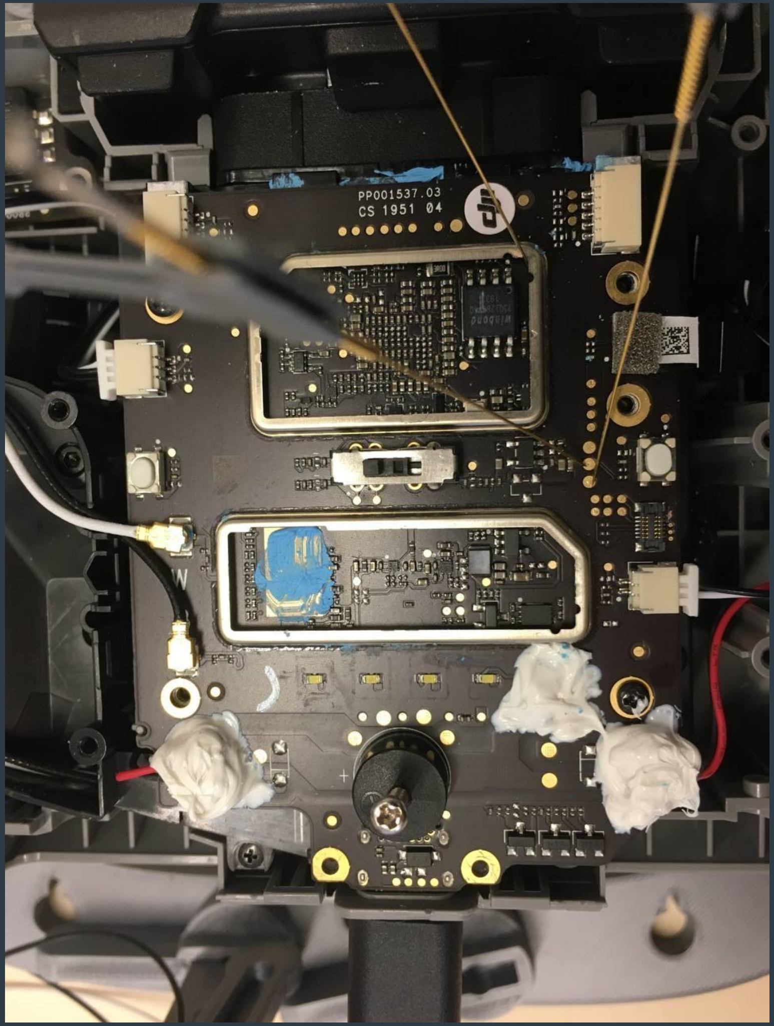
Dynamic Analysis

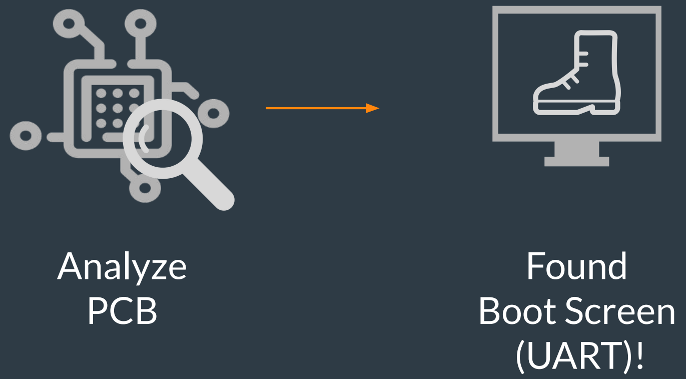
Fuzzing Drones for Pain and Profit



Analyze
PCB

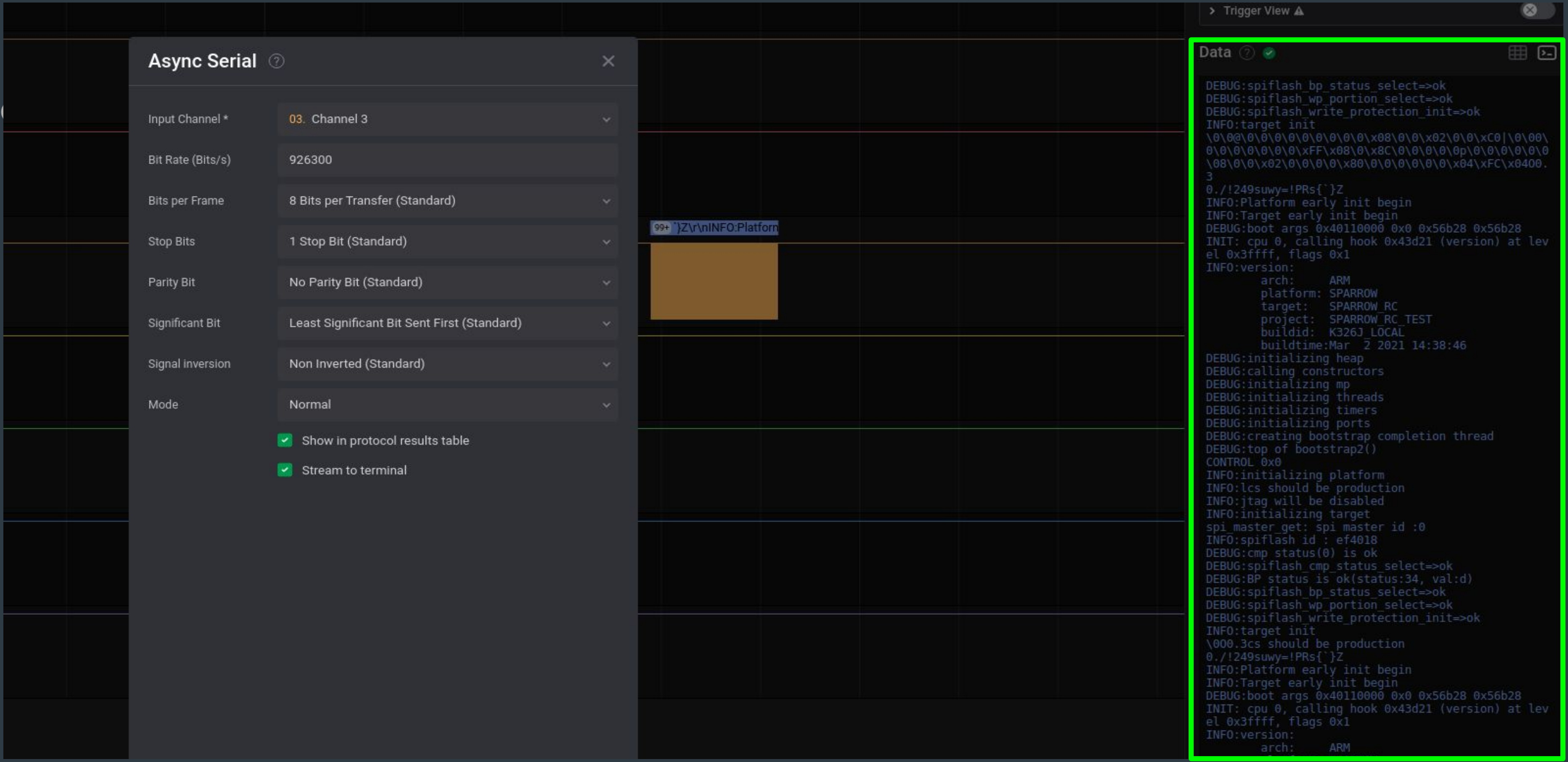



PP001537.03
CS 1951 04



Analyze
PCB

Found
Boot Screen
(UART)!



Async Serial

Input Channel * 03. Channel 3

Bit Rate (Bits/s) 926300

Bits per Frame 8 Bits per Transfer (Standard)

Stop Bits 1 Stop Bit (Standard)

Parity Bit No Parity Bit (Standard)

Significant Bit Least Significant Bit Sent First (Standard)

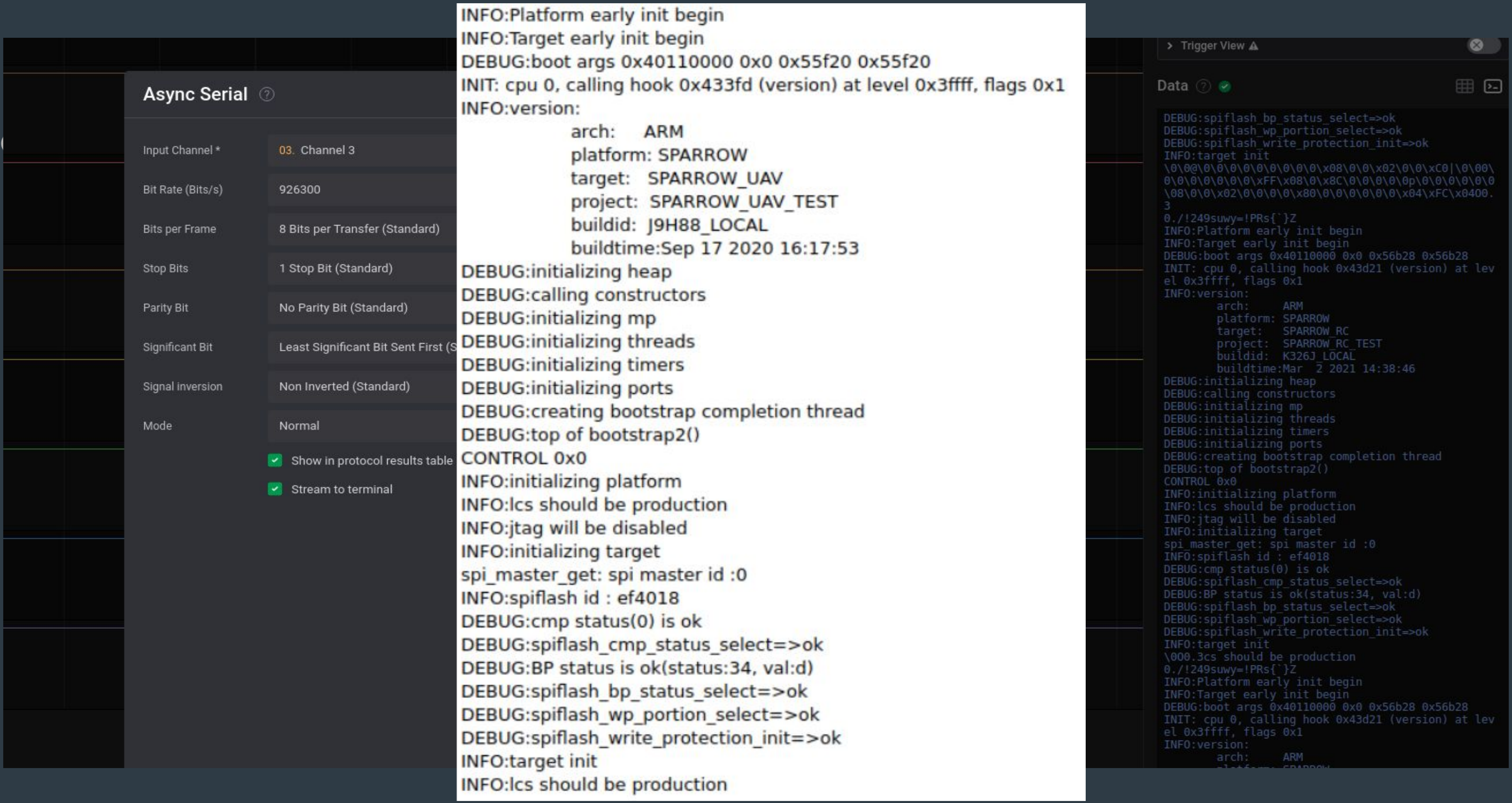
Signal inversion Non Inverted (Standard)

Mode Normal

☑ Show in protocol results table

☑ Stream to terminal

```
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0\0@\0\0\0\0\0\0\0\0\x08\0\0\x02\0\0\xC0|\0\00\
0\0\0\0\0\0\0\xFF\x08\0\x8C\0\0\0\0\0p\0\0\0\0\0
\08\0\0\x02\0\0\0\0\x80\0\0\0\0\0\0\x04\xFC\x0400.
3
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_RC
        project:  SPARROW_RC_TEST
        buildid:  K326J_LOCAL
        buildtime:Mar  2 2021 14:38:46
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\000.3cs should be production
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
```



Async Serial

Input Channel * 03. Channel 3

Bit Rate (Bits/s) 926300

Bits per Frame 8 Bits per Transfer (Standard)

Stop Bits 1 Stop Bit (Standard)

Parity Bit No Parity Bit (Standard)

Significant Bit Least Significant Bit Sent First (S

Signal inversion Non Inverted (Standard)

Mode Normal

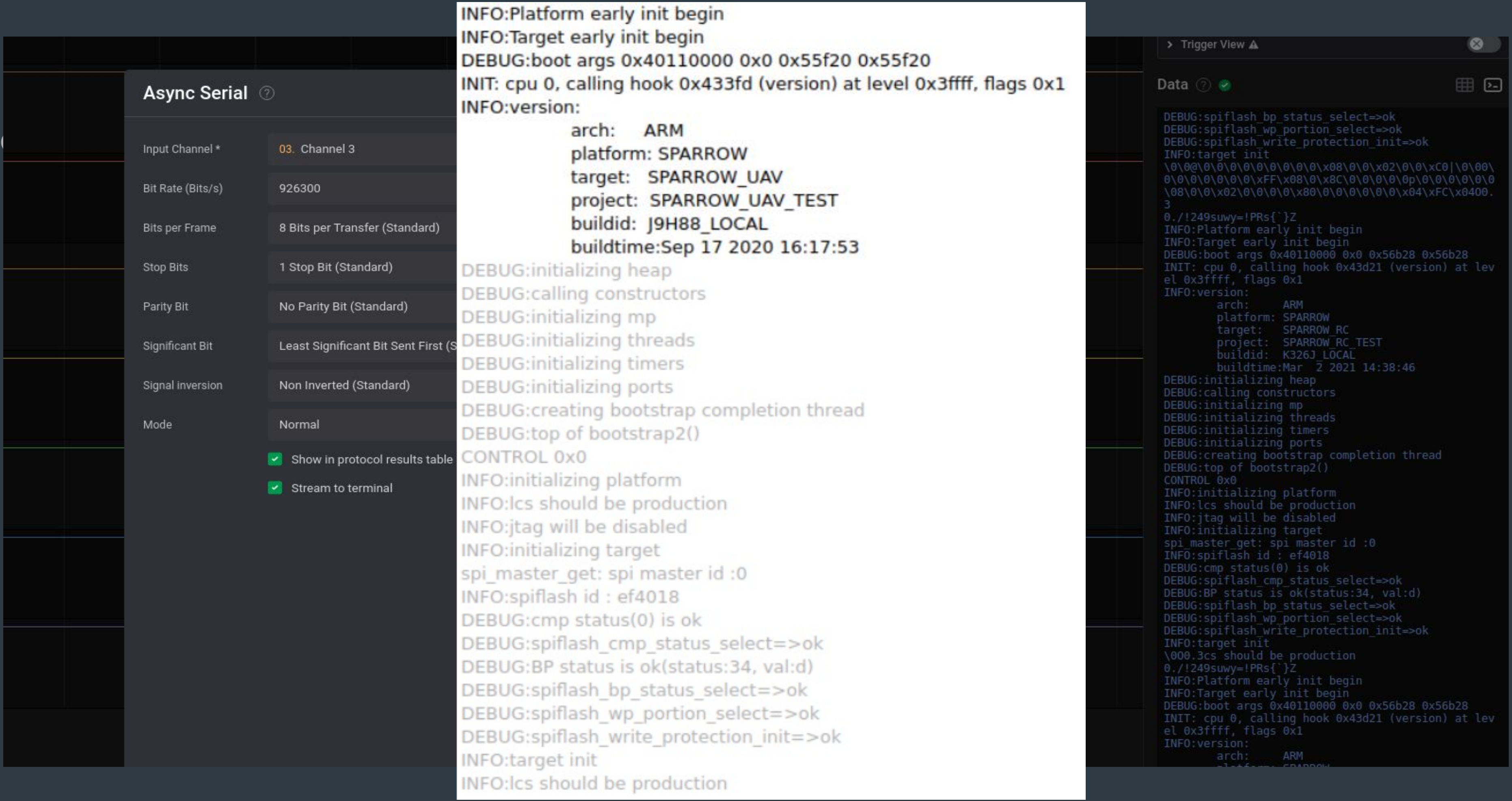
☑ Show in protocol results table

☑ Stream to terminal

```
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x55f20 0x55f20
INIT: cpu 0, calling hook 0x433fd (version) at level 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_UAV
        project:  SPARROW_UAV_TEST
        buildid:  J9H88_LOCAL
        buildtime:Sep 17 2020 16:17:53
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
INFO:lcs should be production
```

Data

```
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0\0@\0\0\0\0\0\0\0\0\0\x08\0\0\x02\0\0\xC0|\0\00\
0\0\0\0\0\0\0\xFF\x08\0\x8C\0\0\0\0\0p\0\0\0\0\0\0
\08\0\0\x02\0\0\0\0\x80\0\0\0\0\0\0\x04\xFC\x0400.
3
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_RC
        project:  SPARROW_RC_TEST
        buildid:  K326J_LOCAL
        buildtime:Mar  2 2021 14:38:46
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0O0.3cs should be production
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
```



Async Serial

Input Channel *: 03. Channel 3

Bit Rate (Bits/s): 926300

Bits per Frame: 8 Bits per Transfer (Standard)

Stop Bits: 1 Stop Bit (Standard)

Parity Bit: No Parity Bit (Standard)

Significant Bit: Least Significant Bit Sent First (S

Signal inversion: Non Inverted (Standard)

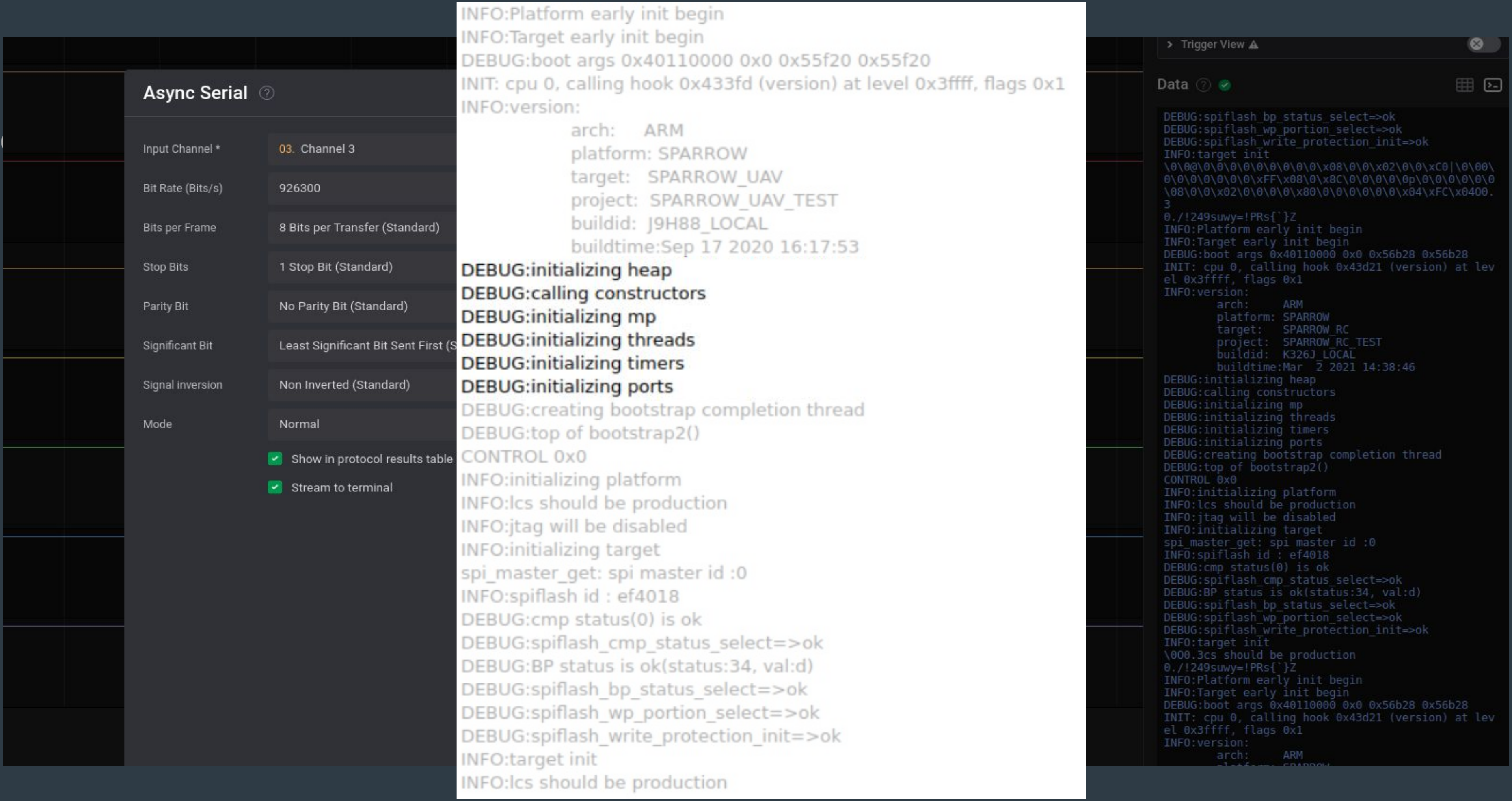
Mode: Normal

☑ Show in protocol results table

☑ Stream to terminal

```
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x55f20 0x55f20
INIT: cpu 0, calling hook 0x433fd (version) at level 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_UAV
        project:  SPARROW_UAV_TEST
        buildid:  J9H88_LOCAL
        buildtime:Sep 17 2020 16:17:53
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
INFO:lcs should be production
```

```
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0\0@\0\0\0\0\0\0\0\0\0\x08\0\0\x02\0\0\xC0|\0\00\
0\0\0\0\0\0\0\xFF\x08\0\x8C\0\0\0\0\0p\0\0\0\0\0\0
\08\0\0\x02\0\0\0\0\x80\0\0\0\0\0\0\x04\xFC\x0400.
3
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_RC
        project:  SPARROW_RC_TEST
        buildid:  K326J_LOCAL
        buildtime:Mar  2 2021 14:38:46
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\000.3cs should be production
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
```



Async Serial

Input Channel * 03. Channel 3

Bit Rate (Bits/s) 926300

Bits per Frame 8 Bits per Transfer (Standard)

Stop Bits 1 Stop Bit (Standard)

Parity Bit No Parity Bit (Standard)

Significant Bit Least Significant Bit Sent First (S

Signal inversion Non Inverted (Standard)

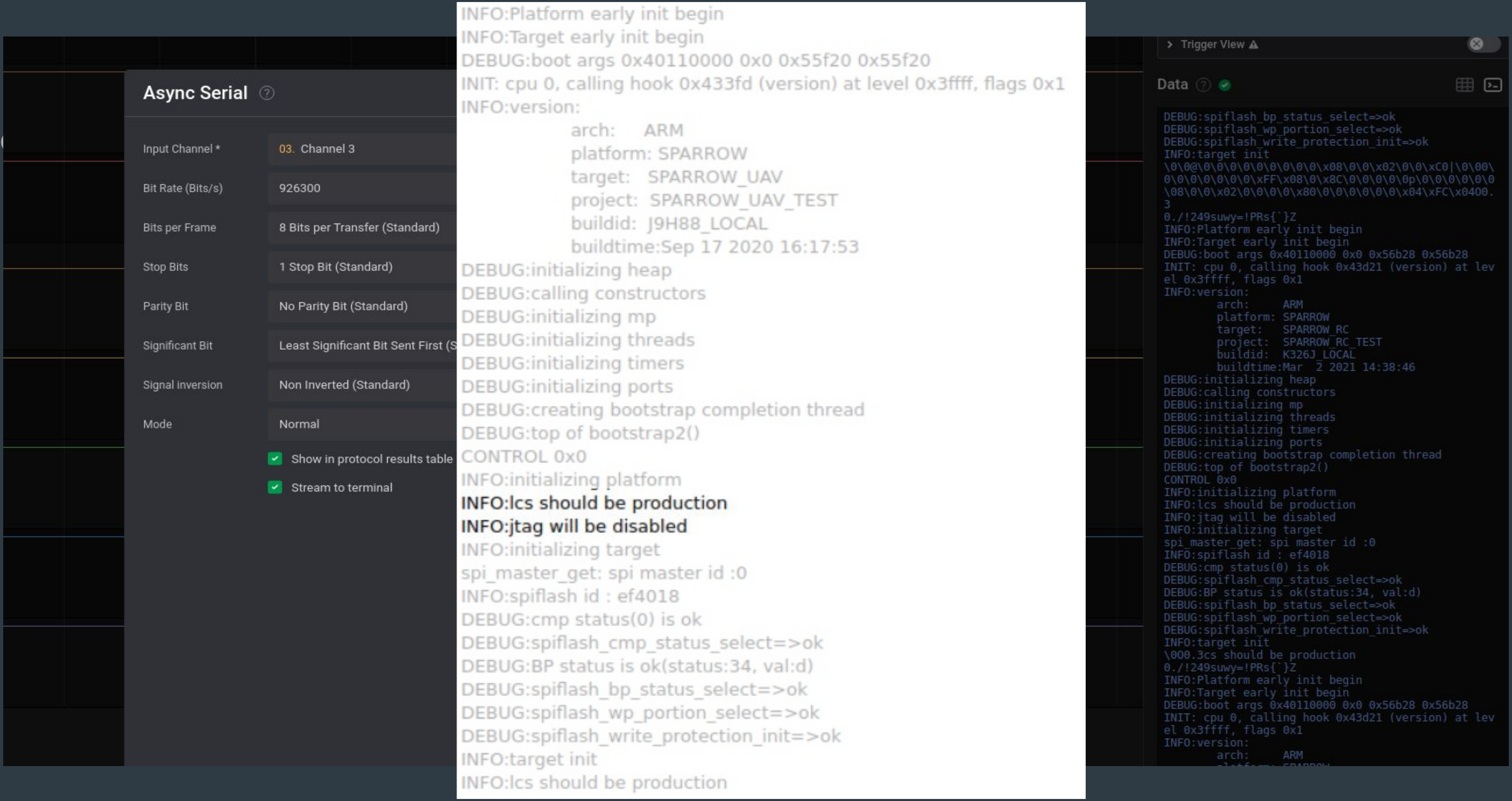
Mode Normal

☑ Show in protocol results table

☑ Stream to terminal

```
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x55f20 0x55f20
INIT: cpu 0, calling hook 0x433fd (version) at level 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_UAV
        project:  SPARROW_UAV_TEST
        buildid:  J9H88_LOCAL
        buildtime:Sep 17 2020 16:17:53
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
INFO:lcs should be production
```

```
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0\0@\0\0\0\0\0\0\0\0\0\x08\0\0\x02\0\0\xC0|\0\00\
0\0\0\0\0\0\0\xFF\x08\0\x8C\0\0\0\0\0p\0\0\0\0\0\0
\08\0\0\x02\0\0\0\0\x80\0\0\0\0\0\0\x04\xFC\x0400.
3
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_RC
        project:  SPARROW_RC_TEST
        buildid:  K326J_LOCAL
        buildtime:Mar  2 2021 14:38:46
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0O0.3cs should be production
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
```



Async Serial

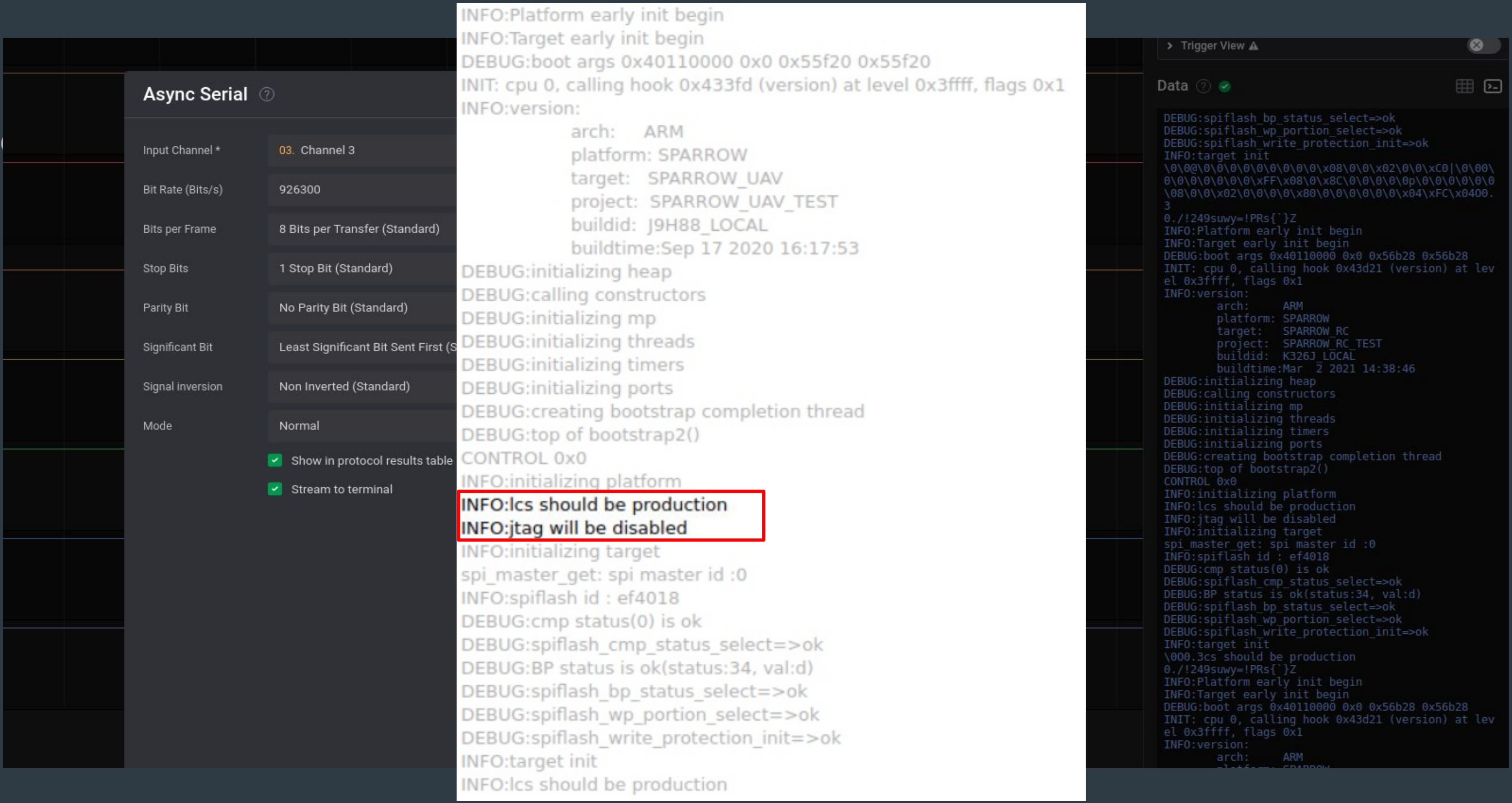
Input Channel *	03. Channel 3
Bit Rate (Bits/s)	926300
Bits per Frame	8 Bits per Transfer (Standard)
Stop Bits	1 Stop Bit (Standard)
Parity Bit	No Parity Bit (Standard)
Significant Bit	Least Significant Bit Sent First (S
Signal inversion	Non Inverted (Standard)
Mode	Normal

☑ Show in protocol results table

☑ Stream to terminal

```
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x55f20 0x55f20
INIT: cpu 0, calling hook 0x433fd (version) at level 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_UAV
        project:  SPARROW_UAV_TEST
        buildid:  J9H88_LOCAL
        buildtime:Sep 17 2020 16:17:53
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
INFO:lcs should be production
```

```
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0\0@\0\0\0\0\0\0\0\0\x08\0\0\x02\0\0\xC0|\0\00\
0\0\0\0\0\0\0\xFF\x08\0\x8C\0\0\0\0\0p\0\0\0\0\0\0
\08\0\0\x02\0\0\0\0\x80\0\0\0\0\0\0\x04\xFC\x0400.
3
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_RC
        project:  SPARROW_RC_TEST
        buildid:  K326J_LOCAL
        buildtime:Mar  2 2021 14:38:46
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\000.3cs should be production
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
```



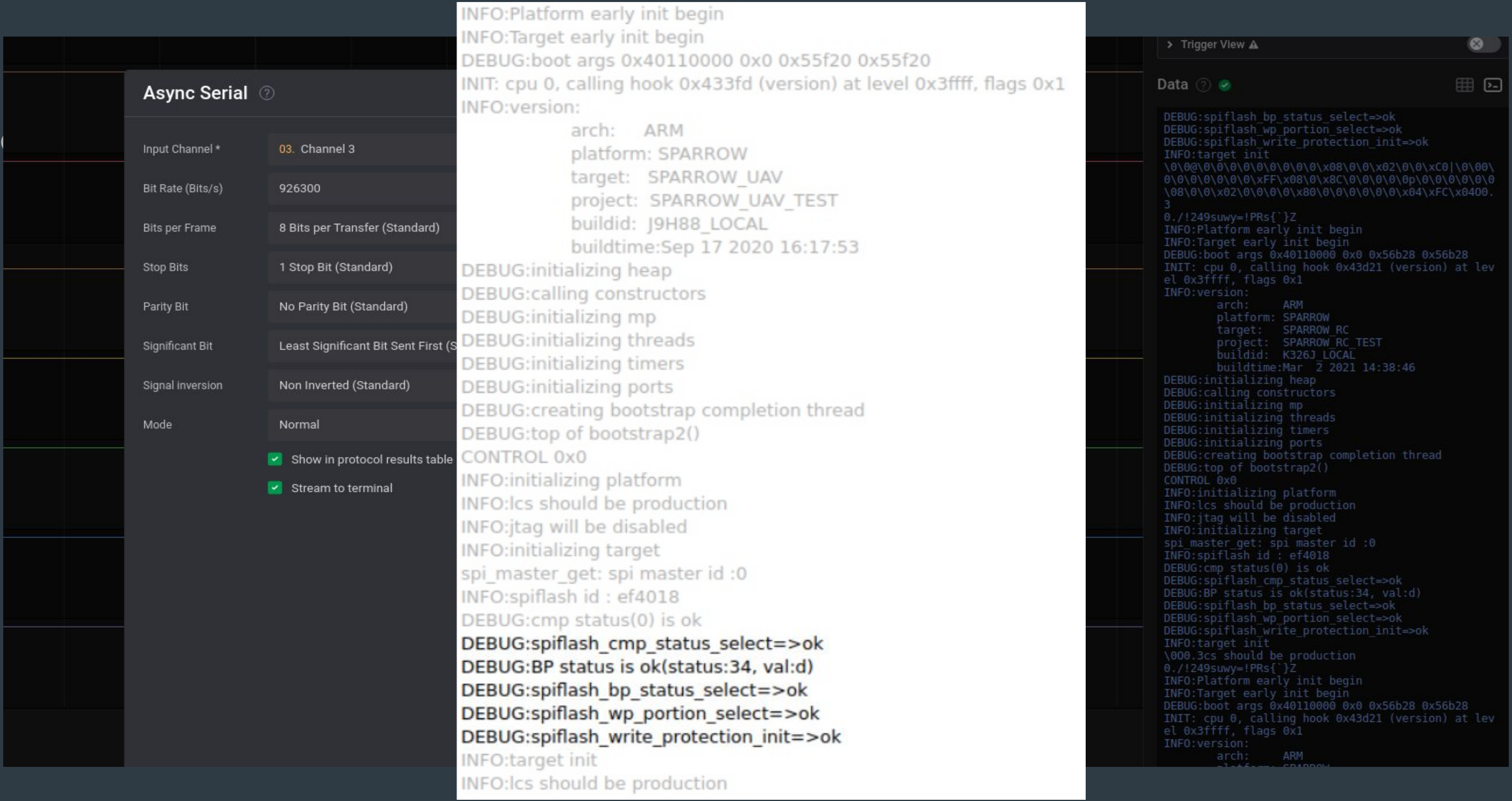
Async Serial

Input Channel * 03. Channel 3
Bit Rate (Bits/s) 926300
Bits per Frame 8 Bits per Transfer (Standard)
Stop Bits 1 Stop Bit (Standard)
Parity Bit No Parity Bit (Standard)
Significant Bit Least Significant Bit Sent First (S
Signal inversion Non Inverted (Standard)
Mode Normal

☑ Show in protocol results table
☑ Stream to terminal

```
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x55f20 0x55f20
INIT: cpu 0, calling hook 0x433fd (version) at level 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_UAV
        project:  SPARROW_UAV_TEST
        buildid:  J9H88_LOCAL
        buildtime:Sep 17 2020 16:17:53
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
INFO:lcs should be production
```

```
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0\0@\0\0\0\0\0\0\0\0\0\x08\0\0\x02\0\0\xC0|\0\00\
0\0\0\0\0\0\0\xFF\x08\0\x8C\0\0\0\0\0p\0\0\0\0\0\0
\08\0\0\x02\0\0\0\0\x80\0\0\0\0\0\0\x04\xFC\x0400.
3
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_RC
        project:  SPARROW_RC_TEST
        buildid:  K326J_LOCAL
        buildtime:Mar  2 2021 14:38:46
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0O0.3cs should be production
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
```



Async Serial

Input Channel * 03. Channel 3

Bit Rate (Bits/s) 926300

Bits per Frame 8 Bits per Transfer (Standard)

Stop Bits 1 Stop Bit (Standard)

Parity Bit No Parity Bit (Standard)

Significant Bit Least Significant Bit Sent First (S

Signal inversion Non Inverted (Standard)

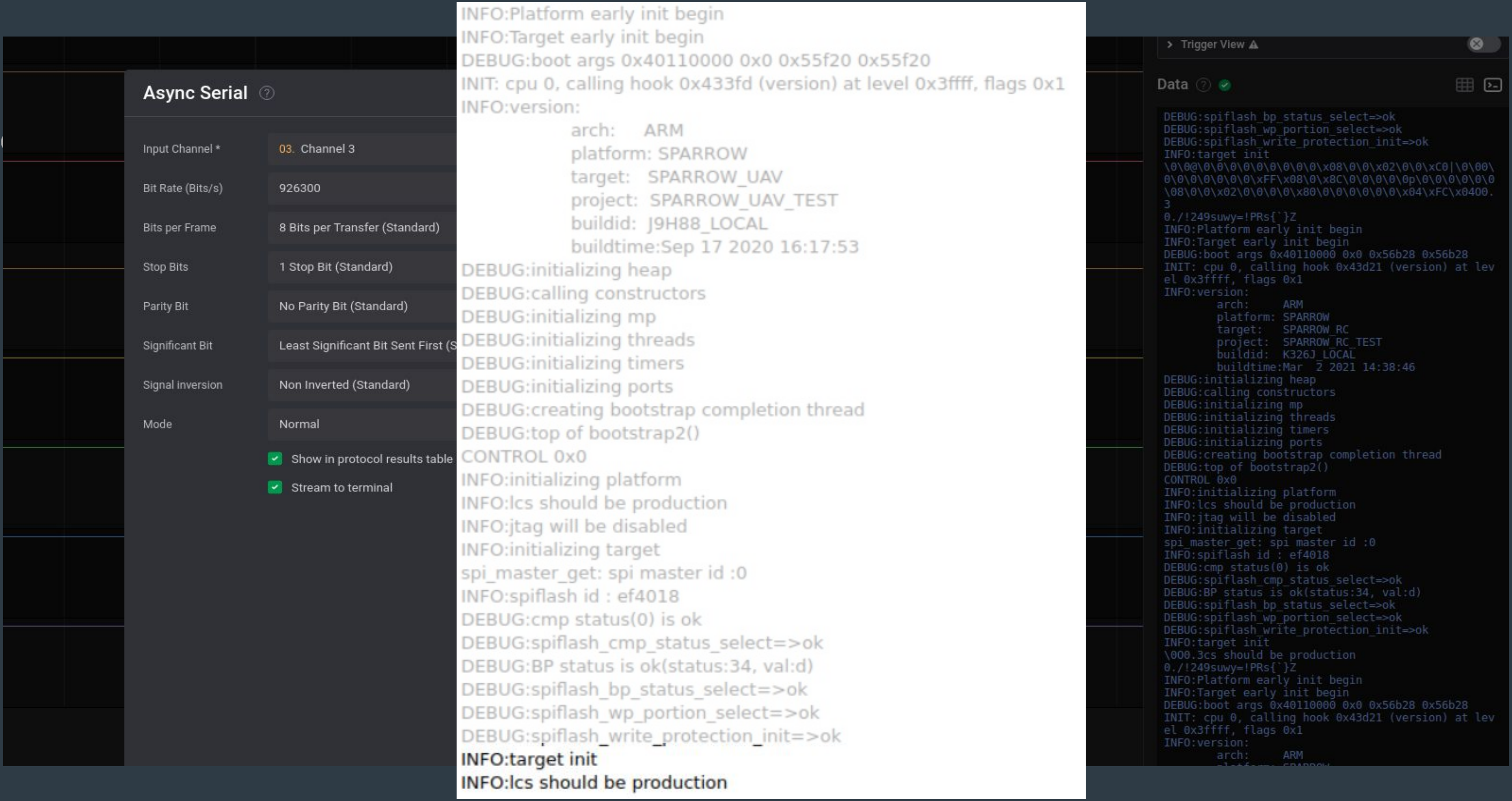
Mode Normal

☑ Show in protocol results table

☑ Stream to terminal

```
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x55f20 0x55f20
INIT: cpu 0, calling hook 0x433fd (version) at level 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_UAV
        project:  SPARROW_UAV_TEST
        buildid:  J9H88_LOCAL
        buildtime:Sep 17 2020 16:17:53
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
INFO:lcs should be production
```

```
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0\0@\0\0\0\0\0\0\0\0\0\x08\0\0\x02\0\0\xC0|\0\00\
0\0\0\0\0\0\0\0\xFF\x08\0\x8C\0\0\0\0\0p\0\0\0\0\0\0
\08\0\0\x02\0\0\0\0\x80\0\0\0\0\0\0\x04\xFC\x0400.
3
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_RC
        project:  SPARROW_RC_TEST
        buildid:  K326J_LOCAL
        buildtime:Mar  2 2021 14:38:46
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\000.3cs should be production
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
```



Async Serial

Input Channel * 03. Channel 3

Bit Rate (Bits/s) 926300

Bits per Frame 8 Bits per Transfer (Standard)

Stop Bits 1 Stop Bit (Standard)

Parity Bit No Parity Bit (Standard)

Significant Bit Least Significant Bit Sent First (S

Signal inversion Non Inverted (Standard)

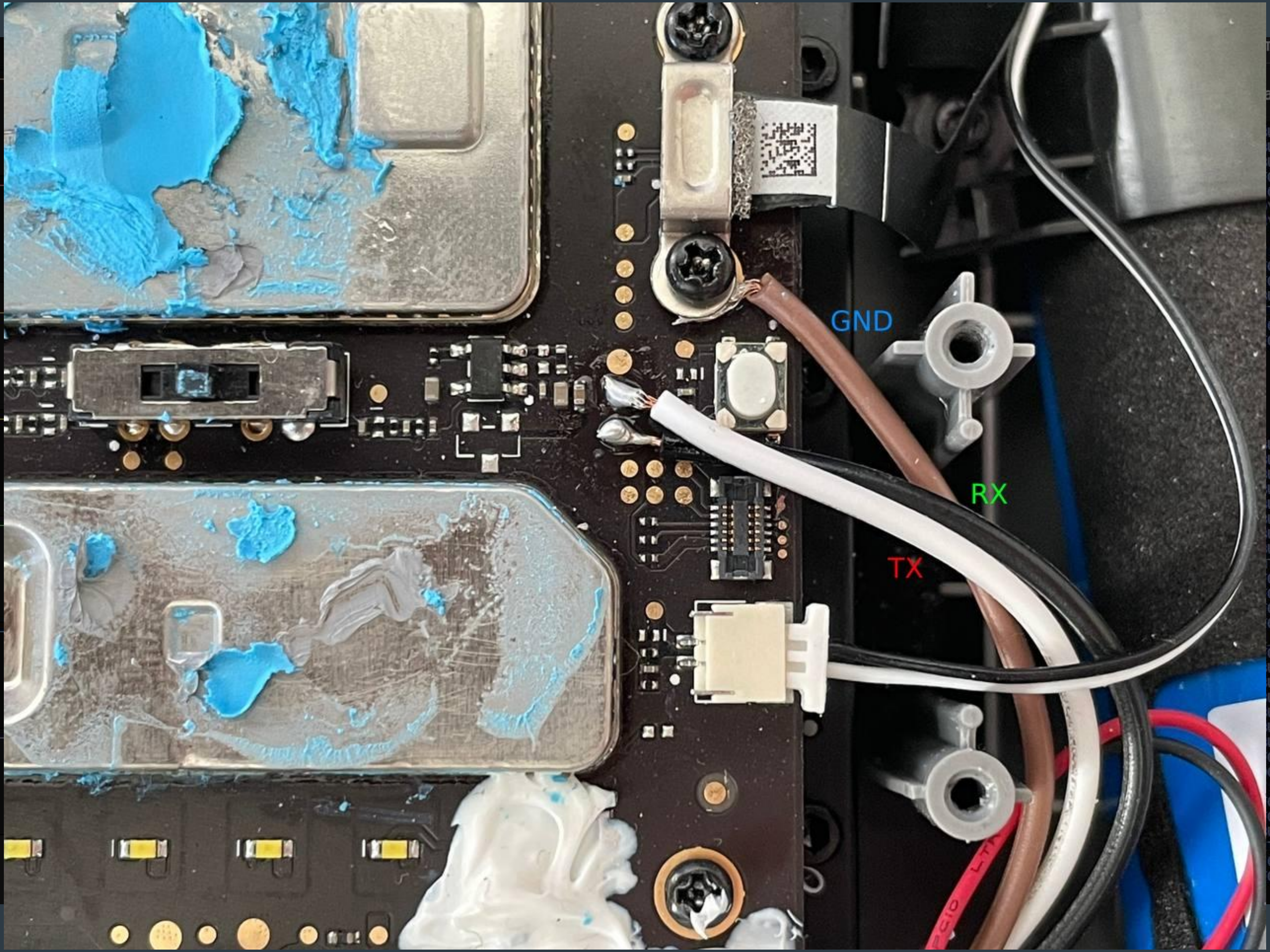
Mode Normal

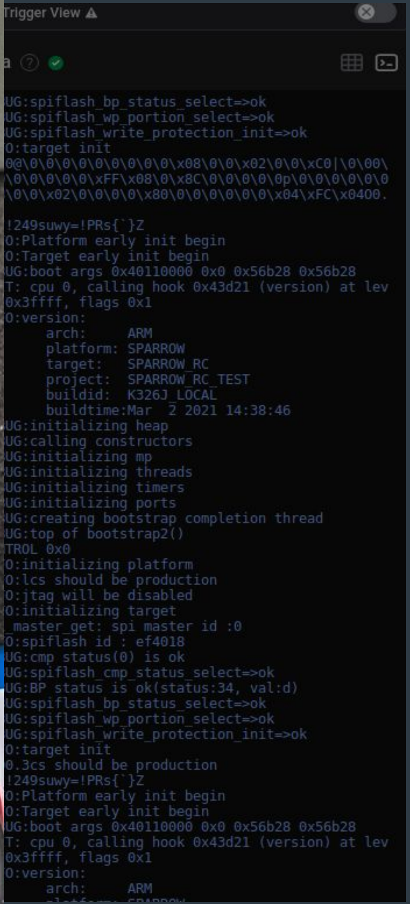
☑ Show in protocol results table

☑ Stream to terminal

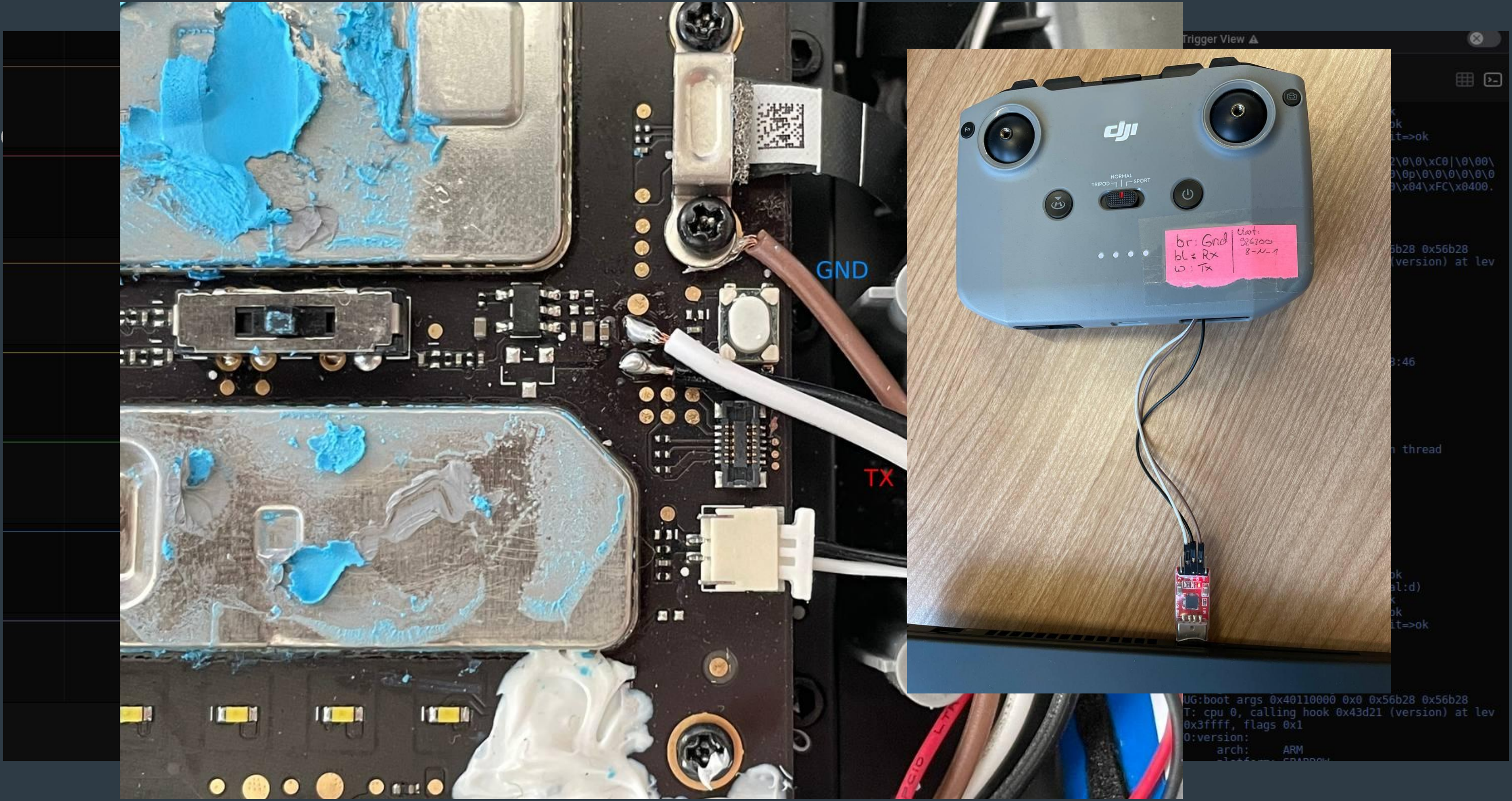
```
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x55f20 0x55f20
INIT: cpu 0, calling hook 0x433fd (version) at level 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_UAV
        project:  SPARROW_UAV_TEST
        buildid:  J9H88_LOCAL
        buildtime:Sep 17 2020 16:17:53
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
INFO:lcs should be production
```

```
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0\0@\0\0\0\0\0\0\0\0\x08\0\0\x02\0\0\xC0|\0\00\
0\0\0\0\0\0\0\xFF\x08\0\x8C\0\0\0\0\0p\0\0\0\0\0\0
\08\0\0\x02\0\0\0\0\x80\0\0\0\0\0\0\x04\xFC\x0400.
3
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
        platform: SPARROW
        target:   SPARROW_RC
        project:  SPARROW_RC_TEST
        buildid:  K326J_LOCAL
        buildtime:Mar  2 2021 14:38:46
DEBUG:initializing heap
DEBUG:calling constructors
DEBUG:initializing mp
DEBUG:initializing threads
DEBUG:initializing timers
DEBUG:initializing ports
DEBUG:creating bootstrap completion thread
DEBUG:top of bootstrap2()
CONTROL 0x0
INFO:initializing platform
INFO:lcs should be production
INFO:jtag will be disabled
INFO:initializing target
spi_master_get: spi master id :0
INFO:spiflash id : ef4018
DEBUG:cmp status(0) is ok
DEBUG:spiflash_cmp_status_select=>ok
DEBUG:BP status is ok(status:34, val:d)
DEBUG:spiflash_bp_status_select=>ok
DEBUG:spiflash_wp_portion_select=>ok
DEBUG:spiflash_write_protection_init=>ok
INFO:target init
\0O0.3cs should be production
0./!249suwy=!PRs{`}Z
INFO:Platform early init begin
INFO:Target early init begin
DEBUG:boot args 0x40110000 0x0 0x56b28 0x56b28
INIT: cpu 0, calling hook 0x43d21 (version) at lev
el 0x3ffff, flags 0x1
INFO:version:
        arch:     ARM
```

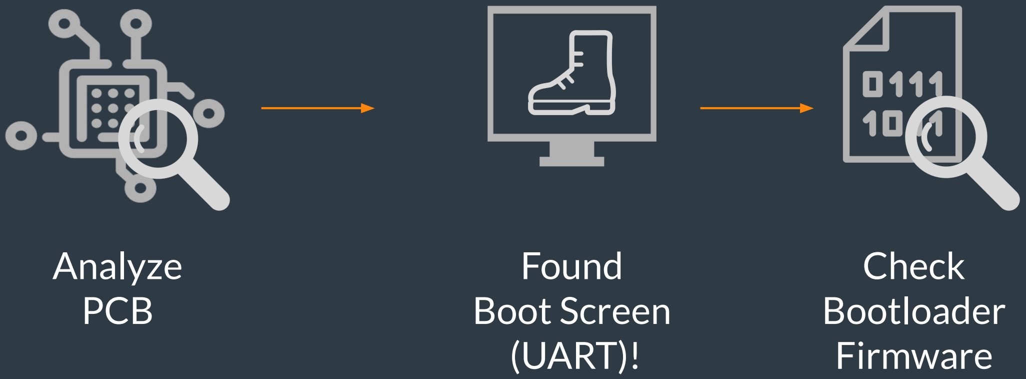




```
UG:spiflash_bp_status_select=>ok
UG:spiflash_wp_portion_select=>ok
UG:spiflash_write_protection_init=>ok
O:target init
0@\0\0\0\0\0\0\0\0\x08\0\0\x02\0\0\xC0|\0\00\
\0\0\0\0\0\xFF\x08\0\x8C\0\0\0\0\0p\0\0\0\0\0\0
\0\0\x02\0\0\0\0\x80\0\0\0\0\0\0\x04\xFC\x0400.

!249suwy=!PRs{`}Z
O:Platform early init begin
O:Target early init begin
UG:boot args 0x40110000 0x0 0x56b28 0x56b28
T: cpu 0, calling hook 0x43d21 (version) at lev
0x3ffff, flags 0x1
O:version:
     arch:     ARM
     platform: SPARROW
     target:   SPARROW_RC
     project:  SPARROW_RC_TEST
     buildid:  K326J_LOCAL
     buildtime:Mar  2 2021 14:38:46
UG:initializing heap
UG:calling constructors
UG:initializing mp
UG:initializing threads
UG:initializing timers
UG:initializing ports
UG:creating bootstrap completion thread
UG:top of bootstrap2()
TROL 0x0
O:initializing platform
O:lcs should be production
O:jtag will be disabled
O:initializing target
_master_get: spi master id :0
O:spiflash id : ef4018
UG:cmp status(0) is ok
UG:spiflash_cmp_status_select=>ok
UG:BP status is ok(status:34, val:d)
UG:spiflash_bp_status_select=>ok
UG:spiflash_wp_portion_select=>ok
UG:spiflash_write_protection_init=>ok
O:target init
0.3cs should be production
!249suwy=!PRs{`}Z
O:Platform early init begin
O:Target early init begin
UG:boot args 0x40110000 0x0 0x56b28 0x56b28
T: cpu 0, calling hook 0x43d21 (version) at lev
0x3ffff, flags 0x1
O:version:
     arch:     ARM
```



```
k
ok
it=>ok

2\0\0\xC0|\0\00\
0\0p\0\0\0\0\0\0
0\x04\xFC\x0400.

6b28 0x56b28
(version) at lev

3:46

n thread

ok
al:d)
k
ok
it=>ok

UG:boot args 0x40110000 0x0 0x56b28 0x56b28
T: cpu 0, calling hook 0x43d21 (version) at lev
0x3ffff, flags 0x1
O:version:
     arch:     ARM
```



Analyze PCB

Found Boot Screen (UART)!

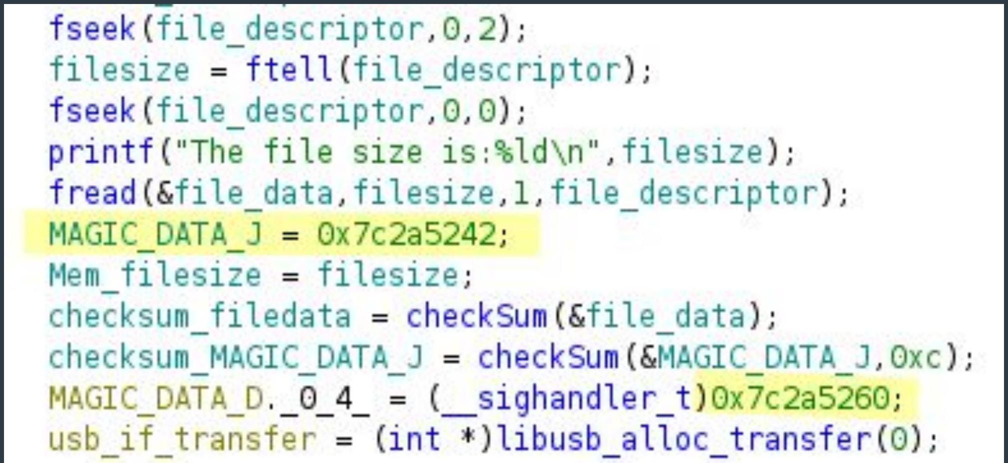
Check Bootloader Firmware

Analyze PCB

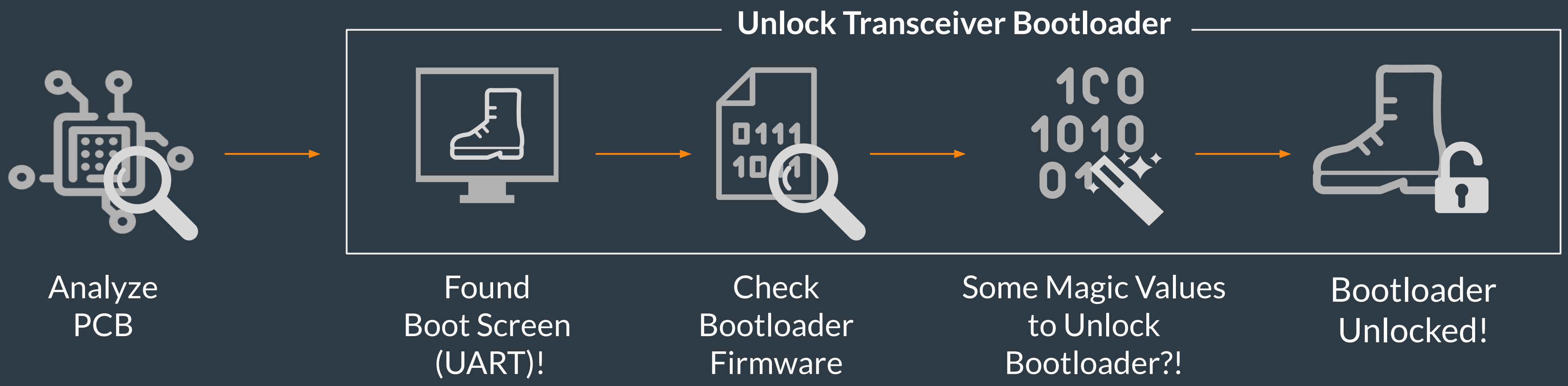
Found Boot Screen (UART)!

Check Bootloader Firmware

Some Magic Values to Unlock Bootloader?!



```
fseek(file_descriptor,0,2);
filesize = ftell(file_descriptor);
fseek(file_descriptor,0,0);
printf("The file size is:%ld\n",filesize);
fread(&file_data,filesize,1,file_descriptor);
MAGIC_DATA_J = 0x7c2a5242;
Mem_filesize = filesize;
checksum_filedata = checkSum(&file_data);
checksum_MAGIC_DATA_J = checkSum(&MAGIC_DATA_J,0xc);
MAGIC_DATA_D._0_4_ = (__sighandler_t)0x7c2a5260;
usb_if_transfer = (int *)libusb_alloc_transfer(0);
```



Unlock Transceiver Bootloader

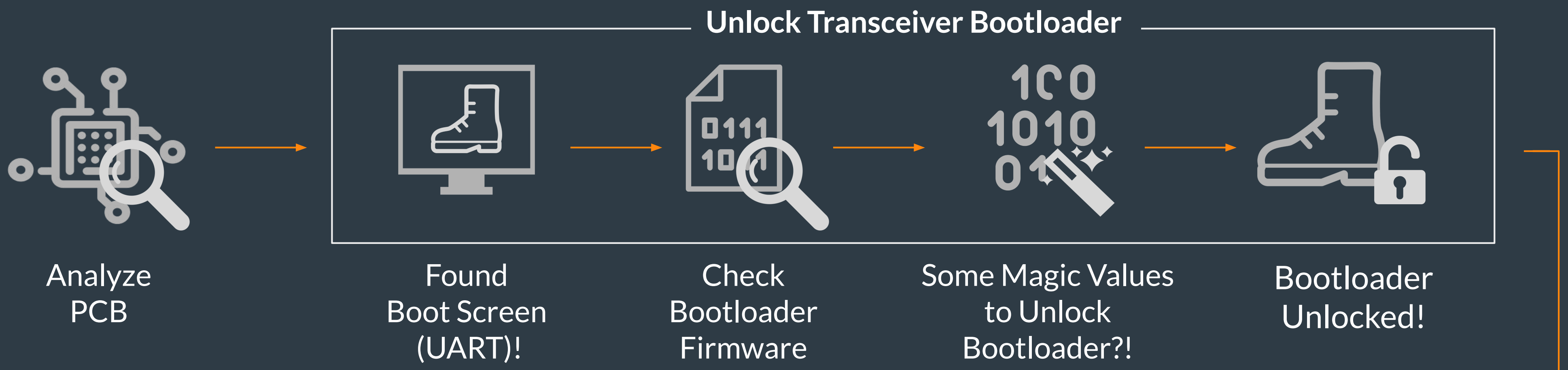
Analyze PCB

Found Boot Screen (UART)!

Check Bootloader Firmware

Some Magic Values to Unlock Bootloader?!

Bootloader Unlocked!



Unlock Transceiver Bootloader

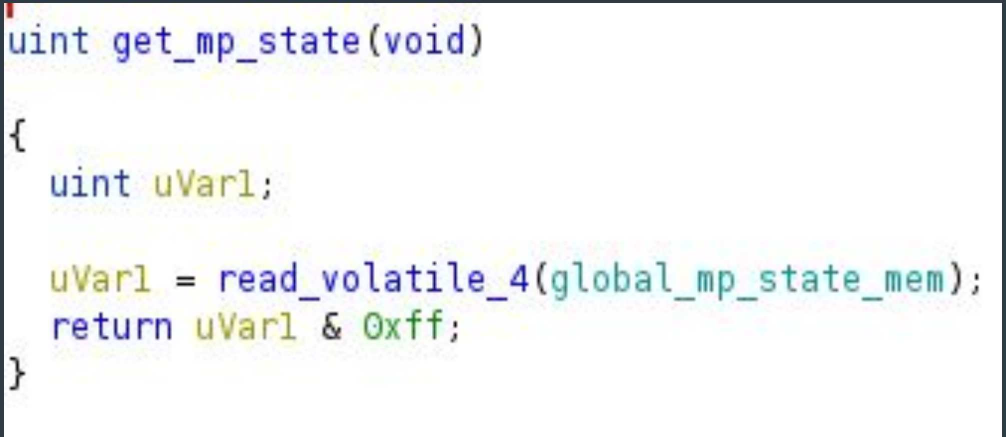
Analyze
PCB

Found
Boot Screen
(UART)!

Check
Bootloader
Firmware

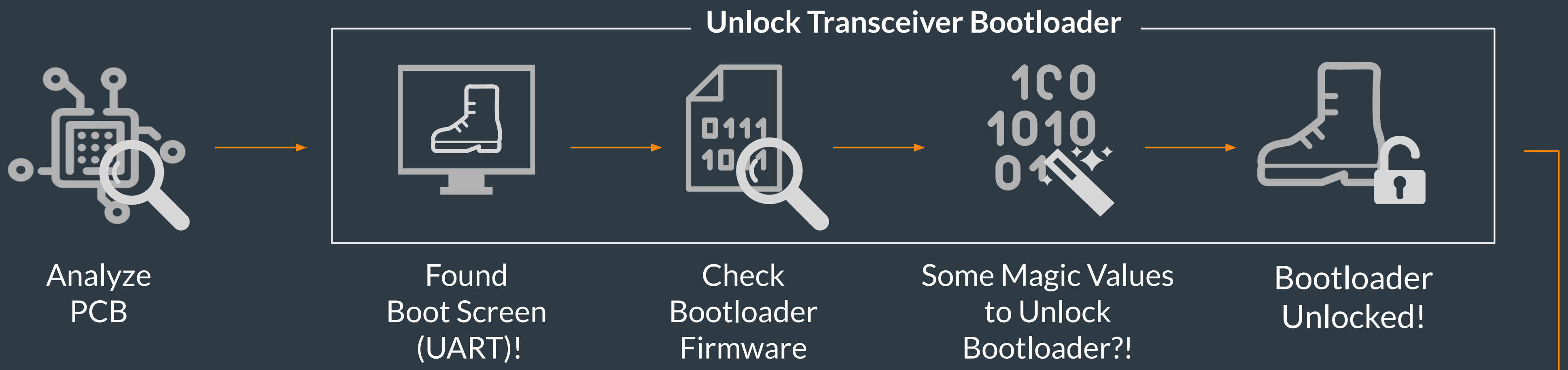
Some Magic Values
to Unlock
Bootloader?!

Bootloader
Unlocked!



```
uint get_mp_state(void)

{
  uint uVar1;

  uVar1 = read_volatile_4(global_mp_state_mem);
  return uVar1 & 0xff;
}
```

Modify
Firmware



Unlock Transceiver Bootloader

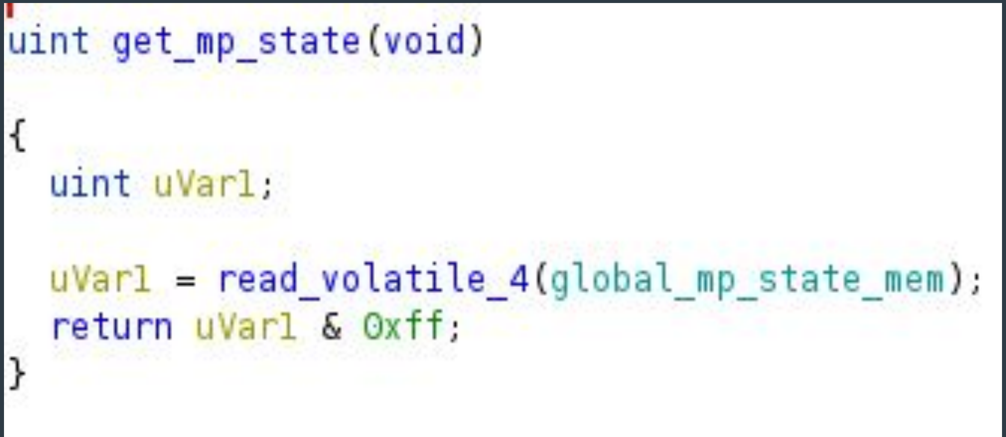
Analyze PCB

Found Boot Screen (UART)!

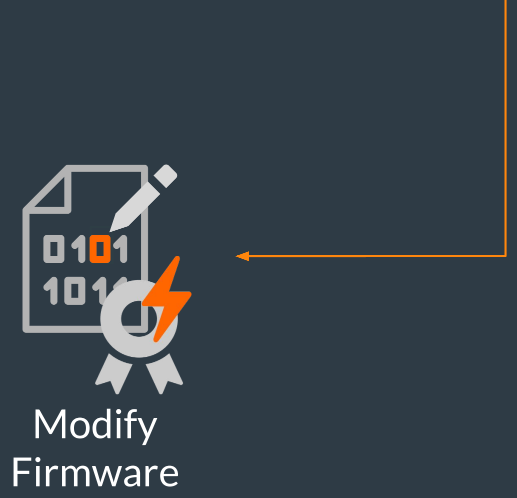
Check Bootloader Firmware

Some Magic Values to Unlock Bootloader?!

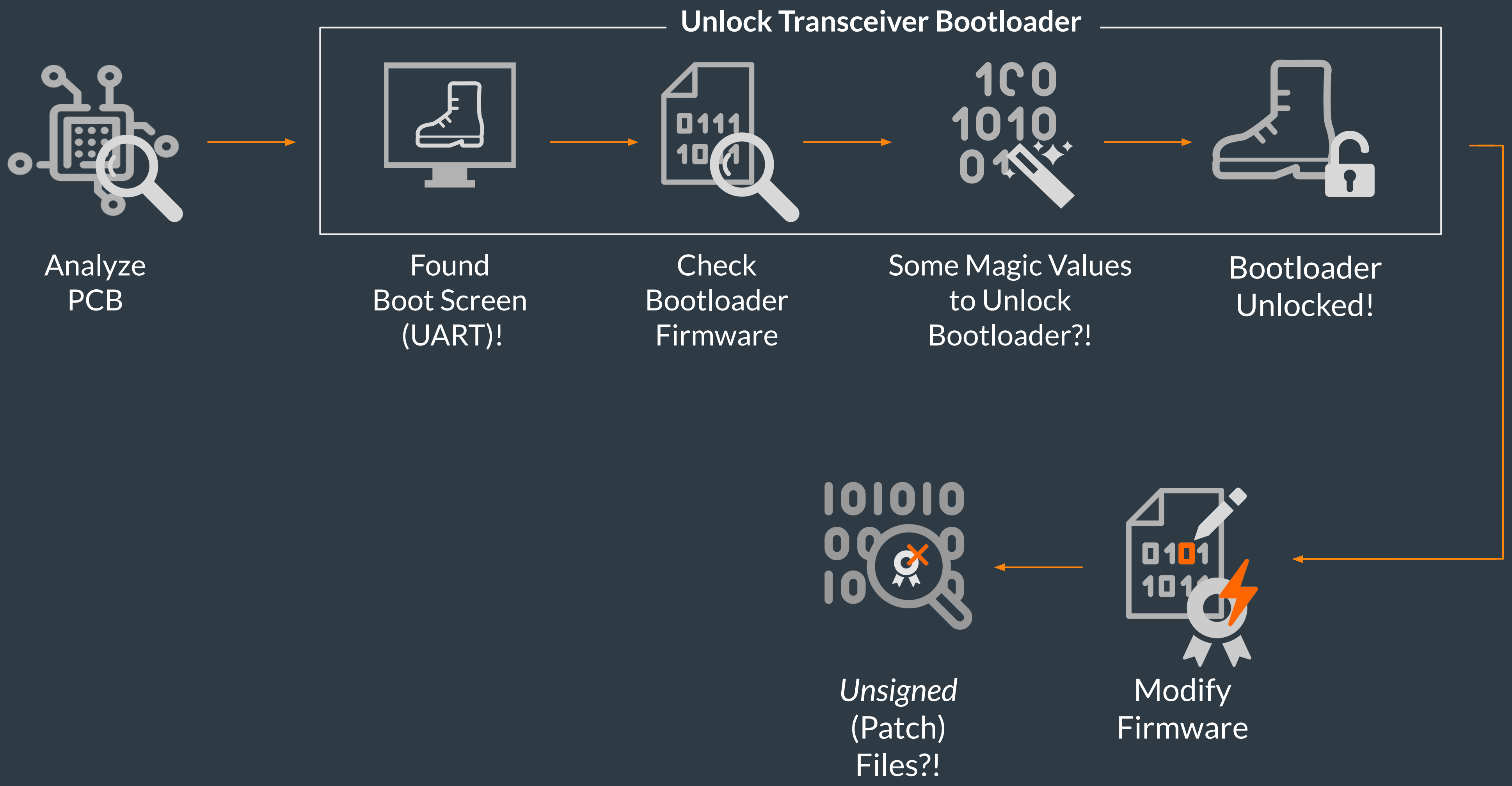
Bootloader Unlocked!



```
uint get_mp_state(void)

{
  uint uVar1;

  uVar1 = read_volatile_4(global_mp_state_mem);
  return uVar1 & 0xff;
}
```



Modify Firmware



Unlock Transceiver Bootloader

Analyze
PCB

Found
Boot Screen
(UART)!

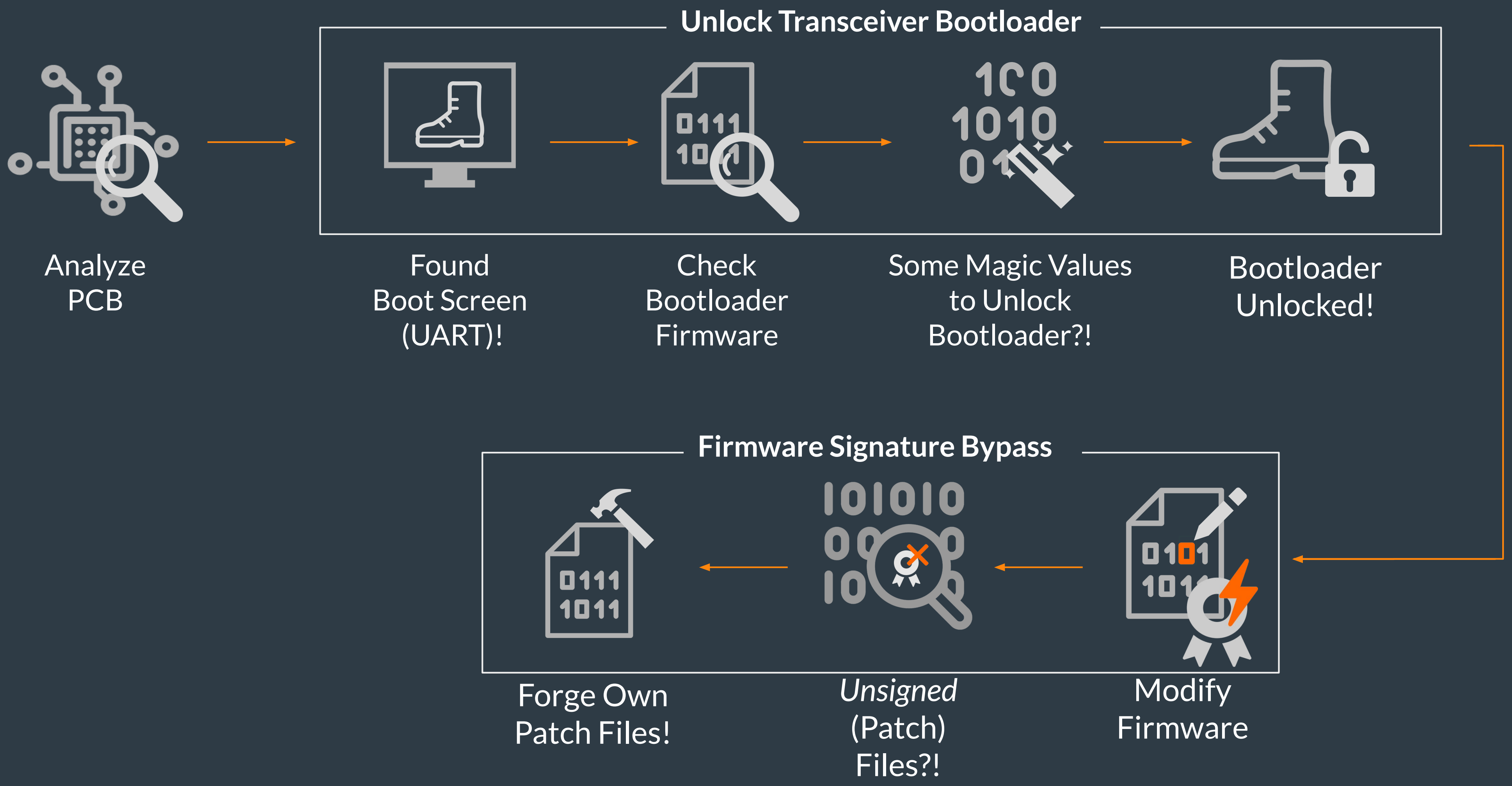
Check
Bootloader
Firmware

Some Magic Values
to Unlock
Bootloader?!

Bootloader
Unlocked!

Modify
Firmware

Unsigned
(Patch)
Files?!



Unlock Transceiver Bootloader

Analyze PCB

Found Boot Screen (UART)!

Check Bootloader Firmware

Some Magic Values to Unlock Bootloader?!

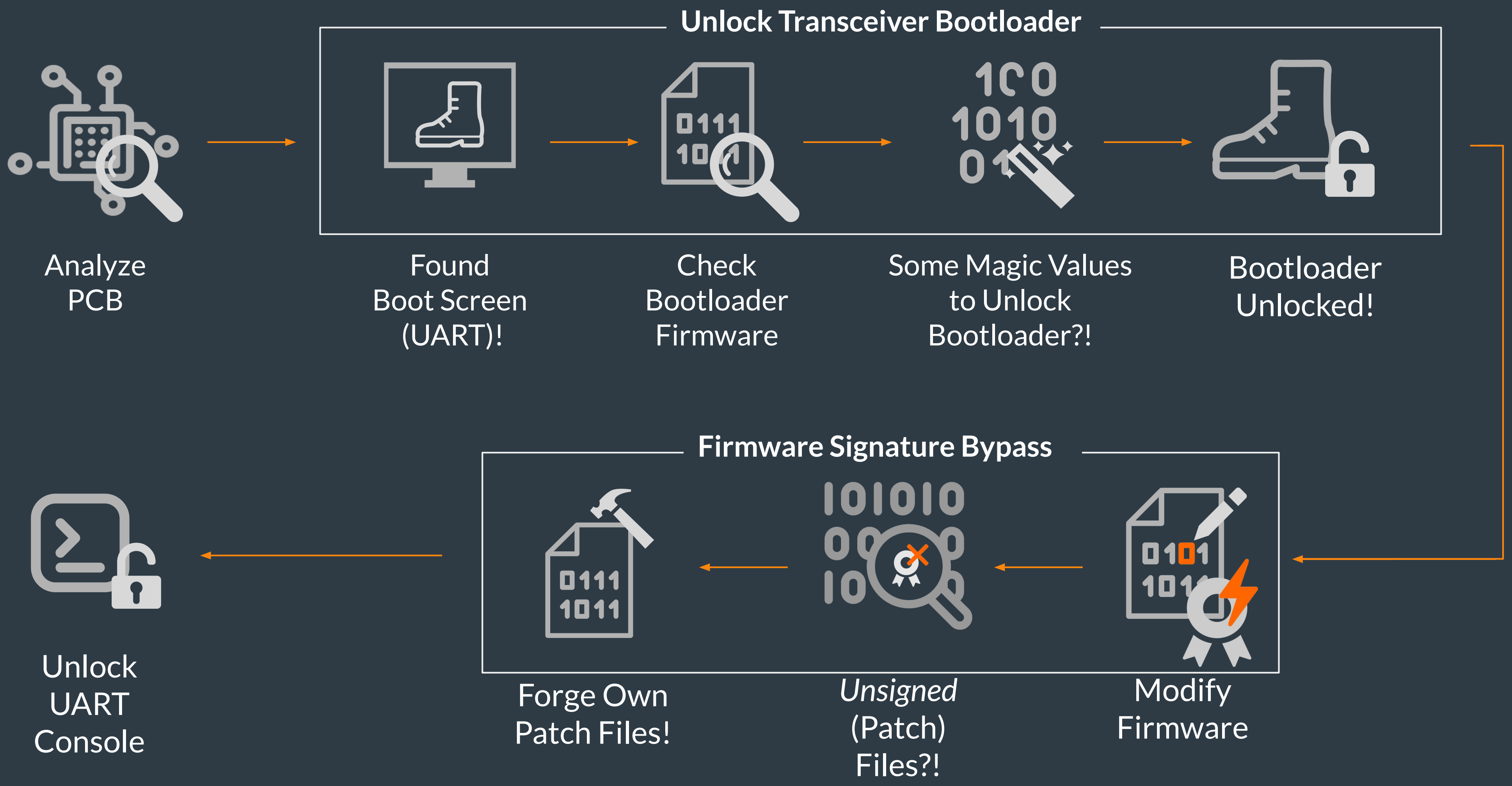
Bootloader Unlocked!

Firmware Signature Bypass

Modify Firmware

Unsigned (Patch) Files?!

Forge Own Patch Files!



Analyze PCB

Unlock Transceiver Bootloader

Found Boot Screen (UART)!

Check Bootloader Firmware

Some Magic Values to Unlock Bootloader?!

Bootloader Unlocked!

Firmware Signature Bypass

Modify Firmware

Unsigned (Patch) Files?!

Forge Own Patch Files!

Unlock UART Console

Summary: Static Analysis

- Full control over the transceiver SoC -> next target: main SoC

- **Static analysis was key for all other steps**
  - For example, when reversing the signal:
    - We needed seeds hidden in the firmware
    - Confirm DroneID packet structure

Wireless Physical Layer

Reversing DJI DroneID

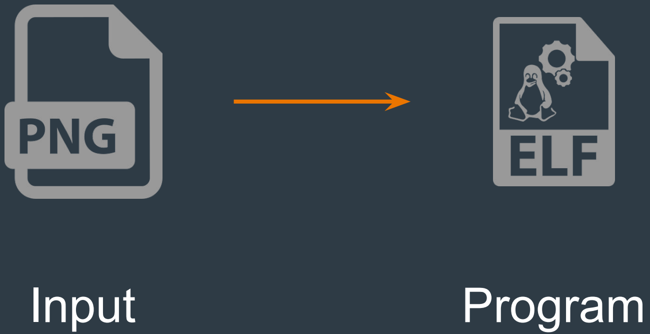
Static Analysis

Hands on the Drone

Dynamic Analysis

Fuzzing Drones for Pain and Profit

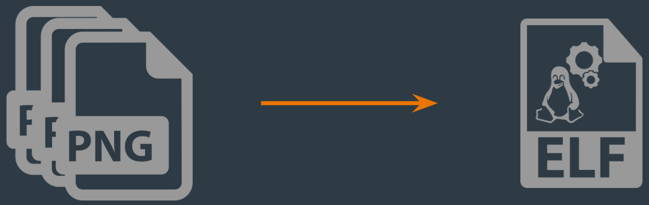
What is Fuzzing?



Input

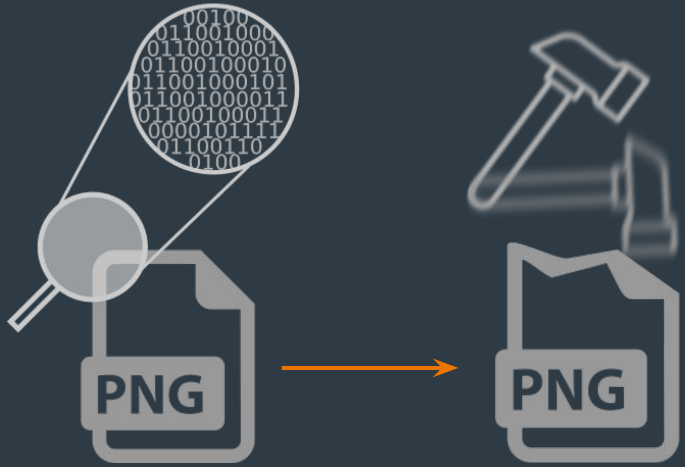
Program

What is Fuzzing?

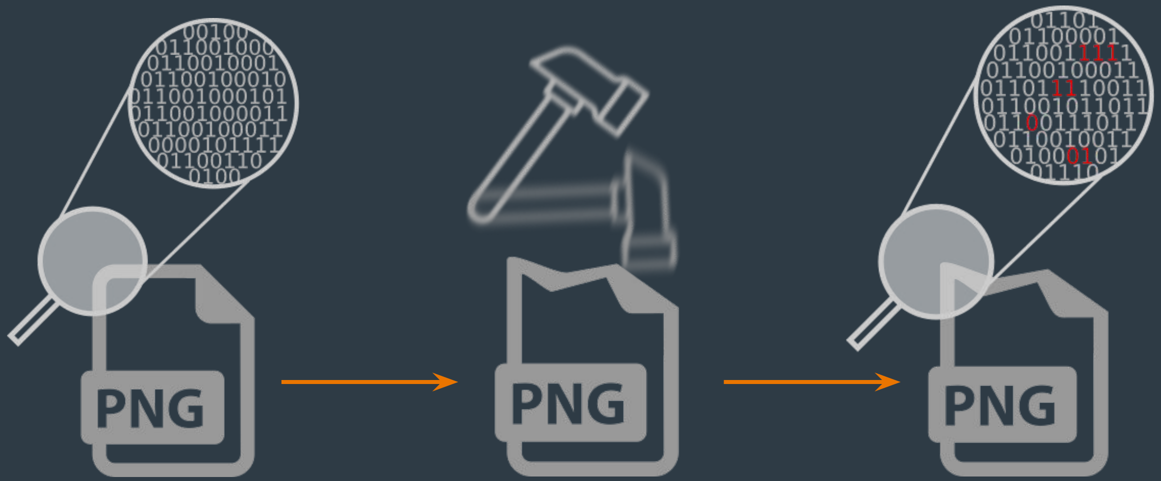


What is Fuzzing?

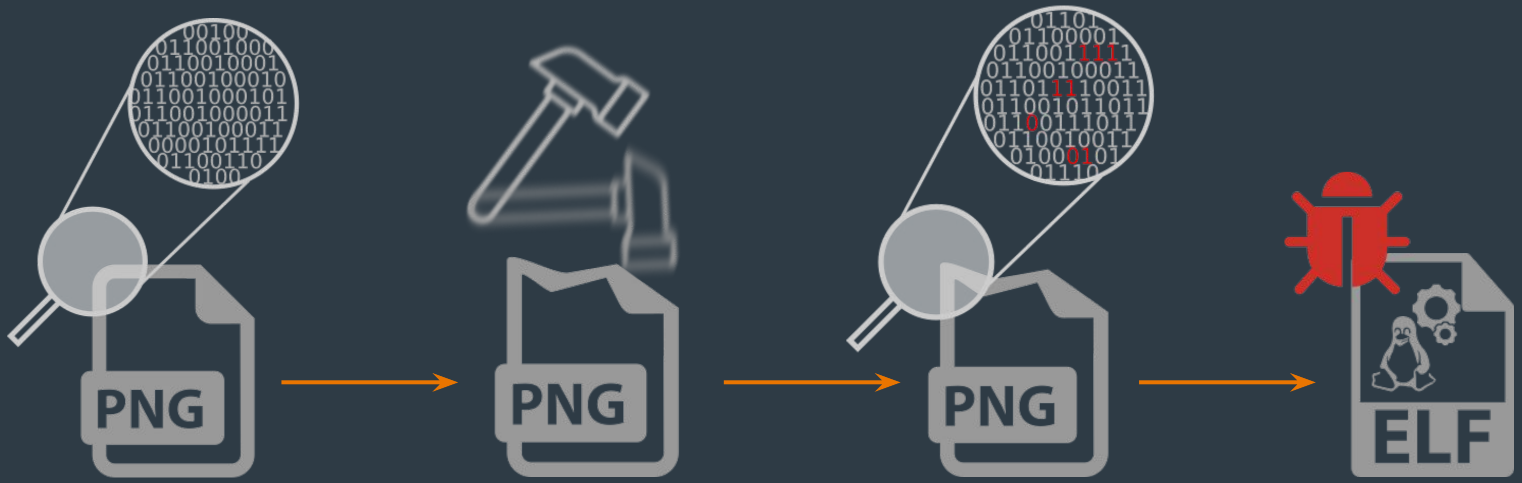
What is Fuzzing?



What is Fuzzing?



What is Fuzzing?



BUT

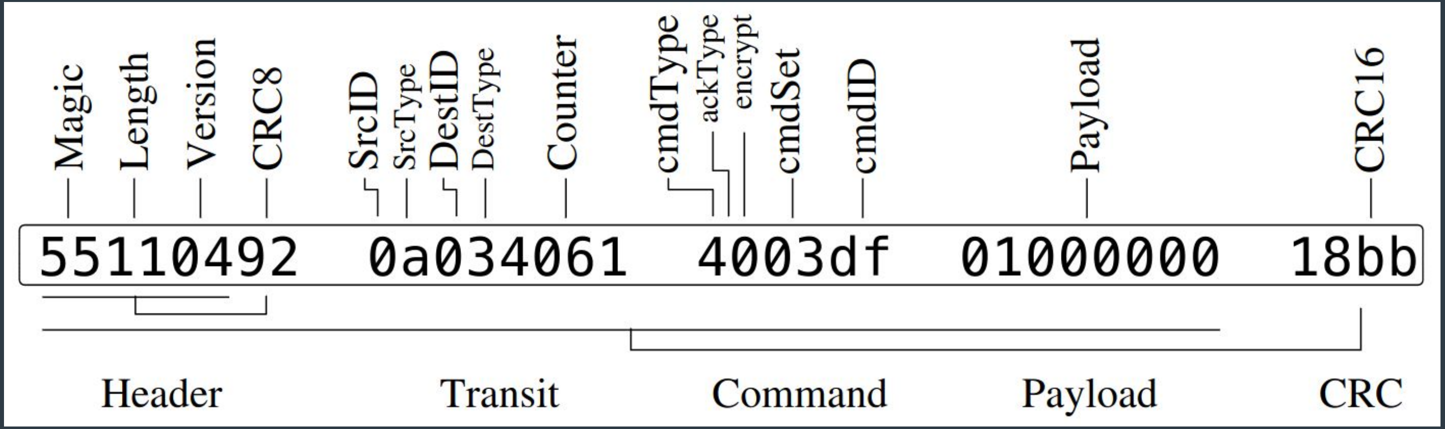
How to Fuzz Drones?

Problems:

- drone != a single binary
  - complex firmware (multiple SoC’s, different OSes)
  - hard to emulate
- no source code we could instrument

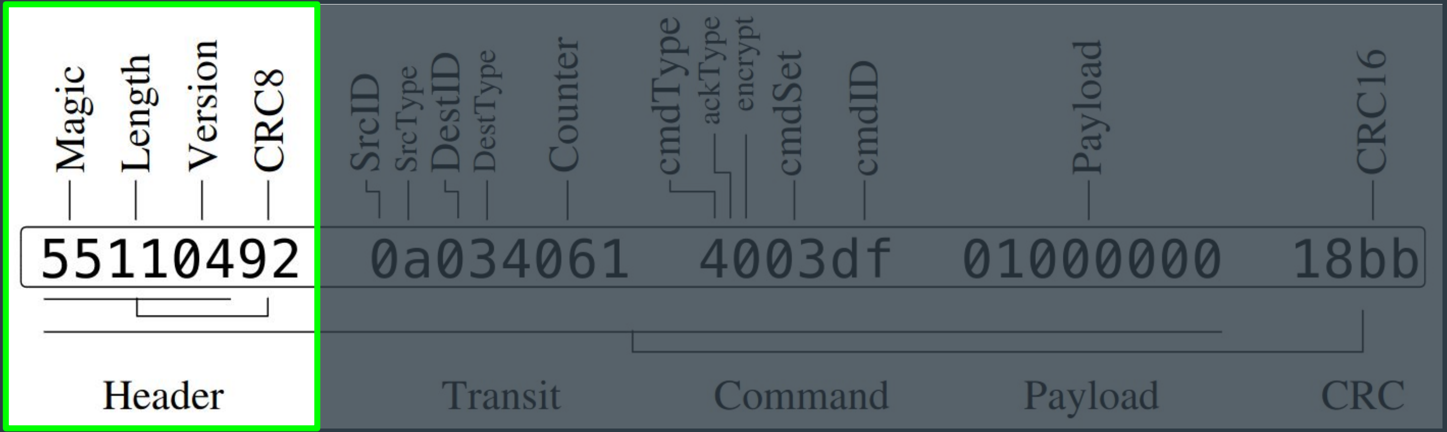
=> no easy off-the-shelf fuzzing solution available

Idea: Let’s target communication protocol

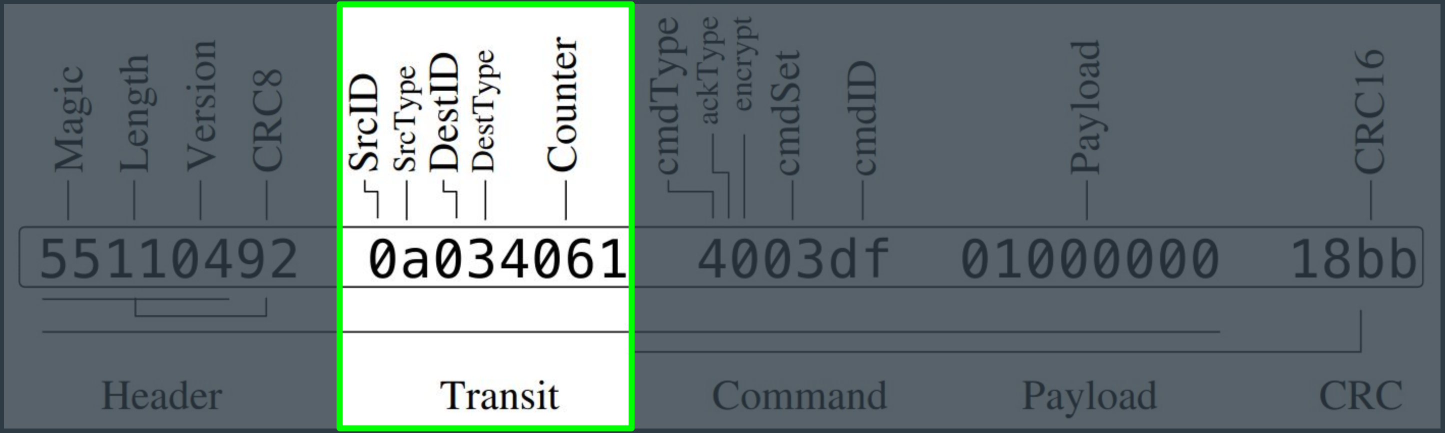


DJI DUML Protocol

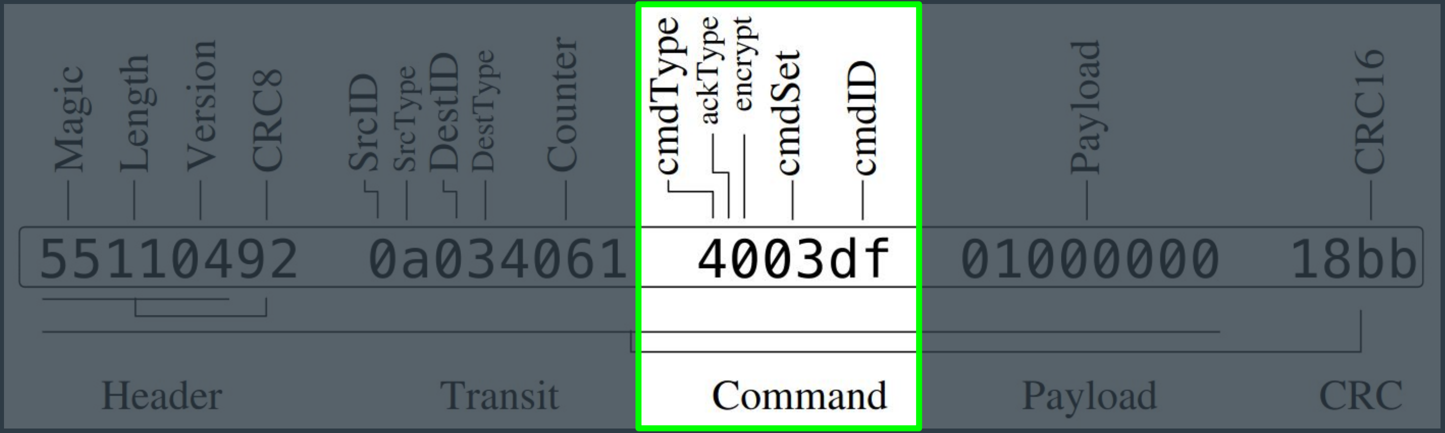
How to Fuzz Drones – DJI DUML Protocol



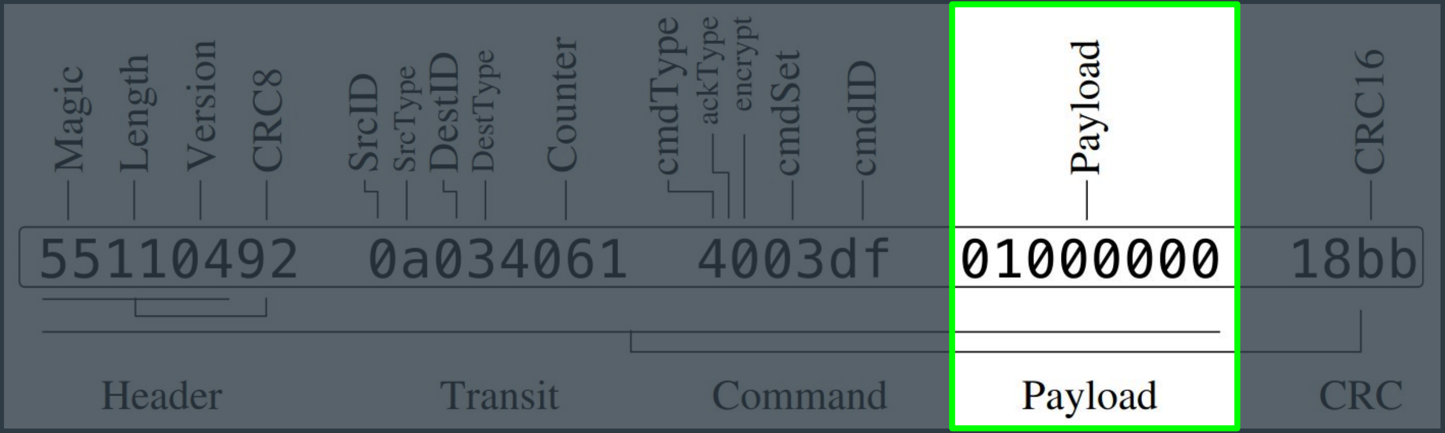
How to Fuzz Drones – DJI DUML Protocol



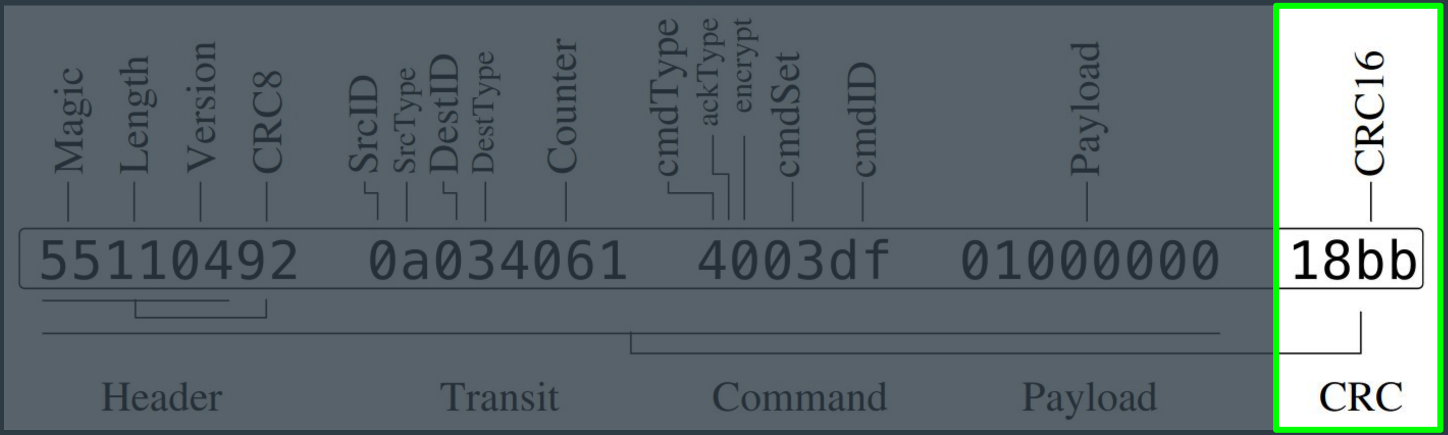
How to Fuzz Drones – DJI DUML Protocol



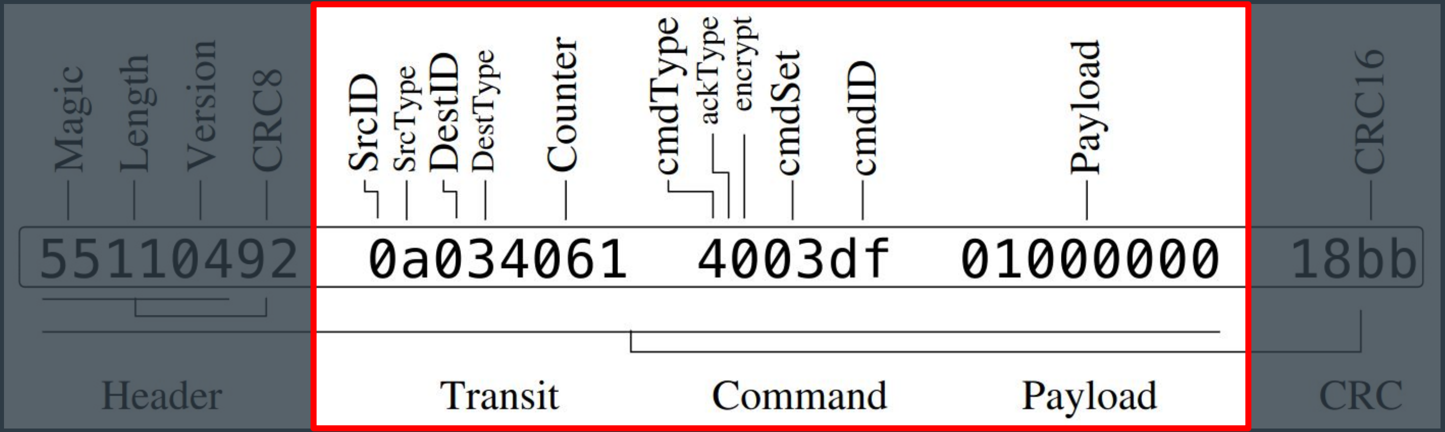
How to Fuzz Drones – DJI DUML Protocol



How to Fuzz Drones – DJI DUML Protocol



How to Fuzz Drones – DJI DUML Protocol



How to Fuzz Drones?

How to Fuzz Drones?

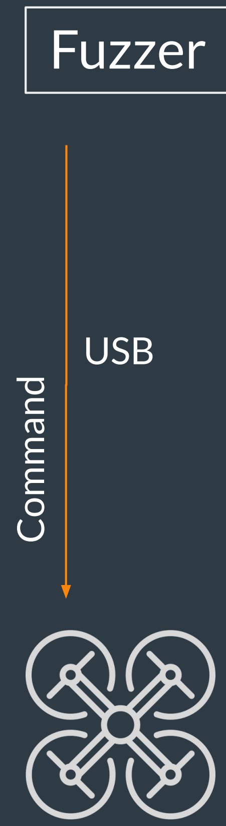
Prerequisites:

- A drone and fuzzer

How to Fuzz Drones?

Prerequisites:

- A drone and fuzzer
- Protocol knowledge


Fuzzer

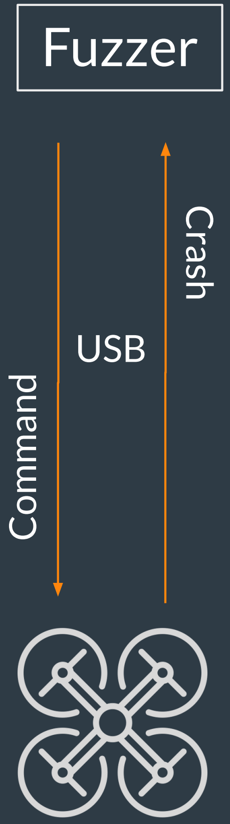
USB

Command

How to Fuzz Drones?

Prerequisites:

- A drone and fuzzer
- Protocol knowledge
- Bug oracle



Fuzzer

Command

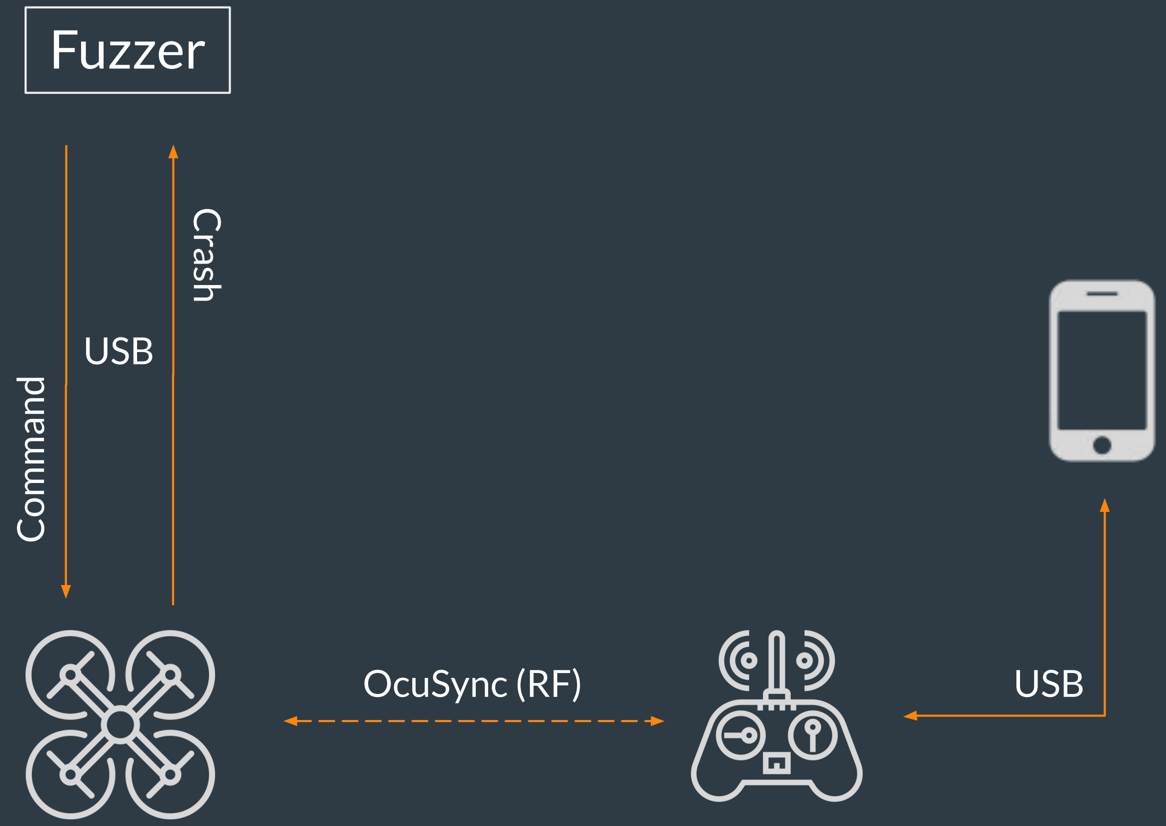
USB

Crash

How to Fuzz Drones?

Prerequisites:

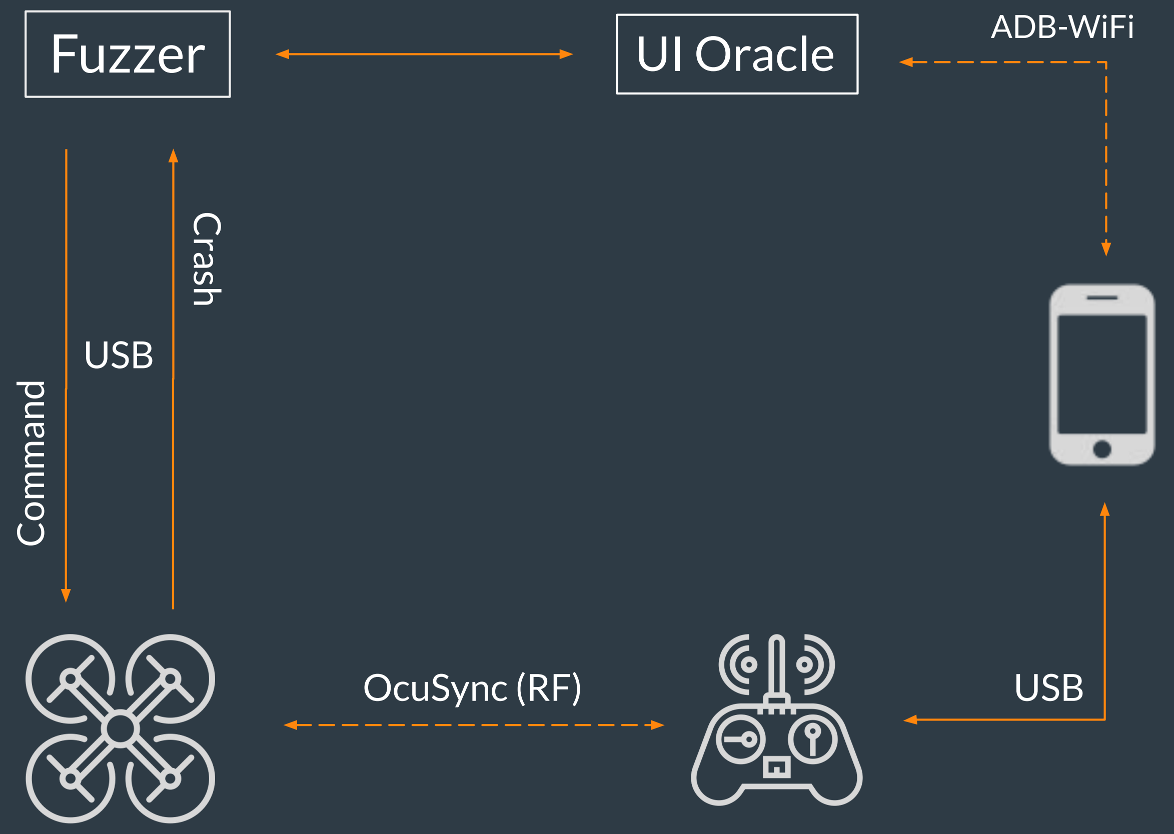
- A drone and fuzzer
- Protocol knowledge
- Bug oracle



How to Fuzz Drones?

Prerequisites:

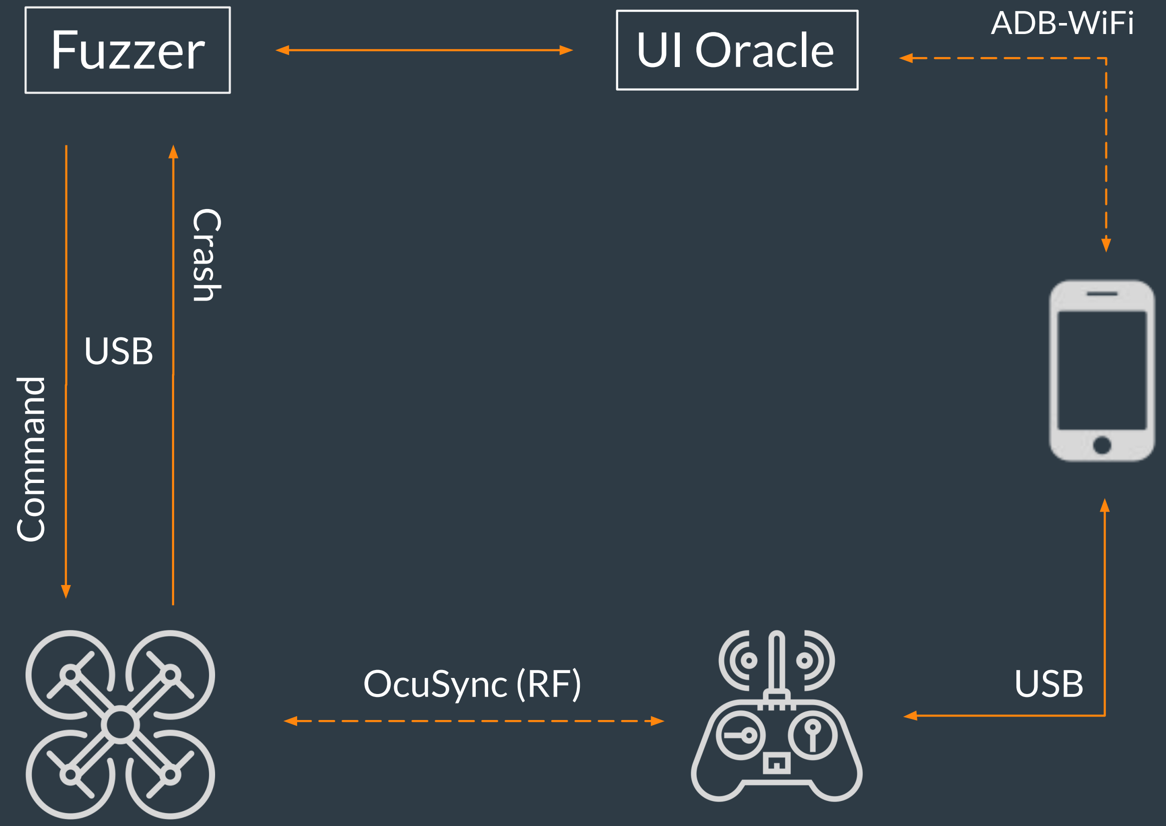
- A drone and fuzzer
- Protocol knowledge
- Bug oracle



How to Fuzz Drones?

Prerequisites:

- A drone and fuzzer
- Protocol knowledge
- Bug oracle

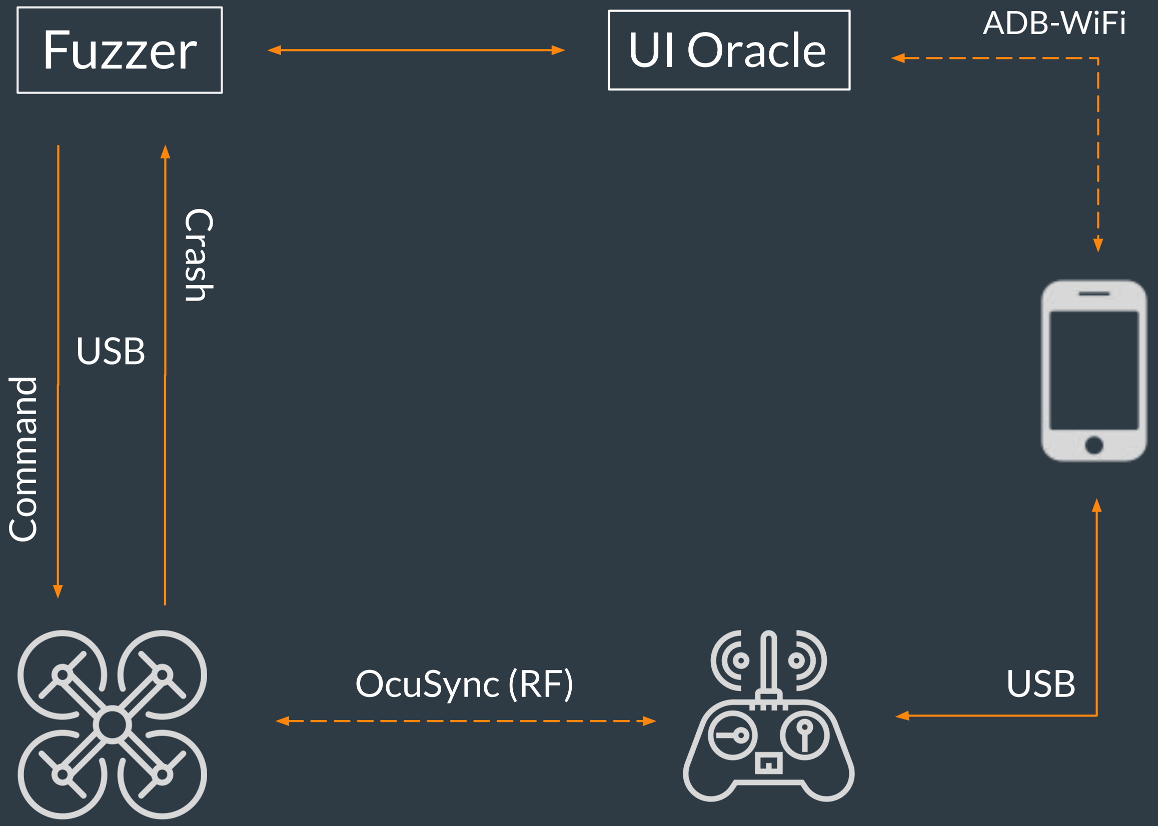


How to Fuzz Drones?

Prerequisites:

- A drone and fuzzer
- Protocol knowledge
- Bug oracle

Reproducible bugs!



Does fuzzing work?

ID	Oracle	Component	Observable Behavior	Classification	Severity	Remote	Vulnerable Devices
#1	ADB check	`dji_sys` binary	ADB started (root access)	arbitrary code exec	mid	✗	Mini 2
#2	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#3	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#4	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#5	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#6	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#7	crash	flight controller	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#8	crash	flight controller	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#9	crash	unknown	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#10	crash	unknown	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#11	crash	unknown	critical error (drone reboot)	denial of service	low	✓	Mini 2
#12	crash	unknown	critical error (drone reboot)	denial of service	low	✓	Mini 2
#13	crash	flight controller	critical error (drone reboot)	denial of service	low	✓	Mavic Air 2
#14	UI change	WiFi chip	change SSID	arbitrary code exec	mid	✓	Mini 2, Mavic 3
#15	UI change	flight controller	change serial number	identity spoofing	mid	✓	Mini 2

*Following responsible disclosure, DJI fixed these bugs.

Does fuzzing work?

ID	Oracle	Component	Observable Behavior	Classification	Severity	Remote	Vulnerable Devices
#1	ADB check	`dji_sys` binary	ADB started (root access)	arbitrary code exec	mid	✗	Mini 2
#2	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#3	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#4	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#5	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#6	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#7	crash	flight controller	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#8	crash	flight controller	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#9	crash	unknown	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#10	crash	unknown	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#11	crash	unknown	critical error (drone reboot)	denial of service	low	✓	Mini 2
#12	crash	unknown	critical error (drone reboot)	denial of service	low	✓	Mini 2
#13	crash	flight controller	critical error (drone reboot)	denial of service	low	✓	Mavic Air 2
#14	UI change	WiFi chip	change SSID	arbitrary code exec	mid	✓	Mini 2, Mavic 3
#15	UI change	flight controller	change serial number	identity spoofing	mid	✓	Mini 2

*Following responsible disclosure, DJI fixed these bugs.

Arbitrary Code Execution

- found by UI oracle: fuzzer changed an immutable value
- missing sanitization of user-controllable input

=> Linux command injection

Arbitrary Code Execution

Goal: root privileges -> start adb server

Problem: command length limited to max 32 characters

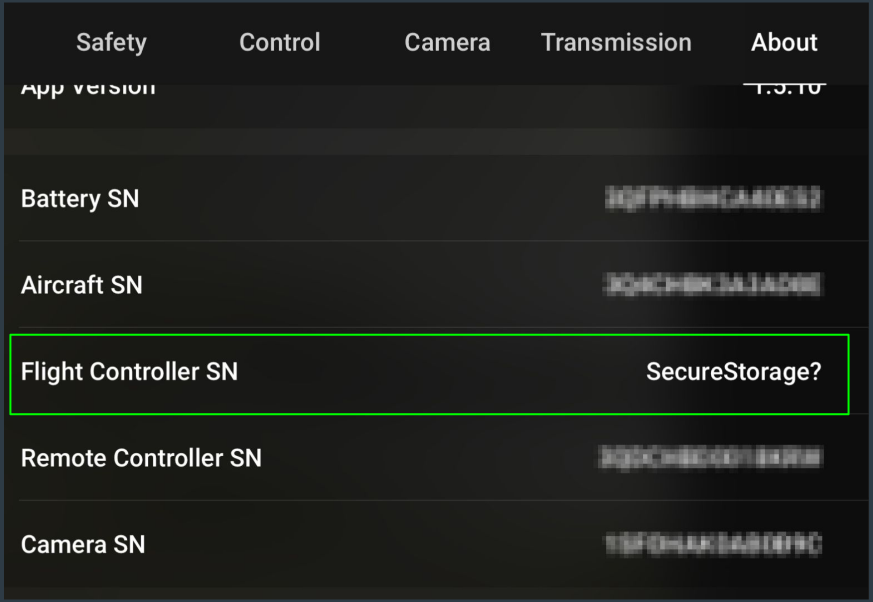
=> transfer exploit script chunkwise

Does fuzzing work?

ID	Oracle	Component	Observable Behavior	Classification	Severity	Remote	Vulnerable Devices
#1	ADB check	`dji_sys` binary	ADB started (root access)	arbitrary code exec	mid	✗	Mini 2
#2	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#3	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#4	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#5	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#6	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#7	crash	flight controller	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#8	crash	flight controller	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#9	crash	unknown	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#10	crash	unknown	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#11	crash	unknown	critical error (drone reboot)	denial of service	low	✓	Mini 2
#12	crash	unknown	critical error (drone reboot)	denial of service	low	✓	Mini 2
#13	crash	flight controller	critical error (drone reboot)	denial of service	low	✓	Mavic Air 2
#14	UI change	WiFi chip	change SSID	arbitrary code exec	mid	✓	Mini 2, Mavic 3
#15	UI change	flight controller	change serial number	identity spoofing	mid	✓	Mini 2

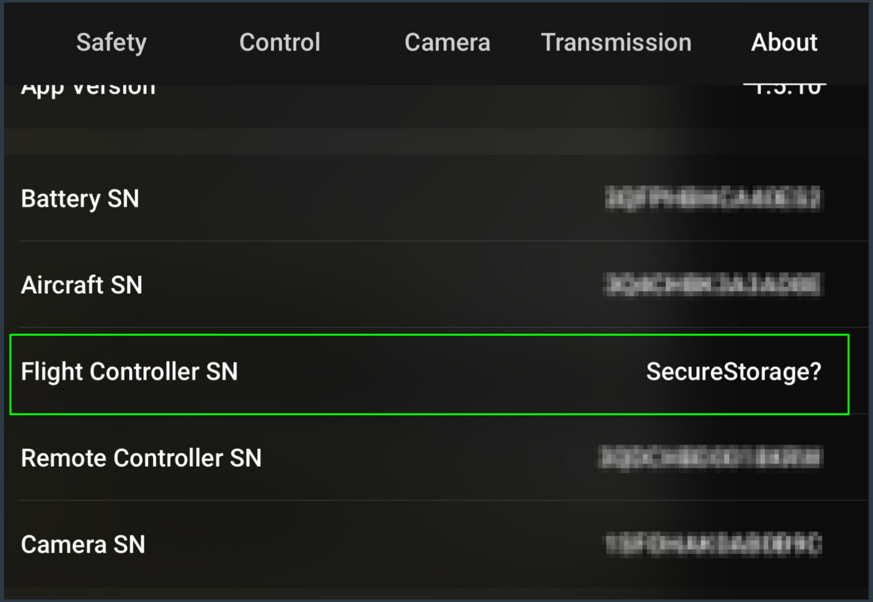
*Following responsible disclosure, DJI fixed these bugs.

Change Immutable Serial Number



App version	1.5.10
Battery SN	[redacted]
Aircraft SN	[redacted]
Flight Controller SN	SecureStorage?
Remote Controller SN	[redacted]
Camera SN	[redacted]

Change Immutable Serial Number



App version 1.5.10

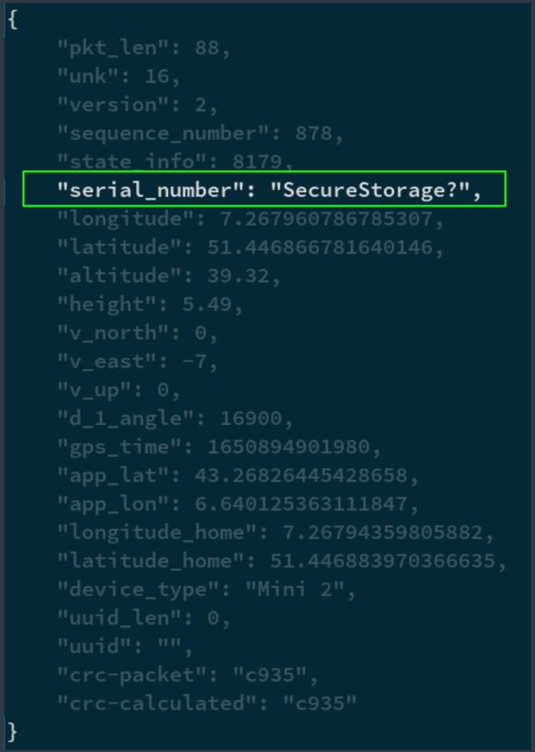
Battery SN [redacted]

Aircraft SN [redacted]

Flight Controller SN SecureStorage?

Remote Controller SN [redacted]

Camera SN [redacted]



```
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_number": 878,
    "state_info": 8179,
    "serial_number": "SecureStorage?",
    "longitude": 7.267960786785307,
    "latitude": 51.446866781640146,
    "altitude": 39.32,
    "height": 5.49,
    "v_north": 0,
    "v_east": -7,
    "v_up": 0,
    "d_1_angle": 16900,
    "gps_time": 1650894901980,
    "app_lat": 43.26826445428658,
    "app_lon": 6.640125363111847,
    "longitude_home": 7.26794359805882,
    "latitude_home": 51.446883970366635,
    "device_type": "Mini 2",
    "uuid_len": 0,
    "uuid": "",
    "crc-packet": "c935",
    "crc-calculated": "c935"
}
```

Does fuzzing work?

ID	Oracle	Component	Observable Behavior	Classification	Severity	Remote	Vulnerable Devices
#1	ADB check	`dji_sys` binary	ADB started (root access)	arbitrary code exec	mid	✗	Mini 2
#2	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#3	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#4	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#5	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#6	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#7	crash	flight controller	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#8	crash	flight controller	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#9	crash	unknown	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#10	crash	unknown	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#11	crash	unknown	critical error (drone reboot)	denial of service	low	✓	Mini 2
#12	crash	unknown	critical error (drone reboot)	denial of service	low	✓	Mini 2
#13	crash	flight controller	critical error (drone reboot)	denial of service	low	✓	Mavic Air 2
#14	UI change	WiFi chip	change SSID	arbitrary code exec	mid	✓	Mini 2, Mavic 3
#15	UI change	flight controller	change serial number	identity spoofing	mid	✓	Mini 2

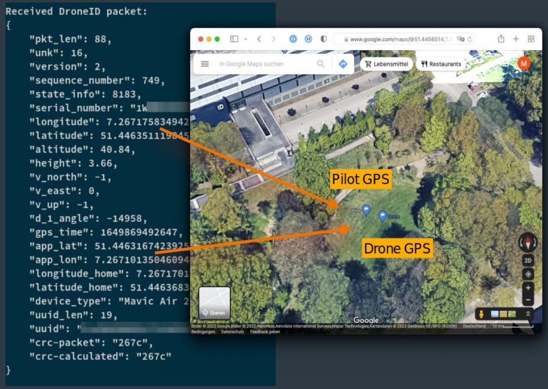
*Following responsible disclosure, DJI fixed these bugs.

Summary: Fuzzing the Drone

- Fuzzing on hardware: Slow & painful but real bugs

- Tailor fuzzer to your target, for example, custom oracles!

Recap: How to analyze drones

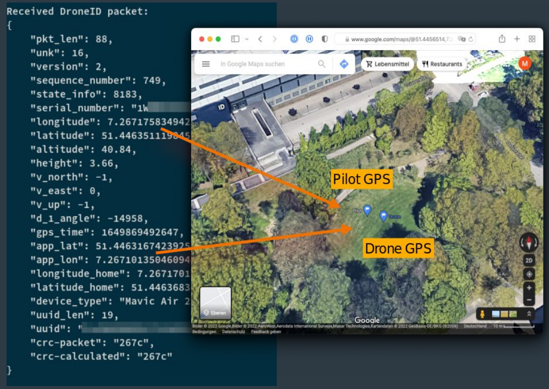


```
Received DroneID packet:
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_number": 749,
    "state_info": 8183,
    "serial_number": "1W[redacted]
    "longitude": 7.267175834942
    "latitude": 51.446351119845
    "altitude": 40.84,
    "height": 3.66,
    "v_north": -1,
    "v_east": 0,
    "v_up": -1,
    "d_1_angle": -14958,
    "gps_time": 1649869492647,
    "app_lat": 51.446316742392
    "app_lon": 7.26710135046094
    "longitude_home": 7.2671701
    "latitude_home": 51.4463683
    "device_type": "Mavic Air 2
    "uuid_len": 19,
    "uuid": "[redacted]
    "crc-packet": "267c",
    "crc-calculated": "267c"
}
```

Drone and pilot’s location tracking

Wireless Analysis

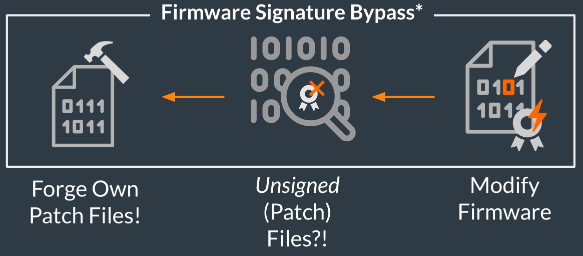
Recap: How to analyze drones



```
Received DroneID packet:
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_number": 749,
    "state_info": 8183,
    "serial_number": "1W[redacted]
    "longitude": 7.267175834942
    "latitude": 51.446351119845
    "altitude": 40.84,
    "height": 3.66,
    "v_north": -1,
    "v_east": 0,
    "v_up": -1,
    "d_1_angle": -14958,
    "gps_time": 1649869492647,
    "app_lat": 51.4463167423925
    "app_lon": 7.26710135046094
    "longitude_home": 7.2671701
    "latitude_home": 51.4463683
    "device_type": "Mavic Air 2
    "uuid_len": 19,
    "uuid": "[redacted]
    "crc-packet": "267c",
    "crc-calculated": "267c"
}
```

Drone and pilot's location tracking

Wireless Analysis



Firmware Signature Bypass*

Forge Own Patch Files!

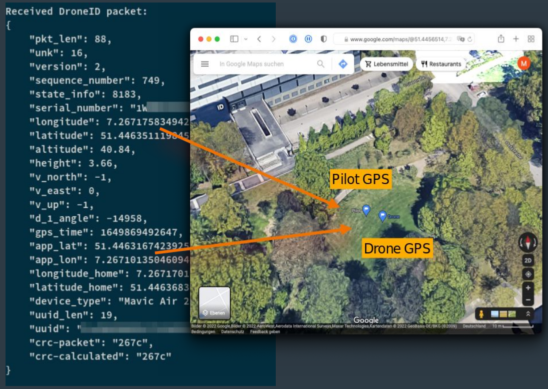
Unsigned (Patch) Files?!

Modify Firmware

Firmware signature verification bypass

Static Analysis

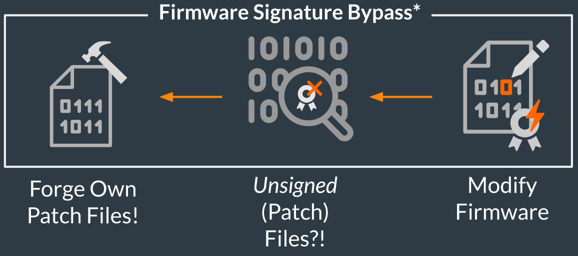
Recap: How to analyze drones



```
Received DroneID packet:
{
    "pkt_len": 88,
    "unk": 16,
    "version": 2,
    "sequence_number": 749,
    "state_info": 8183,
    "serial_number": "1W[redacted]
    "longitude": 7.267175834942
    "latitude": 51.446351119845
    "altitude": 40.84,
    "height": 3.66,
    "v_north": -1,
    "v_east": 0,
    "v_up": -1,
    "d_1_angle": -14958,
    "gps_time": 1649869492647,
    "app_lat": 51.4463167423925
    "app_lon": 7.26710135046094
    "longitude_home": 7.2671701
    "latitude_home": 51.4463683
    "device_type": "Mavic Air 2
    "uuid_len": 19,
    "uuid": "[redacted]
    "crc-packet": "267c",
    "crc-calculated": "267c"
}
```

Drone and pilot’s location tracking

Wireless Analysis



Firmware signature verification bypass

Static Analysis

ID	Oracle	Component	Observable Behavior	Classification	Severity	Remote	Vulnerable Devices
#1	ADB check	`dji_sys` binary	ADB started (root access)	arbitrary code exec	mid	✗	Mini 2
#2	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#3	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#4	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#5	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#6	crash	flight controller	critical error (drone reboot)	buffer overflow	mid	✓	Mavic Air 2
#7	crash	flight controller	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#8	crash	flight controller	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#9	crash	unknown	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#10	crash	unknown	critical error (drone reboot)	denial of service	mid	✓	Mini 2
#11	crash	unknown	critical error (drone reboot)	denial of service	low	✓	Mini 2
#12	crash	unknown	critical error (drone reboot)	denial of service	low	✓	Mini 2
#13	crash	flight controller	critical error (drone reboot)	denial of service	low	✓	Mavic Air 2
#14	UI change	WiFi chip	change SSID	arbitrary code exec	mid	✓	Mini 2, Mavic 3
#15	UI change	flight controller	change serial number	identity spoofing	mid	✓	Mini 2

Vulnerability detection via fuzzing

Dynamic Analysis

Takeaways

- Holistic approach (analysis of different components/layers) needed
- Hardware-in-the-loop fuzzing is difficult but rewarding
- Countermeasures seem to be insufficient

Takeaways

- Holistic approach (analysis of different components/layers) needed
- Hardware-in-the-loop fuzzing is difficult but rewarding
- Countermeasures seem to be insufficient

RUB-SysSec/DroneSecurity

Nico: @74ck_0
Moritz: @m_u00d | https://mschloegel.me

Paper