ZERO NIGHTS 2018

2^3 EDITION

IoTSecFuzz

make your smart kettle afraid of you

2018.ZERONIGHTS.ORG

ZERO NIGHTS 2018

2^3 EDITION

Soff

2^3 EDITION

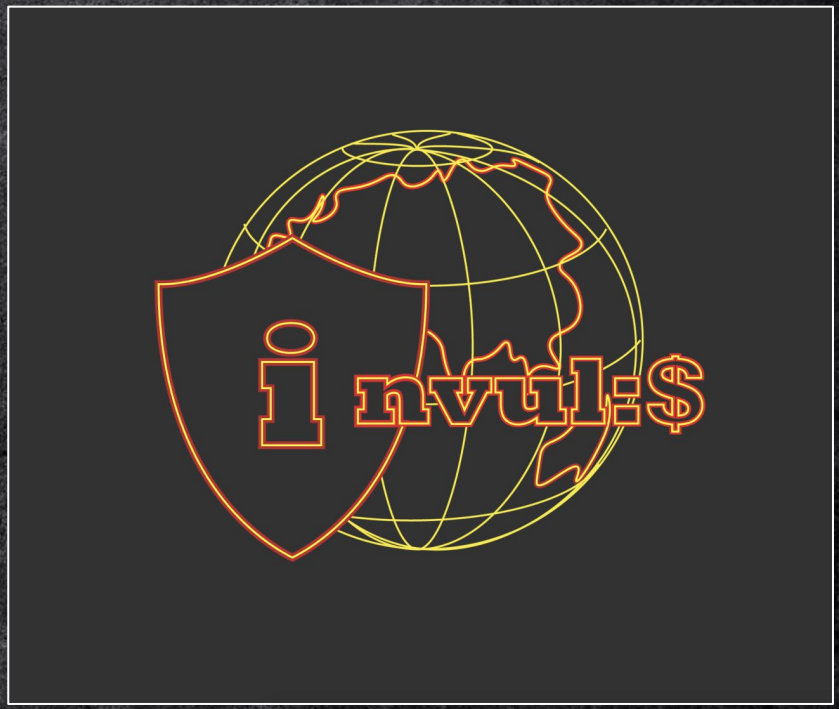
Who am I?

- HSE Miem Student 3
- InfoSec Pentester Intern
- Lunary & Invuls CTF teams player
- Ambassador of the laboratory Kaspersky
- Ex IOS developer

2³ EDITION

Who are we?

- International team
- 60+ participants
- PHDays StandOff 2017 participants.



ZERO NIGHTS 2018

2³ EDITION

What is IoT?

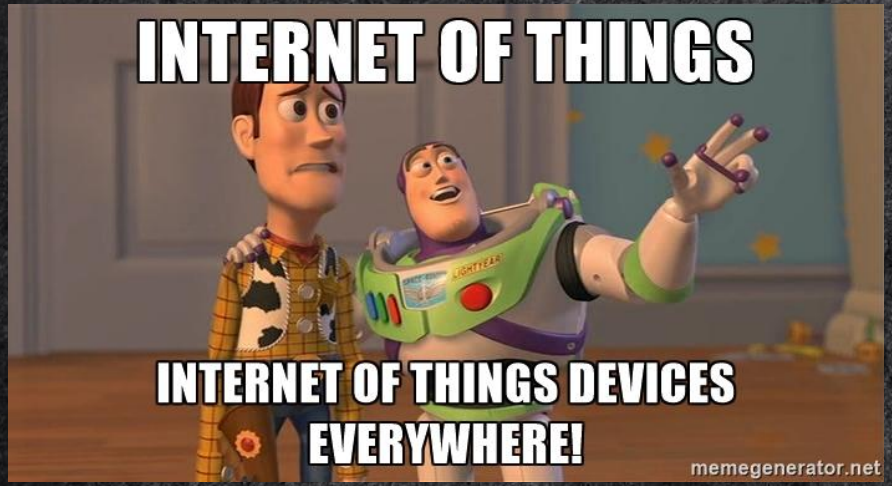
The Internet of Things - network of devices that are connected to the Internet, controlled through it, and can communicate with each other.

But sometimes device can be classified as part of internet of things without an internet connection.

ZERO NIGHTS 2018

2^3 EDITION

IoT security issues and risks



The total number of devices connected to the Internet is 23 billion with the prospect of increasing to 30 billion by 2020.

A significant part of the devices has security problems.

Ignoring these problems leads to the creation of botnets(Mirai, Satori) and leakage of personal data.

ZERO NIGHTS 2018

2³ EDITION

IoT security issues and risks

Due to the irresponsible approach towards IoT security, search engines for IoT have appeared:

- Shodan
- Censys
- ZoomEye

2³ EDITION

IoT security issues and risks

Violations of the principles of development:

- use of hardcoded and hidden service credentials
- use of the same keys and PIN codes
- lack of access control when accessing a known settings page
- incorrect processing of received data causing buffer overflow

Three main pillars

1. Hardware security
2. Software security
3. Radio security

(4. The fourth pillar Mobile security is also often singled out, but we will not consider it today.)

2^3 EDITION

Hardware security

Hardware security is :

- incorrect access to the board
- work with debug ports
- work with memory
- work with the bootloader

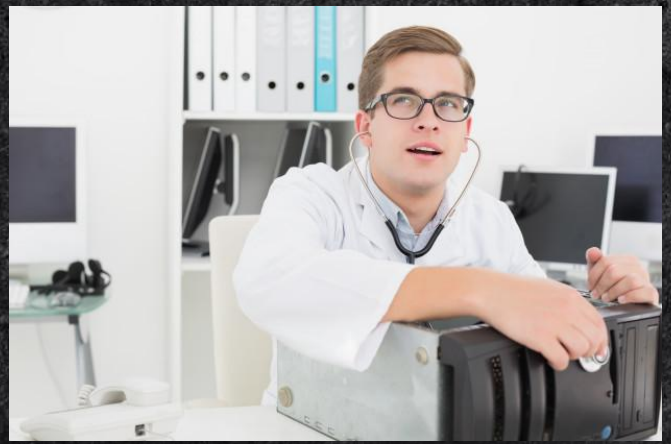
(to receive a memory dump or administrative shell)

2³ EDITION

Little more about software

Black Box

we can sniff output from ports

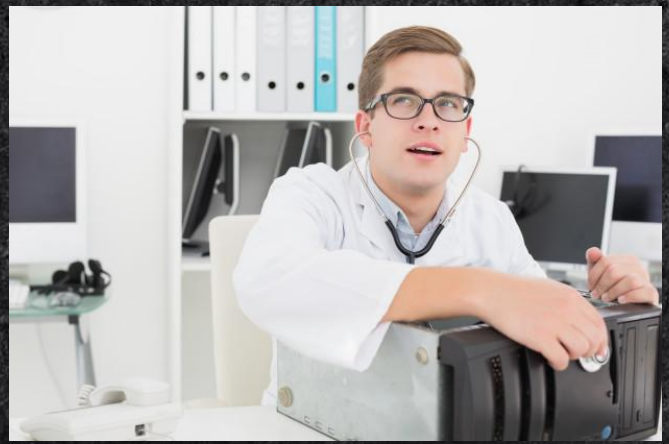


2^3 EDITION

Little more about software

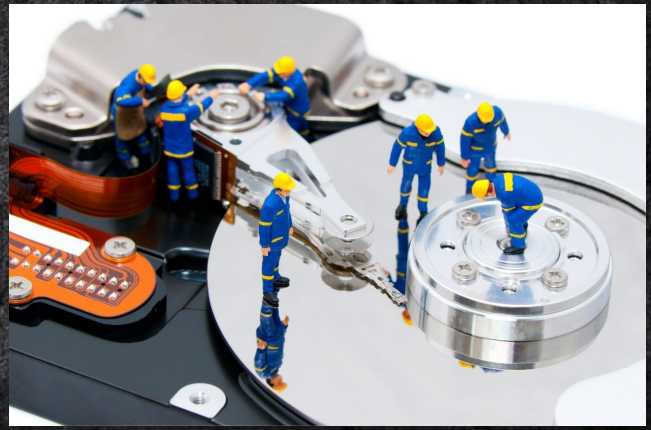
Black Box

we can sniff output from ports



White Box

we have a memory dump or firmware and we can analyze it



ZERO NIGHTS 2018

2^3 EDITION

The biggest problem

2^3 EDITION

The biggest problem

admin: admin

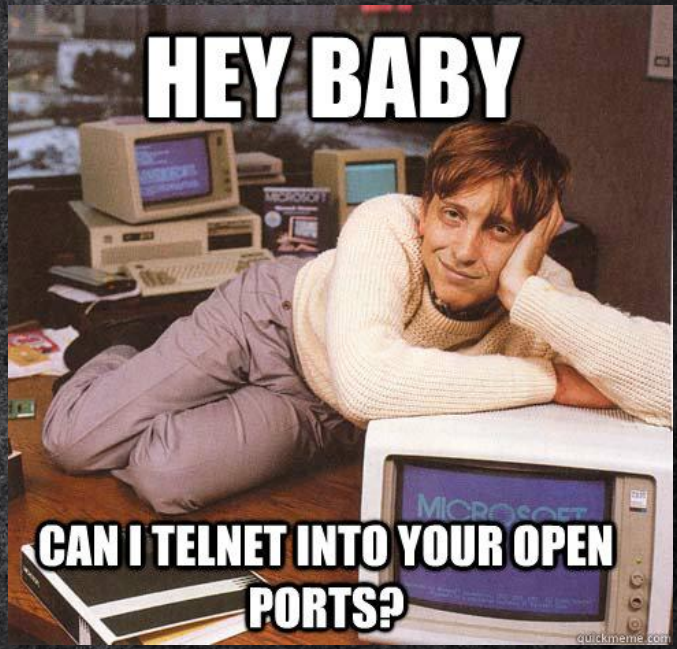
ZERO NIGHTS 2018

2^3 EDITION

Security of communication

Channels through which these devices communicate with other software and devices.

At this level, security holes are most often found.



2^3 EDITION

A little bit of everything

You need to know:

- TCP/IP stack
- Microcontroller Basics
- Radio protocols
- How to conduct reverse engineering firmware or compiled programs
- Web vulnerability search
- Exploiting binary vulnerabilities

ZERO NIGHTS 2018

2³ EDITION

IoT hacker Toolbox

ZERO NIGHTS 2018

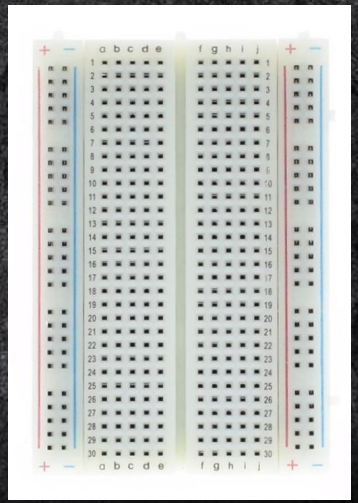
2³ EDITION

IoT hacker Toolbox

ZERO NIGHTS 2018

2^3 EDITION

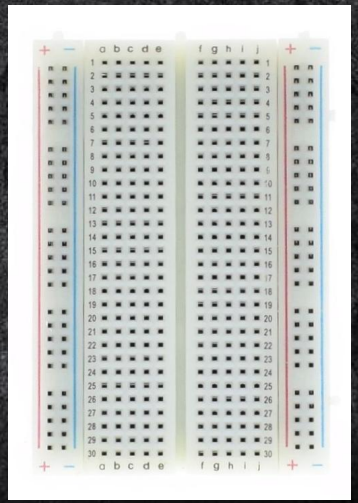
IoT hacker Toolbox



ZERO NIGHTS 2018

2³ EDITION

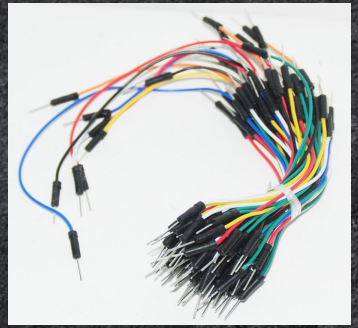
IoT hacker Toolbox

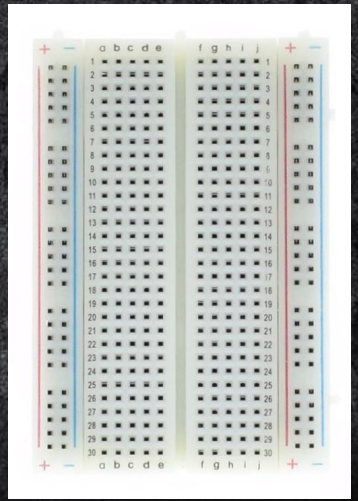


ZERO NIGHTS 2018

2^3 EDITION

IoT hacker Toolbox

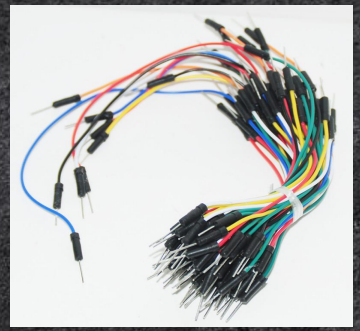


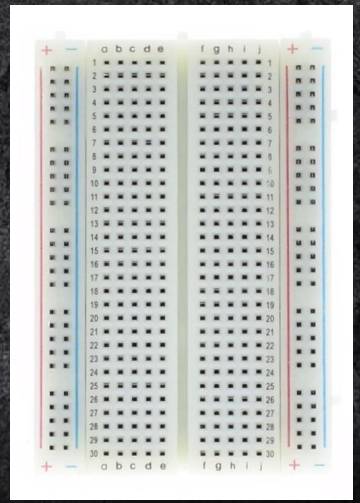


ZERO NIGHTS 2018

2^3 EDITION

IoT hacker Toolbox

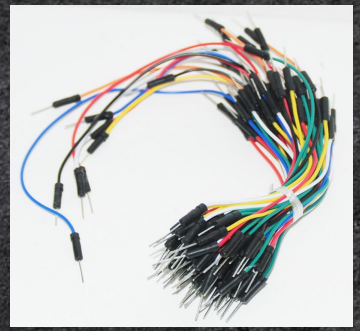


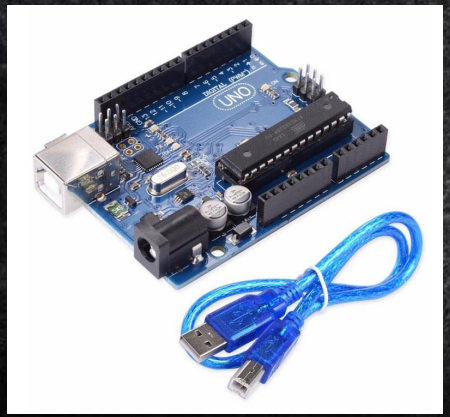


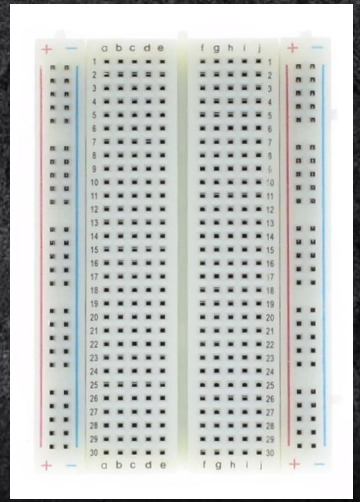
ZERO NIGHTS 2018

2^3 EDITION

IoT hacker Toolbox



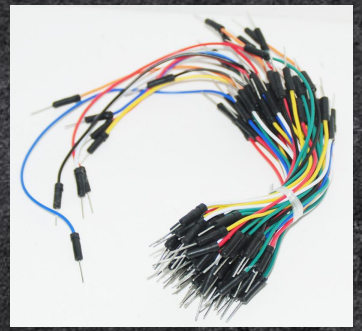


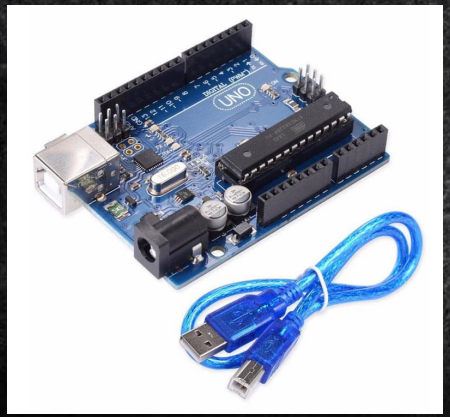


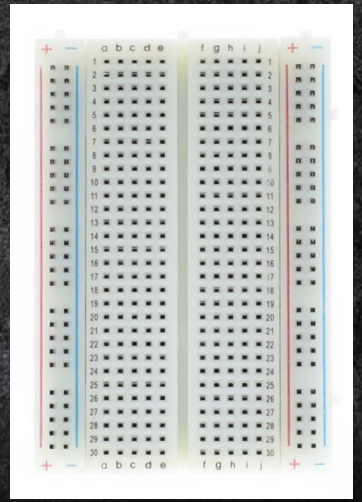
ZERO NIGHTS 2018

2³ EDITION

IoT hacker Toolbox







2³ EDITION

How can I hack my fridge?

1. Search for information about the device without interacting with it.
   (Federal Communications Commission ID)

2³ EDITION

How can I hack my fridge?

1. Search for information about the device without interacting with it. (Federal Communications Commission ID)
2. Create a general scheme that can interact with this device and through which communication channels.

2³ EDITION

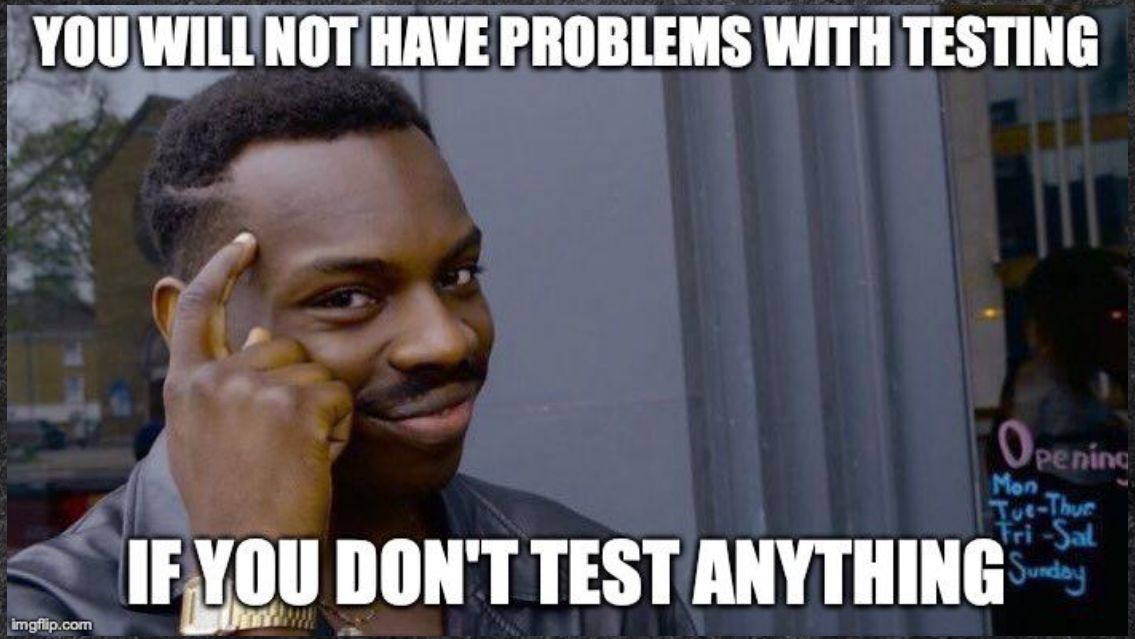
How can I hack my fridge?

1. Search for information about the device without interacting with it. (Federal Communications Commission ID)
2. Create a general scheme that can interact with this device and through which communication channels.
3. For each of the devices and communication channels make a list of threats for which they need to be checked.

ZERO NIGHTS 2018

2³ EDITION

Testing problems



2^3 EDITION

Testing problems

There are two chairs…

- Write small programs each time for specific tasks and delete / forget them later
- Write large programs, refine and store them somehow

2^3 EDITION

Testing problems

A little more about the first chair:

- No structure
- Suitable for specialized tasks
- You will never use it again

Testing problems

Second cherie:

- Complicated structure
- Hard to modify
- Hard to support

ZERO NIGHTS 2018

2^3 EDITION

Drakylar

2018.ZERONIGHTS.ORG

About myself

- BMSTU student ~~1~~ ~~2~~ 3 course
- SFT0 & Invuls CTF teams player
- Hacked not so smart fridge for free pizza
- Rostelecom RedTeam Analyst
- Tested over 30 smart devices (at last year)
- Not funny kidding(but i don’t care - suffer:)

ZERO NIGHTS 2018

2^3 EDITION

What do you know about other IoT frameworks?

ZERO NIGHTS 2018

2^3 EDITION

What do you know about other IoT frameworks?

ZERO NIGHTS 2018

2³ EDITION

IoTSecFuzz - IoT security testing framework

ZERO NIGHTS 2018

2³ EDITION

IoTSecFuzz - IoT security testing framework

- Python 3.7

ZERO NIGHTS 2018

2³ EDITION

IoTSecFuzz - IoT security testing framework

- Python 3.7
- > 30 submodules

ZERO NIGHTS 2018

2³ EDITION

IoTSecFuzz - IoT security testing framework

- Python 3.7
- > 30 submodules
- Console interface

ZERO NIGHTS 2018

2³ EDITION

IoTSecFuzz - IoT security testing framework

- Python 3.7
- > 30 submodules
- Console interface
- GUI interface

ZERO NIGHTS 2018

2³ EDITION

IoTSecFuzz - IoT security testing framework

- Python 3.7
- > 30 submodules
- Console interface
- GUI interface
- Profile system

ZERO NIGHTS 2018

2³ EDITION

IoTSecFuzz - IoT security testing framework

- Python 3.7
- > 30 submodules
- Console interface
- GUI interface
- Profile system
- Import as library


ZERO
NIGHTS
2018
2^3
EDITION
Hardware modules
KERS IN THE AREA
HACKERS IN THE
HACKERS IN THE AR
2018.ZERONIGHTS.ORG

2^3 EDITION

UART

Baudrate bruteforce

```
Trying 9600:
. . . . . a . < . * . . ' . . ; . b
ASCII: 30%
Trying 115200:
L o a d i n g _ f i r m w a r e _ _
ASCII: 100%
. . .
Device baudrate: 115200
```

2^3 EDITION

UART

Baudrate bruteforce

```
Trying 9600:
. . . . . a . < . * . . ' . . ; . b
ASCII: 30%
Trying 115200:
L o a d i n g _ f i r m w a r e _ _
ASCII: 100%
. . .
Device baudrate: 115200
```

2^3 EDITION

UART -> Uboot loader

Creating shell root terminal

```
U-Boot> setenv extra_boot_args init=/bin/sh

U-Boot> setenv optargs init=/bin/sh

U-Boot> setenv bootargs ${bootargs} single
init=/bin/sh

U-Boot> boot
```

2^3 EDITION

UART -> Uboot loader

Memory dumping

```
U-Boot> md.l 0x40000

00040000: 33323130 37363534 62613938 66656463 0123456789abcdef
00040010:
. . .
U-Boot>
```

UART -> Uboot loader

Filesystem dumping

```
U-Boot> cramfsls /etc/
. . .
-rw-r--r-- 1 root root 2659 Sep 17 01:46 /etc/passwd
. . .
U-Boot> cramfsload /etc/passwd
```

UART -> Uboot loader

Filesystem dumping

```
U-Boot> cramfsls /etc/
. . .
-rw-r--r-- 1 root root 2659 Sep 17 01:46 /etc/passwd
. . .
U-Boot> cramfsload /etc/passwd
```

ZERO NIGHTS 2018

2^3 EDITION

TTY Shell

Password bruteforce

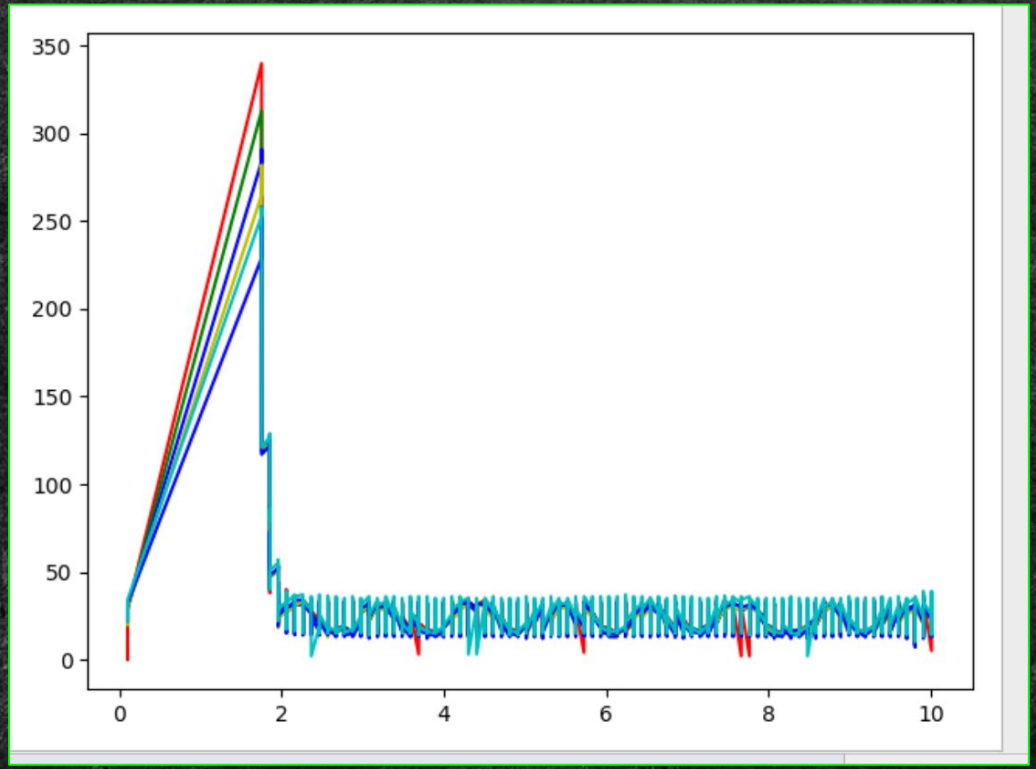
```
. . .
> Enter login: root

> Enter password: toor

> Wrong password!
. . .
```

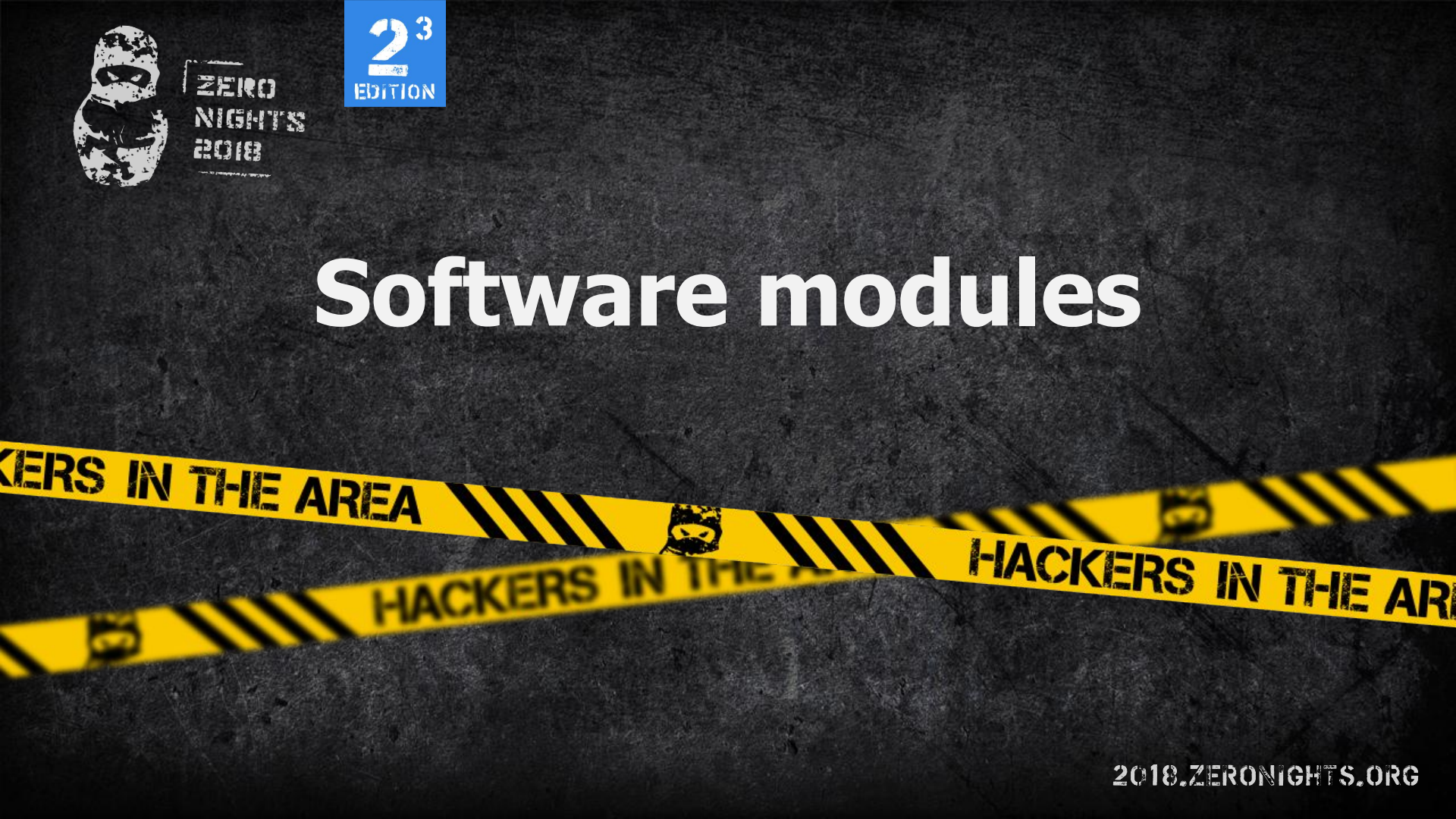
ZERO NIGHTS 2018

2^3 EDITION

Cheap logic analyzer

Arduino UNO




ZERO
NIGHTS
2018
2^3
EDITION
Software modules
KERS IN THE AREA
HACKERS IN THE
HACKERS IN THE AR
2018.ZERONIGHTS.ORG

2^3 EDITION

Binwalk API module

```
> binwalk firmware.bin
< [
    [6264, 'LZO compressed data'],
    . . .
  ]
```

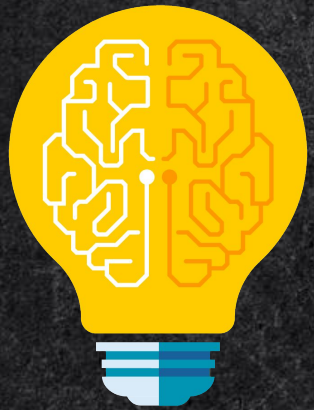
ZERO
NIGHTS
2018

2^3
EDITION

2018.ZERONIGHTS.ORG

ZERO
NIGHTS
2018

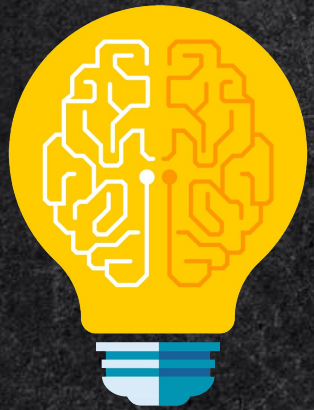
2^3
EDITION



2018.ZERONIGHTS.ORG

ZERO NIGHTS 2018

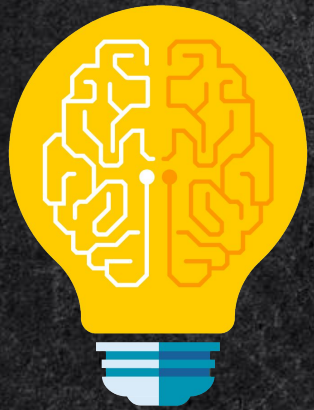
2^3 EDITION

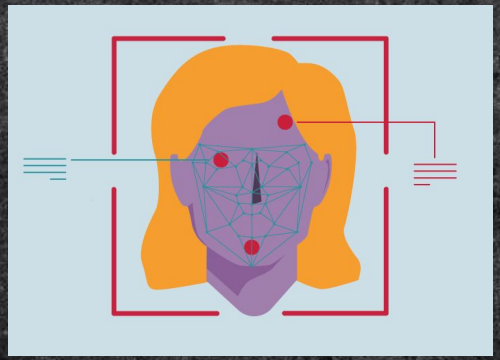


2018.ZERONIGHTS.ORG

ZERO NIGHTS 2018

2^3 EDITION



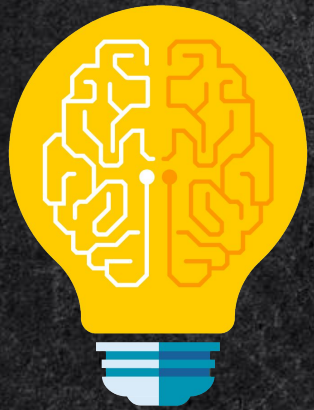


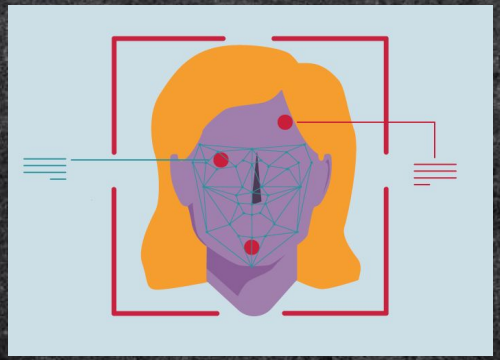
2018.ZERONIGHTS.ORG

ZERO NIGHTS 2018

2^3 EDITION

Firmware hardcoded credentials





2^3 EDITION

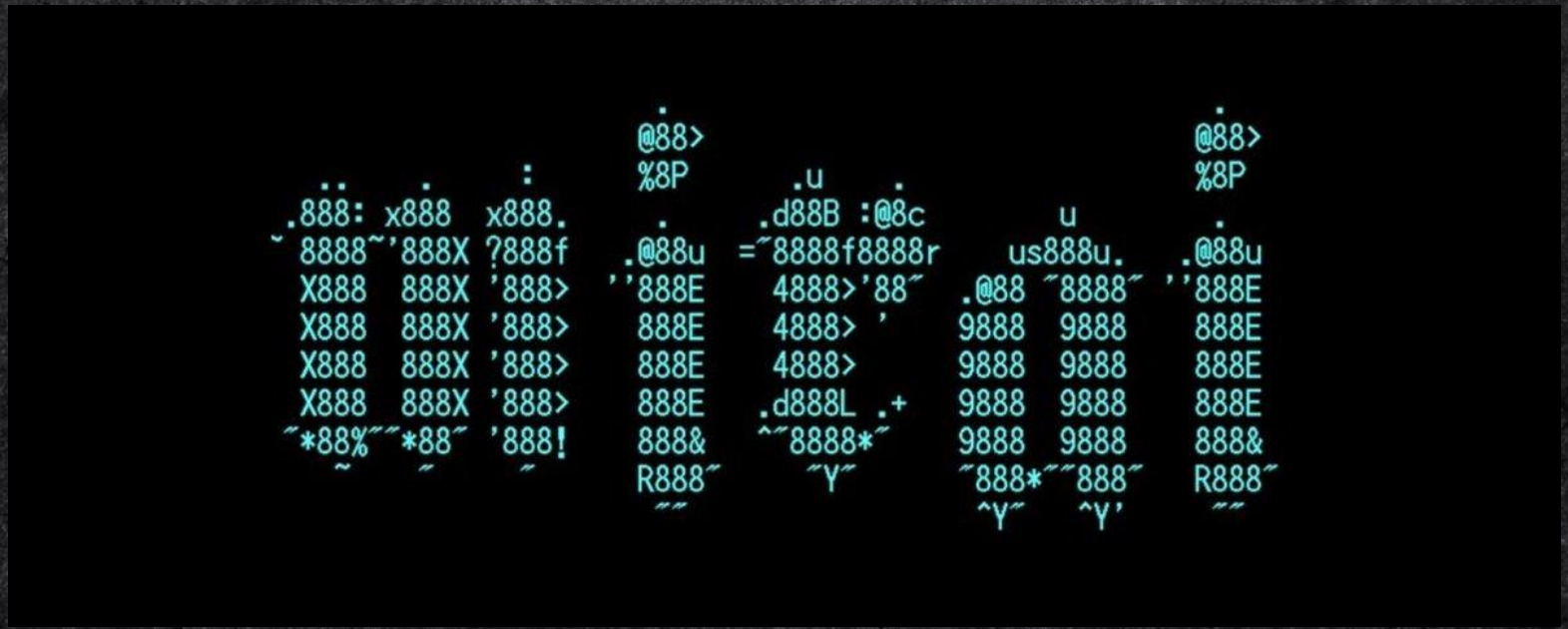
Firmware hardcoded credentials

```
Found file: /etc/passwd
Found hardcoded password:
$1$V.dD1/c1$VHercFBRjVLXgKQTQqzGG.

Found file: /etc/proftpd/ftpd.passwd
Found hardcoded password:
$1$nbFjxIr4$5xMoTwMGJ9.ja5ovXgRFN1
. . .
```

ZERO NIGHTS 2018

2³ EDITION

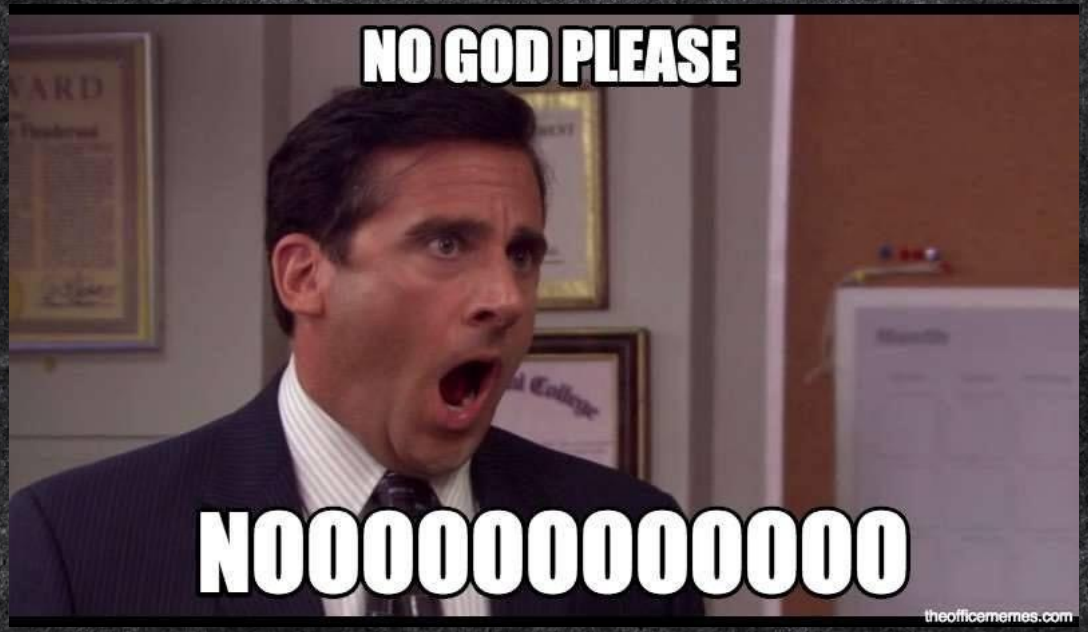
Password fast dictionary bruteforce module



ZERO NIGHTS 2018

2^3 EDITION

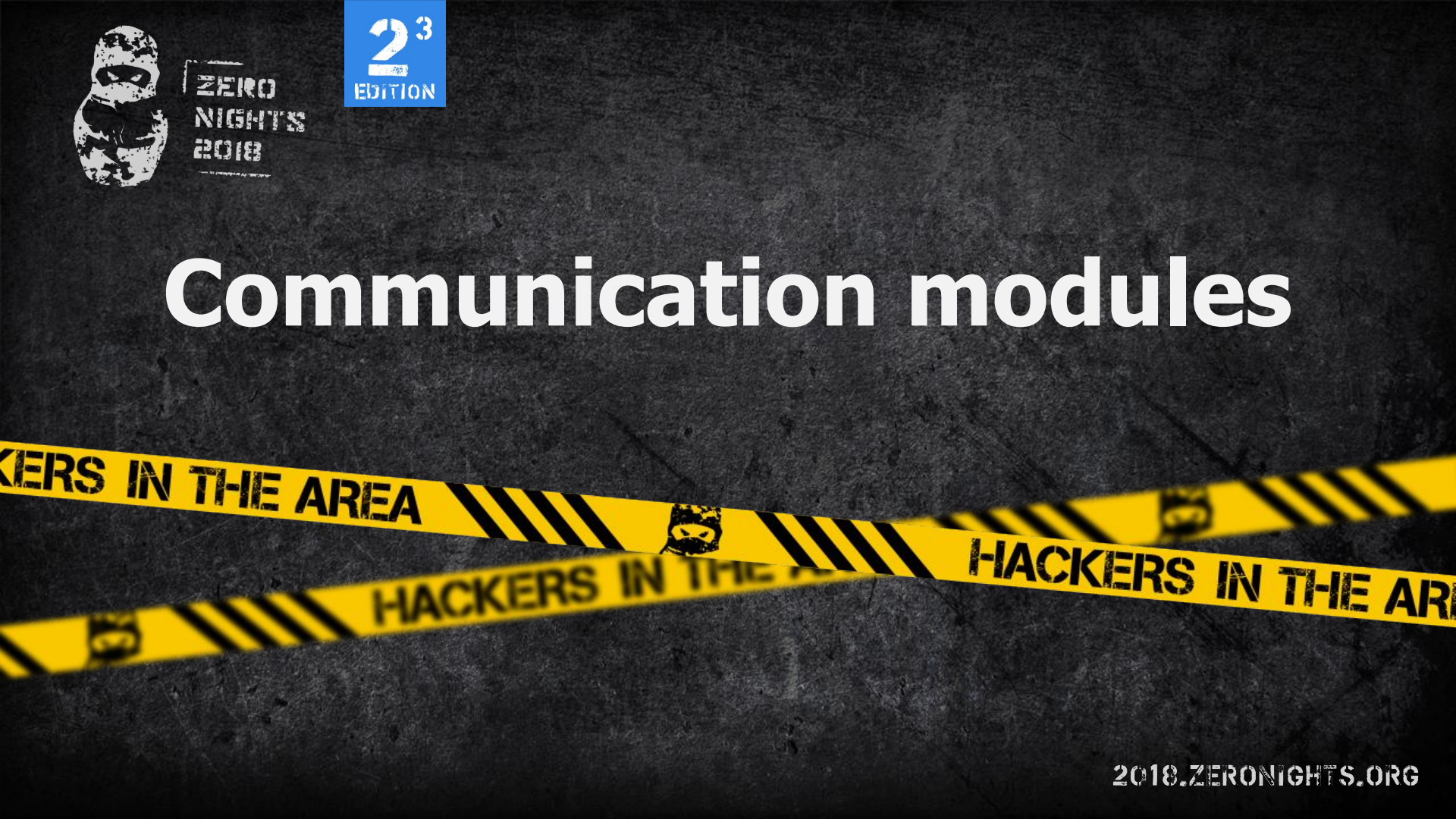
Password fast dictionary bruteforce module



2^3 EDITION

Password fast dictionary bruteforce module

```
Import password: $1$V.dD1/c1$VHercFBRjVLXgKQTQqzGG.

Trying admin..
Trying root..
. . .
Trying toor..
Success!
```


ZERO
NIGHTS
2018
2^3
EDITION
Communication modules
KERS IN THE AREA
HACKERS IN THE
HACKERS IN THE ARE
2018.ZERONIGHTS.ORG

ZERO NIGHTS 2018

2^3 EDITION

WIFI -> Deauth

```
SSID MAC of WIFI network: 00:11:22:33:44:55

Client MAC: AA:BB:CC:DD:EE:FF

Deauthing client...

OK! Create your fake WIFI access point!
```

ZERO NIGHTS 2018

2^3 EDITION

WIFI -> WPA2 Handshake Cracker

```
SSID MAC of WIFI network: 00:11:22:33:44:55

AP Name: kill-me-please

Dictionary: /tmp/10k.txt

Capturing handshake... Captured!

Trying 10000 passwords
. . .
Found: P73wefim458fjwe30234jfma
```

2^3 EDITION

WIFI -> Probe sniffing

```
Starting probe sniffing!
Device AA:BB:CC:DD:EE:FF tried to connect to "HP"
Device 00:11:33:33:77:00 tried to connect to "Samsang"
. . .
Return:
        • "AA:BB:CC:DD:EE:FF": "HP"
        • "00:11:33:33:77:00": "Samsang"
```

2^3 EDITION

ARP -> Finding

```
Starting ARP-Finding!

Found: AA:BB:CC:DD:EE:FF
Found: 00:11:22:33:44:55
. . .
List of found addresses:
    ["AA:BB:CC:DD:EE:FF", "00:11:22:33:44:55", . . . ]
```

ZERO NIGHTS 2018

2^3 EDITION

ARP -> Spoofing

```
Gateway ip: 192.168.1.1
Target: 192.168.1.123
Packets amount: 1000

Starting...
Got MAC address of 192.168.1.1
Got MAC address of 192.168.1.123
Capturing packets...
Finished!
```

2^3 EDITION

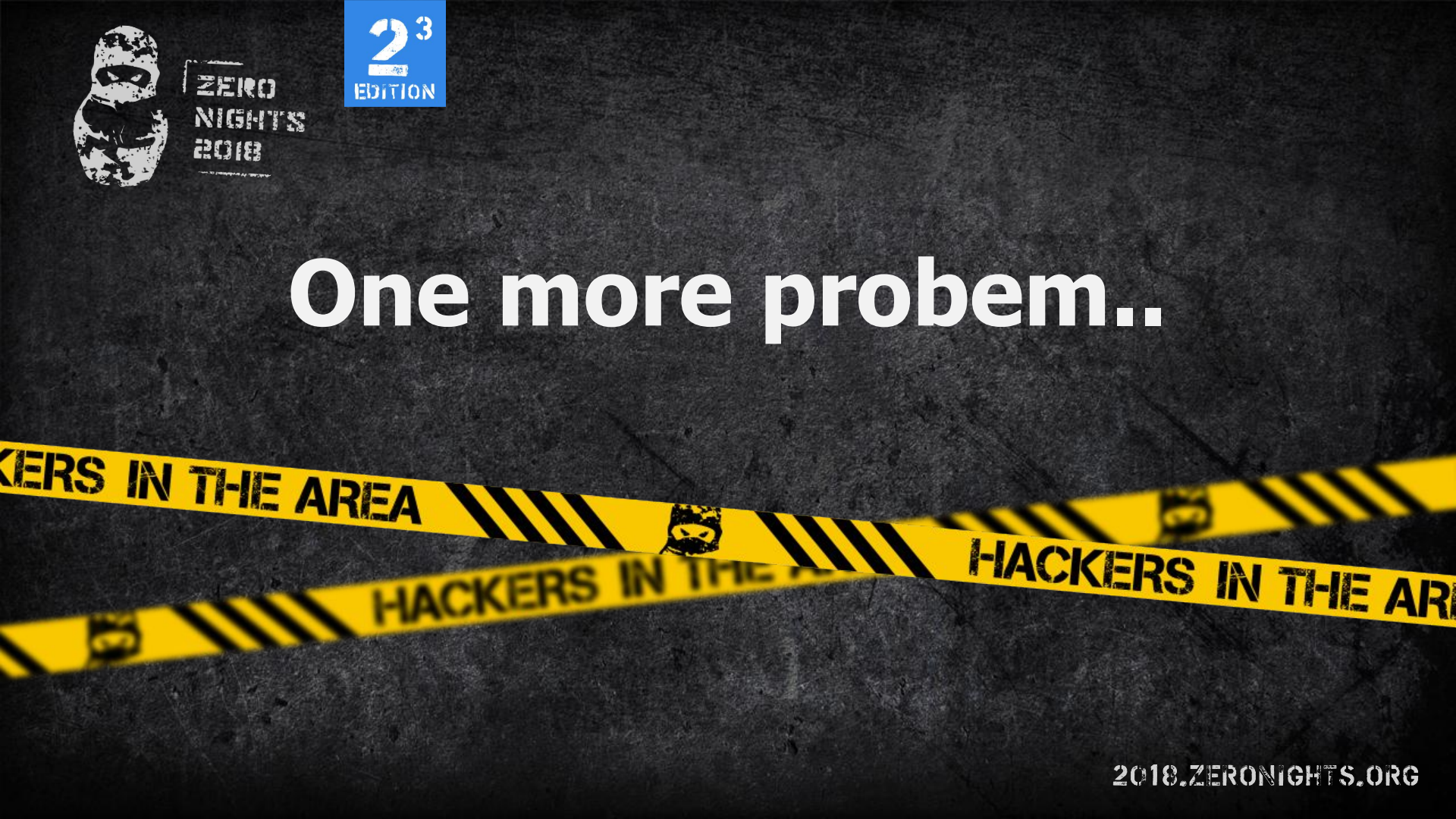
NRF24 -> Find addresses

```
Sniffing for NRF24 Stream.
Changing channel to: 1
Changing channel to: 2
. . .
Most common addresses:

    ["00:00:00:00:00", "AA:BB:CC:DD:EE", . . . ]
```

2^3 EDITION

NRF24 -> Send bytes

```
Device address: AA:BB:CC:DD:EE
Channel: 11
Bytes: 0x1337
Sending bytes... Done!
```


ZERO
NIGHTS
2018
2^3
EDITION
One more probem..
KERS IN THE AREA
HACKERS IN THE
HACKERS IN THE AR
2018.ZERONIGHTS.ORG

2^3 EDITION

- +100500 same options depend on manufacturers/devices
- Options transmission between security specialists

ZERO NIGHTS 2018

2^3 EDITION

- +100500 same options depend on manufacturers/devices
- Options transmission between security specialists

ZERO NIGHTS 2018

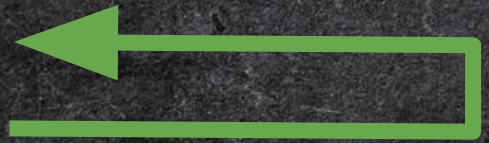
2^3 EDITION

Profile system

ZERO NIGHTS 2018

2³ EDITION

Profile system



- Save & Restore options

ZERO NIGHTS 2018

2³ EDITION

Profile system

- Save & Restore options
- Share options with friends

ZERO NIGHTS 2018

2³ EDITION

Profile system

- Save & Restore options
- Share options with friends
- Share options with everyone

ZERO NIGHTS 2018

2^3 EDITION

BronzeBee

2018.ZERONIGHTS.ORG

ZERO NIGHTS 2018

2³ EDITION

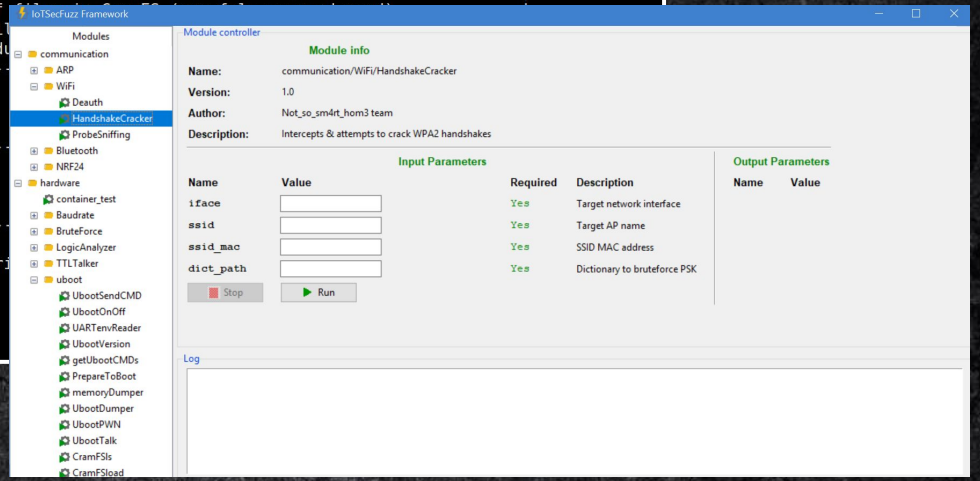
bronzebee@zn2018:~$ whoami

- MIEM HSE student (3rd grade)
- Sberbank cyber security department intern
- Lunary & Invuls CTF team member
- Node.js developer

2³ EDITION

Console & GUI mode

```
ISFFramework > search CramFS
[+] Search results:
+-------------------------+-----------+----------------------+-----------------------------------------------------------------------+
| Name                    |   Version | Author               | Description                                                           |
|-------------------------+-----------+----------------------+-----------------------------------------------------------------------|
| hardware/uboot/CramFSls |         1 | Not_so_sm4rt_hom3 team | Get list of                                                         |
| hardware/uboot/CramFSload |         1 | Not_so_sm4rt_hom3 team | Download fil                                                      |
| hardware/uboot/CramFSdump |         1 | Not_so_sm4rt_hom3 team | Recursive du                                                      |
+-------------------------+-----------+----------------------+-----------------------------------------------------------------------+
ISFFramework > use hardware/uboot/CramFSls
ISFFramework (hardware/uboot/CramFSls) > params
[*] Module input parameters:
+------------------+----------------+--------+-----------+---------------
---------+
| Name             | Current value  | Type   | Required  | Default value
         |
|------------------+----------------+--------+-----------+---------------
---------|
| Device           |                | str    | False     | /dev/tty.usbseri
         |
| Baudrate         |                | int    | False     | 115200
         |
| Timeout          |                | float  | False     | 0.1
```



2^3 EDITION

Console mode

20+ commands

```
ISFFramework > help
+----------------+----------------------+-----------------------------------+--------------------------+
| Name           | Parameters           | Description                       | Aliases                  |
|----------------+----------------------+-----------------------------------+--------------------------|
| back           |                      | Clears the current module         |                          |
| help           |                      | Displays help                     |                          |
| list           |                      | Lists all loaded modules          |                          |
| options        |                      | Prints module in/out parameters   | params                   |
| presets        |                      | Lists presets for current profile |                          |
| presets_import | presets file name    | Imports presets file              |                          |
| profile_import | presets file name    | Imports profile file              | pimport, pr_import       |
| profile_unset  | profile name         | Unsets selected profile           | punset, pr_unset         |
| profile_use    | profile name         | Selects profile to use            | puse, pr_use             |
| profile_list   |                      | Lists all loaded profiles         | profiles, plist, pr_list |
| run            |                      | Executes selected module          | start                    |
| search         |                      | Finds module by name substring    | find                     |
| set            | parameter name, value | Executes selected module          |                          |
| use            | module name          | Selects module to use             | select, sel              |
+----------------+----------------------+-----------------------------------+--------------------------+
ISFFramework >
```

2³ EDITION

Console mode

module usage

```
[+] Search results:
+-----------------------------------+-----------+-----------------------+-----------------------------------+
| Name                              |   Version | Author                | Description                       |
|-----------------------------------+-----------+-----------------------+-----------------------------------|
| hardware/Baudrate/ClassicBruteforce |         1 | Not_so_sm4rt_hom3 team | Bruteforce of device UART baudrate |
+-----------------------------------+-----------+-----------------------+-----------------------------------+
ISFFramework > use hardware/Baudrate/ClassicBruteforce
ISFFramework (hardware/Baudrate/ClassicBruteforce) > params
[*] Module input parameters:
+--------+-----------------+--------+------------+---------------------------+---------------------------------+
| Name   | Current value   | Type   | Required   | Default value             | Description                     |
|--------+-----------------+--------+------------+---------------------------+---------------------------------|
| Device |                 | str    | False      | /dev/tty.usbserial-00000000 | Path to arduino. Example: COM14 |
| Debug  |                 | bool   | False      | False                     | Use verbose output              |
| Time   |                 | int    | False      | 1                         | Time for any baudrate           |
+--------+-----------------+--------+------------+---------------------------+---------------------------------+
[*] Module output parameters:
+----------+-----------------------------------------------------+
| Name     | Description                                         |
|----------+-----------------------------------------------------|
| Baudrate | Baudrate list with the most ascii-readable chars. |
+----------+-----------------------------------------------------+
```

2³ EDITION

Console mode

profiles

```
ISFFramework > profiles
[+] Loaded profiles:
+--------+---------------------+--------------------+
| Name   | Description         |   Number of presets |
|--------+---------------------+--------------------|
| test   | Simple test profile |                  3 |
+--------+---------------------+--------------------+
ISFFramework > profile_use test
[+] Profile set: test
ISFFramework (~@test) > use firmware/test/test1
ISFFramework (firmware/test/test1@test) > params
[*] Module input parameters:
+---------+-----------------+--------+------------+-----------------+--------------------+
| Name    | Current value   | Type   | Required   |   Default value | Description        |
|---------+-----------------+--------+------------+-----------------+--------------------|
| TARGET  |                 | str    | True       |                 | The target path    |
| VERBOSE |                 | bool   | False      |           False | Use verbose output |
| T1      |                 | int    | True       |            1337 | A test param       |
+---------+-----------------+--------+------------+-----------------+--------------------+
[*] Module output parameters:
+--------+--------------+
| Name   | Description  |
|--------+--------------|
| kek    | So kek       |
+--------+--------------+
ISFFramework (firmware/test/test1@test) >
```

2^3 EDITION

Profiles
on the inside

```
{
  "name": "test",
  "description": "Simple test profile",
  "presets_packs": {
    "root": {
      "description": "",
      "origin": "~",
      "presets": {
        "firmware/test/test1:T1": 1337,
        "firmware/test/test2:T2": 8888
      }
    }
  }
}
```

2^3 EDITION

Profiles

presets for existing ones

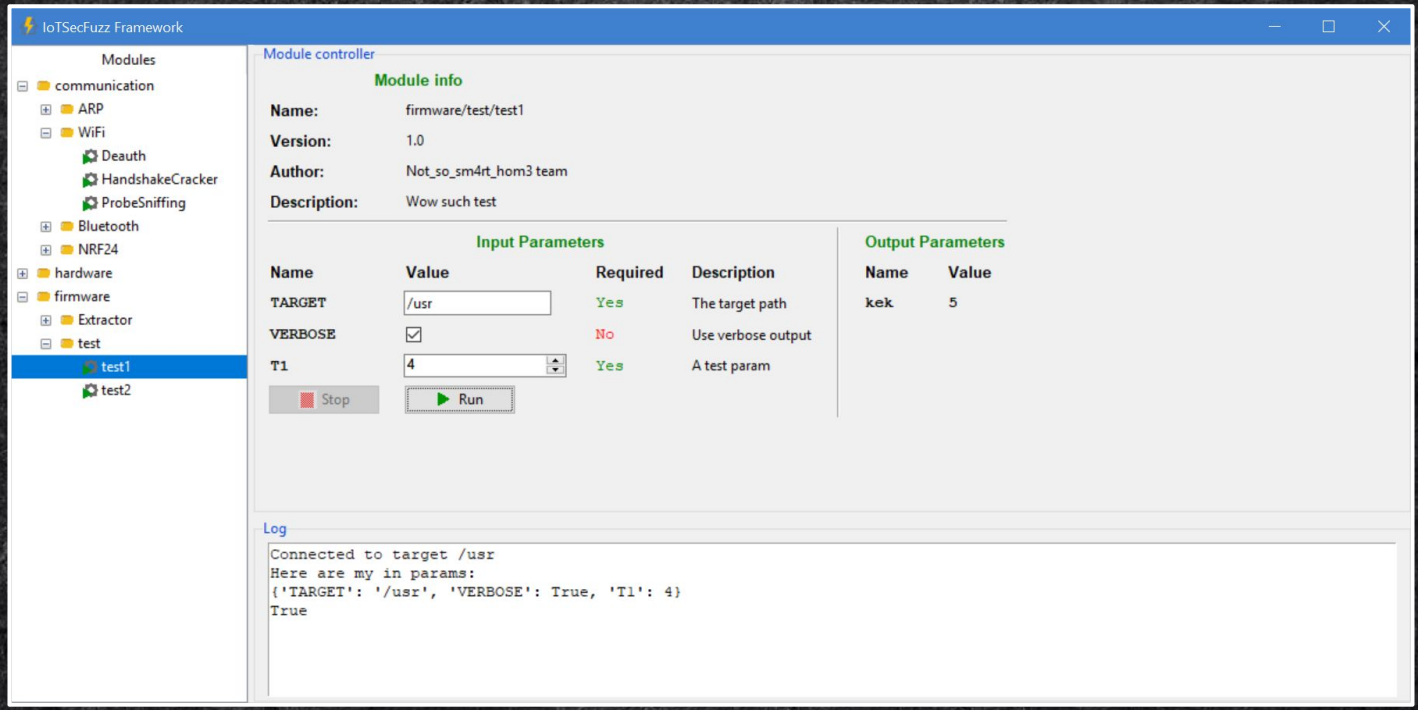
```
{
    "name": "preset_test",
    "description": "Preset pack test",
    "profiles": {
        "test": {
            "hardware/container_test:TARGET": "/"
        }
    }
}
```

imported via *presets_import* command

ZERO NIGHTS 2018

2³ EDITION

GUI mode



ZERO NIGHTS 2018

2³ EDITION

GUI - module tree

IoTSecFuzz Framework

Modules

- communication
  - ARP
  - WiFi
    - Deauth
    - HandshakeCracker
    - ProbeSniffing
  - Bluetooth
  - NRF24
- hardware
- firmware
  - Extractor
  - test
    - test1
    - test2

Module controller

Module info

Name: firmware/test/test1

Version: 1.0

Author: Not_so_sm4rt_hom3 team

Description: Wow such test

Input Parameters

Name	Value	Required	Description
TARGET	/usr	Yes	The target path
VERBOSE	☑	No	Use verbose output
T1	4	Yes	A test param

Stop Run

Output Parameters

Name	Value
kek	5

Log

```
Connected to target /usr
Here are my in params:
{'TARGET': '/usr', 'VERBOSE': True, 'T1': 4}
True
```

ZERO NIGHTS 2018

2³ EDITION

GUI - module controller

IoTSecFuzz Framework

Modules

- communication
  - ARP
  - WiFi
    - Deauth
    - HandshakeCracker
    - ProbeSniffing
  - Bluetooth
  - NRF24
- hardware
- firmware
  - Extractor
  - test
    - test1
    - test2

Module controller

Module info

Name:	firmware/test/test1
Version:	1.0
Author:	Not_so_sm4rt_hom3 team
Description:	Wow such test

Input Parameters

Name	Value	Required	Description
TARGET	/usr	Yes	The target path
VERBOSE	☑	No	Use verbose output
T1	4	Yes	A test param

Stop Run

Output Parameters

Name	Value
kek	5

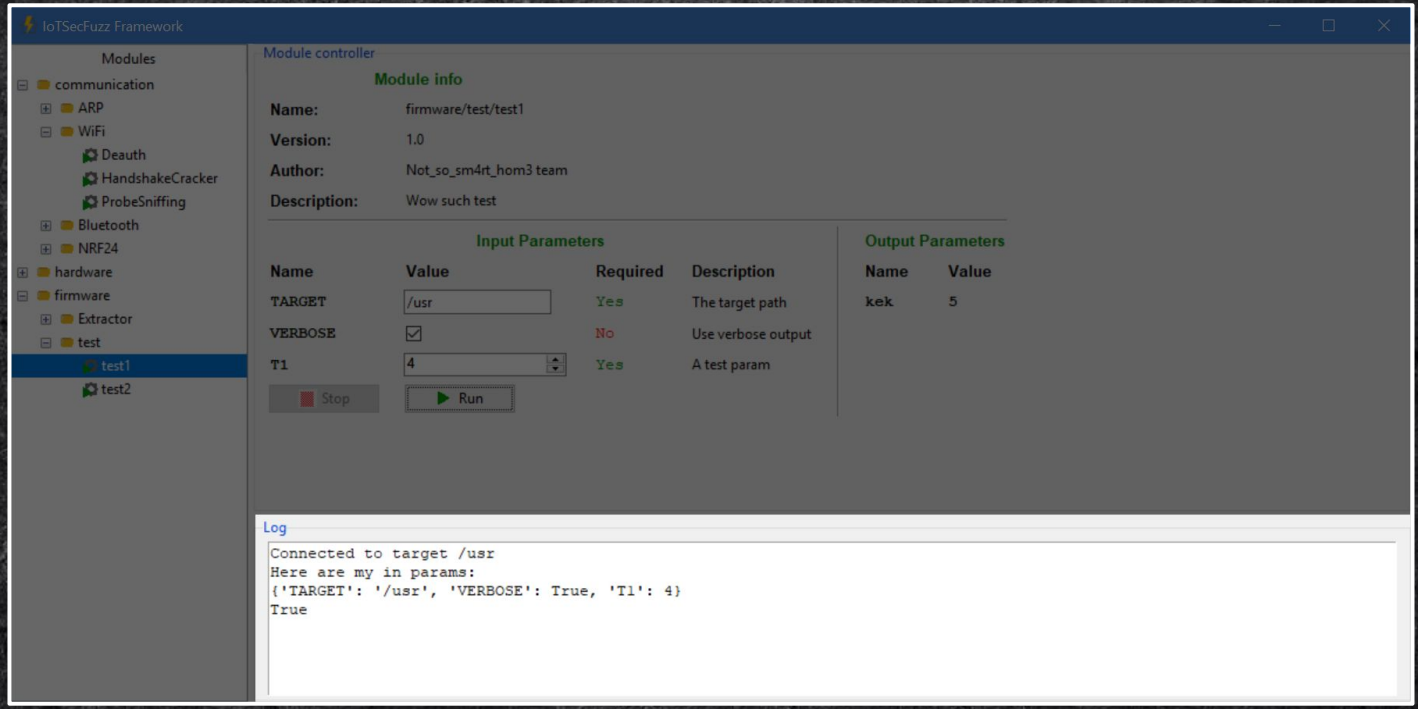
Log

```
Connected to target /usr
Here are my in params:
{'TARGET': '/usr', 'VERBOSE': True, 'T1': 4}
True
```

ZERO NIGHTS 2018

2³ EDITION

GUI - log window



2^3 EDITION

What's under the hood

```
class IOFModule:

    def __init__(self, *, name, version="1.0", description=None,
                 author=None):
        if state != State.LOADING_MODULES:
            raise InvalidStateException("Can only load modules while in %s " +
                                        "state (current state is %s)") % (
                                            State.LOADING_MODULES.name,
                                            state.name))
        self.name = name
        self.version = version
        self.description = description
        self.author = author

    def __call__(self, cls):
        if cls.__module__.startswith("modules"):
            raise ModuleLoadingException(
```

?

2³ EDITION

Module

atomic work unit of framework

Module lifecycle:

1. Initialized
2. Executed with pre-validated input parameters
3. Returns output parameters (if any)
4. Destroyed

2³ EDITION

Software interface of module

- Defined in separate class
- Ability to set parameter description, type, default value, etc.
- name/version/description/author provided in the *@ISFModule* decorator

```
from core.ISFFramework import ISFModule, Param


@ISFModule(name="MyModule",
           version="1.0",
           description="Description",
           author="Author")
class MyModule:
    in_params = {
        "my_in_param": Param("An input parameter", required=False,
                             value_type=int, default_value=10)
    }

    out_params = {
        "my_out_param": Param("An output parameter")
    }

    def run(self, params):
        return {"my_out_param": params["my_in_param"] + 1}
```

2³ EDITION

Software interface of module

- Input parameters description provided through *in_params* field
- Execution with specified parameters through *run* method (also returns output parameters)
- Access other modules using *get_module_class* method

```
from core.ISFFramework import ISFModule, Param


@ISFModule(name="MyModule",
           version="1.0",
           description="Description",
           author="Author")
class MyModule:
    in_params = {
        "my_in_param": Param("An input parameter", required=False,
                             value_type=int, default_value=10)
    }

    out_params = {
        "my_out_param": Param("An output parameter")
    }

    def run(self, params):
        return {"my_out_param": params["my_in_param"] + 1}
```

Atomic module problem

Overview

- Good for targeted exploits, but not for general-purpose tasks
- Terrible for hardware multi-step modules: requires either creating a library or initialize new session each time sub-task is executed
- Still makes you to split each sub-task into different file with module wrapper & same input parameters to connect to device

Atomic module problem
CramFS example

CramFSdump command using this approach:

Initialize CramFSls module (open connection)

Atomic module problem
CramFS example

CramFSdump command using this approach:
Initialize CramFSls module (open connection)
-> run CramFSls

Atomic module problem
CramFS example

CramFSdump command using this approach:
Initialize CramFSls module (open connection)
-> run CramFSls -> destroy CramFSls (close connection)

2^3 EDITION

Atomic module problem
CramFS example

CramFSdump command using this approach:
Initialize CramFSls module (open connection) -> run CramFSls -> destroy CramFSls (close connection) -> Initialize CramFSload (open connection)

Atomic module problem
CramFS example

CramFSdump command using this approach:
Initialize CramFSls module (open connection) -> run CramFSls -> destroy CramFSls (close connection) -> Initialize CramFSload (open connection) -> run CramFSload

Atomic module problem
CramFS example

CramFSdump command using this approach:
Initialize CramFSls module (open connection) -> run CramFSls -> destroy CramFSls (close connection) -> Initialize CramFSload (open connection) -> run CramFSload -> destroy CramFSload (close connection)

Atomic module problem
CramFS example

CramFSdump command using this approach:
Initialize CramFSls module (open connection) -> run CramFSls -> destroy CramFSls (close connection) -> Initialize CramFSload (open connection) -> run CramFSload -> destroy CramFSload (close connection) -> Initialize CramFSdump (open connection)

Atomic module problem
CramFS example

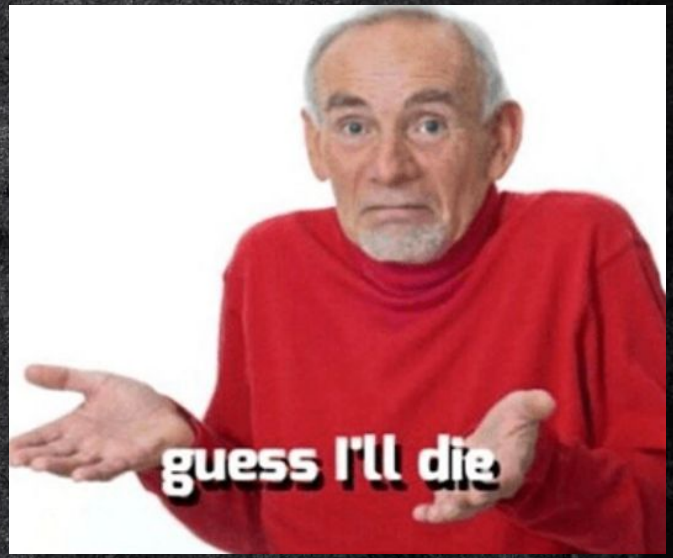
CramFSdump command using this approach:
Initialize CramFSls module (open connection) -> run CramFSls -> destroy CramFSls (close connection) -> Initialize CramFSload (open connection) -> run CramFSload -> destroy CramFSload (close connection) -> Initialize CramFSdump (open connection) -> run CramFSdump

Atomic module problem
CramFS example

CramFSdump command using this approach:
Initialize CramFSls module (open connection) -> run CramFSls -> destroy CramFSls (close connection) -> Initialize CramFSload (open connection) -> run CramFSload -> destroy CramFSload (close connection) -> Initialize CramFSdump (open connection) -> run CramFSdump -> destroy CramFSdump (close connection)

Atomic module problem
CramFS example

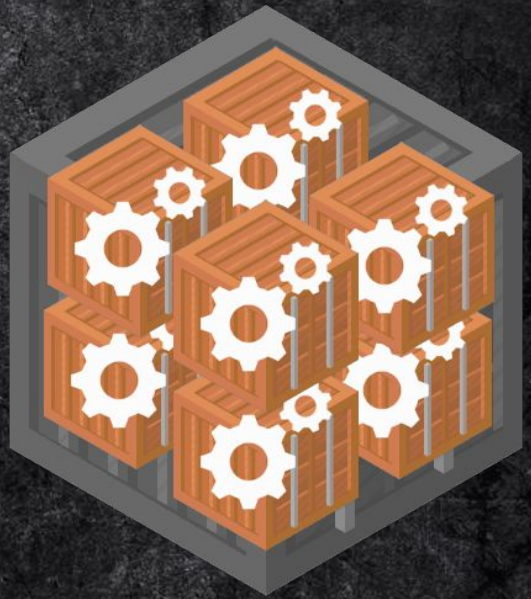
CramFSdump command using this approach:
Initialize CramFSls module (open connection) -> run CramFSls -> destroy CramFSls (close connection) -> Initialize CramFSload (open connection) -> run CramFSload -> destroy CramFSload (close connection) -> Initialize CramFSdump (open connection) -> run CramFSdump -> destroy CramFSdump (close connection)



2^3 EDITION

Atomic module problem
Solution

- Single class to contain multiple modules and common input parameters which is capable of context management (annotated with *@ISFContainer* decorator)
- Run method of each submodule is defined as method in container class and takes additional input parameters from *@submodule* decorator



2^3 EDITION

Atomic module problem
Solution

- Allows the developer to call submodules within the same function as if they were simple functions
- Ability to access container using *get_container_class* method
- From user's perspective, containers/submodules are regular modules (all the work with context is done behind the scene)

2^3 EDITION

Atomic module problem
Solution

```
from core.ISFFramework import ISFContainer, submodule, Param


@ISFContainer(version="1.0",
              author="Not_so_sm4rt_hom3 team")
class TestContainer:

    def __init__(self, in_params):
        self.connected = True

    in_params = {
        "TARGET": Param("The target path", required=True),
        "Baudrate": Param("Device baudrate", value_type=int,
                          required=True)
    }
```

2^3 EDITION

Atomic module problem
Solution

```
@submodule(name="Submodule",
           description="Description",
           in_params={
               "in_param": Param("A test param", value_type=int,
                                 required=True)
           },
           out_params={"out_param": Param("So kek")})
def test_one(self, in_params):
    print("Here are my in params: ")
    print(in_params)
    print(self.connected)
    return {"out_param": in_params["in_param"] * 2}
```

2³ EDITION

Use framework as library

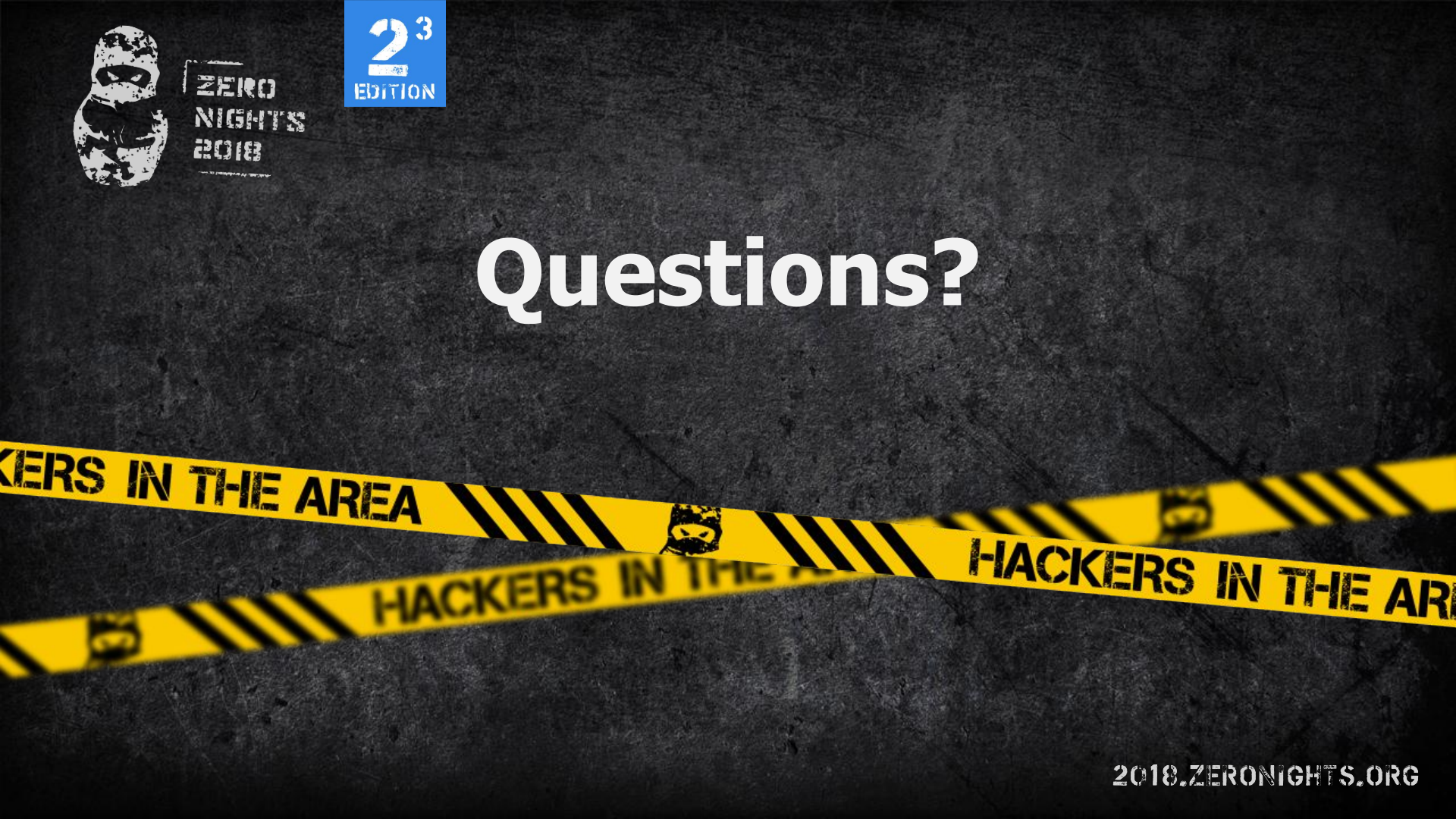
Yep, it's that simple!

```
import core.ISFFramework as ISFFramework

ISFFramework.start()

a = ISFFramework.get_container_class('hardware/Baudrate')

b = a({
    'Device': '/dev/tty.usbserial-00000000',
    'Debug': True
})


baudrate = b.baudrateBruteforce({
    'Time': 1
})['Baudrate'][0]

print(baudrate)
```

2^3 EDITION

What's next?

- More modules, profiles & supported devices
- Community development: website, forum
- Centralized module & profile organization: ability to submit modules for framework repositories & install them directly from application interface


ZERO
NIGHTS
2018
2^3
EDITION
Questions?
KERS IN THE AREA
HACKERS IN THE
HACKERS IN THE AR
2018.ZERONIGHTS.ORG

2^3 EDITION

Thanks for attention!

Gitlab: https://gitlab.com/drakylar1/iotsecfuzz

Marakhovich Sofia: @Soff_M

Shaposhnikov Ilya: @drakylar

Bliznyuk Sergey: @bronzebee