Attacks from a New Front Door in 4G & 5G mobile networks

Dr. Altaf Shaik & Shinjo Park

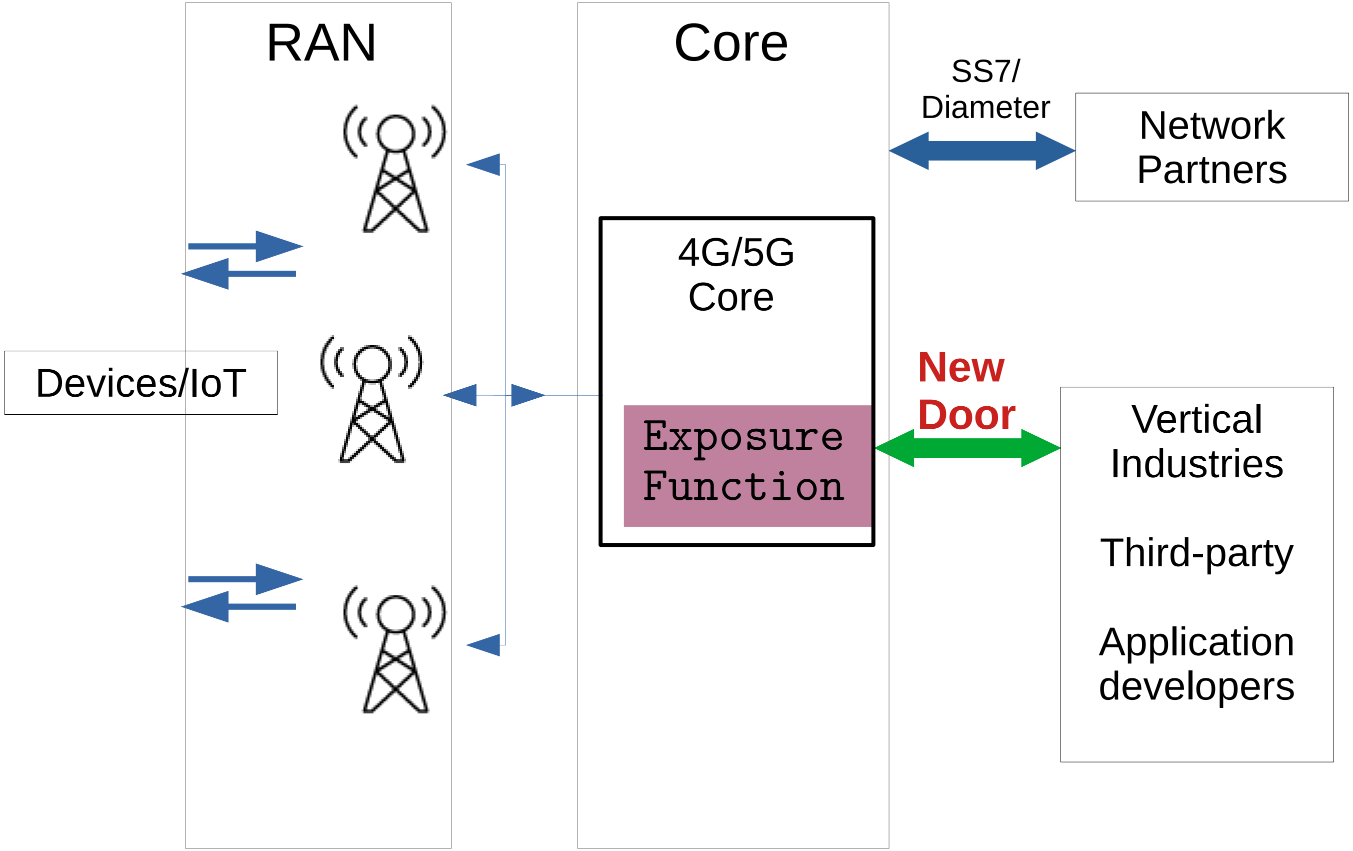
TU Berlin
& FastIoT

Blackhat USA 2022

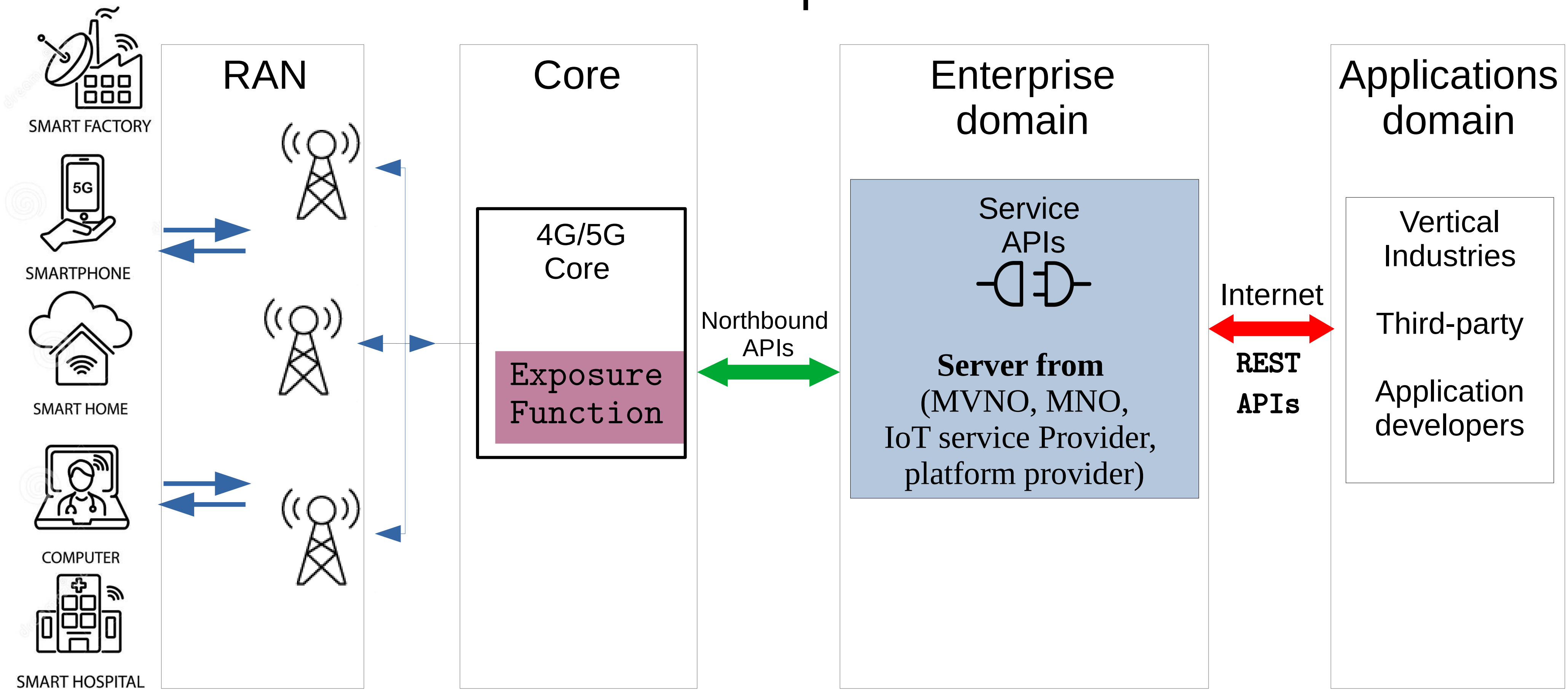
Attacks so far in mobile networks

- Radio access network – IMSI catchers, False base stations
- Signaling interconnect – SS7, Diameter interfaces
- SIM attacks – authentication, sim jacker
- SMS spam, smshing
- Backdoors (Wiretapping)

New front door: exposure function



New front door: exposure function



SMART FACTORY

SMARTPHONE

SMART HOME

COMPUTER

SMART HOSPITAL

RAN

Core

4G/5G Core

Exposure Function

Northbound APIs

Enterprise domain

Service APIs

Server from
(MVNO, MNO,
IoT service Provider,
platform provider)

Internet

REST APIs

Applications domain

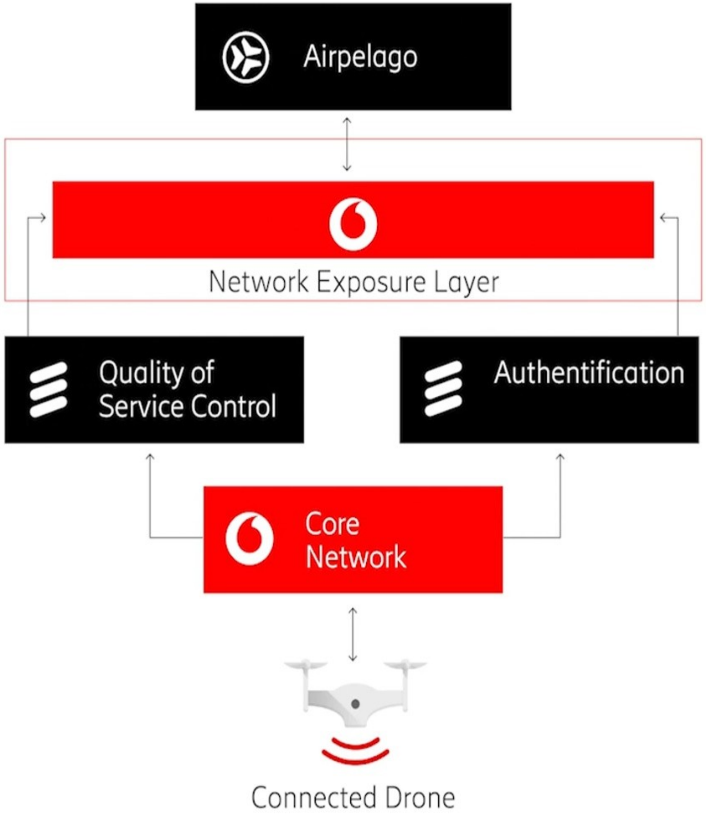
Vertical Industries

Third-party

Application developers

Exposure function: Drone use-case

Cellular-connected Drones to Form Part of Vodafone’s ‘Telco as a Service’ (‘TaaS’) Model



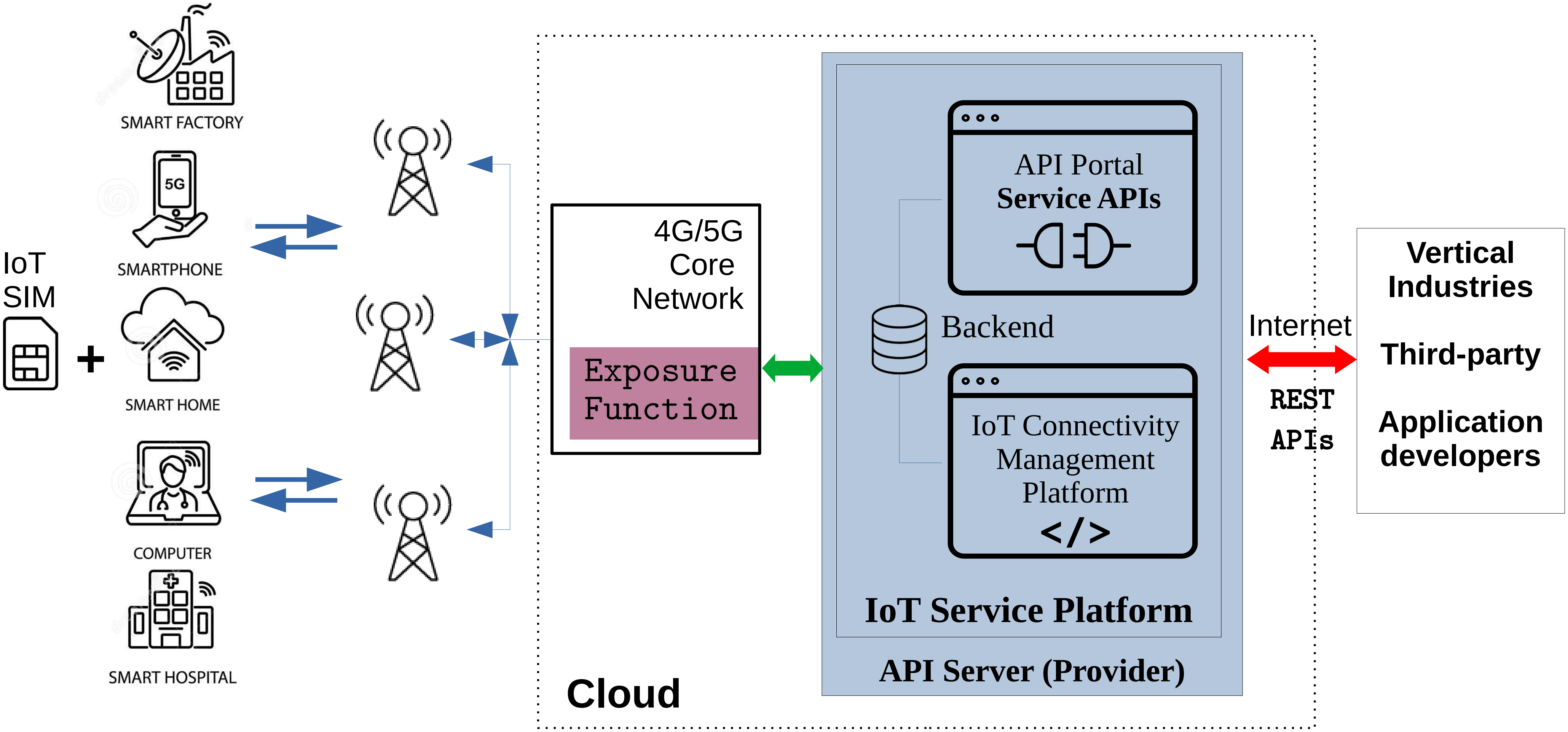
Vodafone’s 5G Mobility Lab in Aldenhoven, Germany

Source: https://www.connectivity.technology/2021/01/cellular-connected-drones-to-form-part.html

Overview

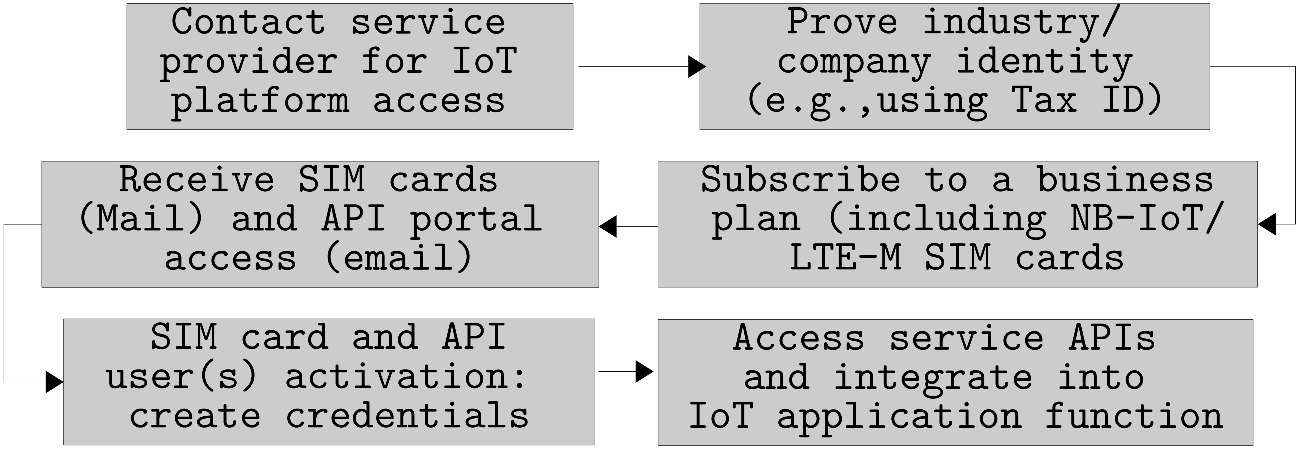
- Access to network exposure
- Features and configurations
- Security investigation
- Design risks
- Findings (vulnerabilities)
- Responsible Disclosure
- Takeaways

Control IoT with 4G and 5G networks



Access to network exposure services via IoT service platforms

- IoT SIM cards (with IP-data and SMS tariff)
  - e.g., 750MB, 250 SMS, 10 year lifetime, roaming free, 10 $$
- Radio connectivity: 4G networks (NB-IoT, LTE-M, 2G)
-



Contact service provider for IoT platform access

Prove industry/ company identity (e.g.,using Tax ID)

Subscribe to a business plan (including NB-IoT/ LTE-M SIM cards

Receive SIM cards (Mail) and API portal access (email)

SIM card and API user(s) activation: create credentials

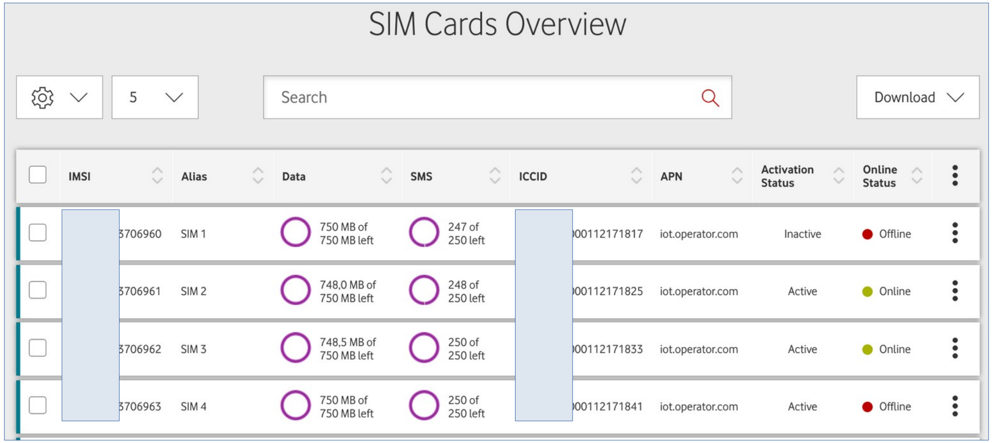
Access service APIs and integrate into IoT application function

Flow diagram: obtaining access to exposure services

Access to network exposure services via IoT service platforms

After business agreement, access is granted to

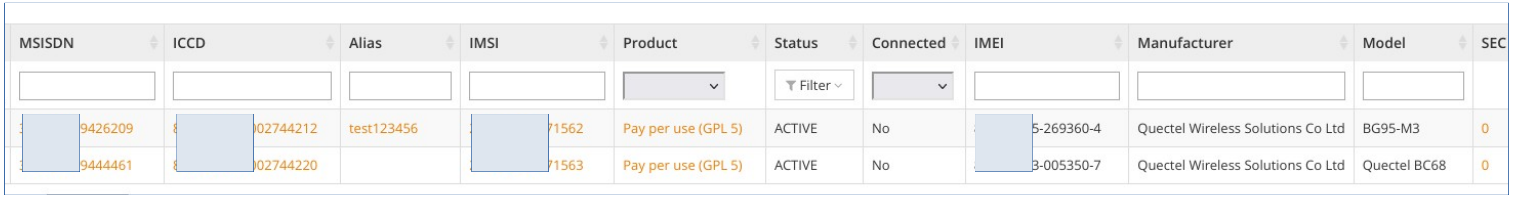
- **IoT connectivity management platform**
  - User/SIM management web application
  - SIM status, activation and deactivation



SIM Cards Overview

□	IMSI	Alias	Data	SMS	ICCID	APN	Activation Status	Online Status
□	[redacted]	SIM 1	750 MB of 750 MB left	247 of 250 left	[redacted]	iot.operator.com	Inactive	● Offline
□	[redacted]	SIM 2	748,0 MB of 750 MB left	248 of 250 left	[redacted]	iot.operator.com	Active	● Online
□	[redacted]	SIM 3	748,5 MB of 750 MB left	250 of 250 left	[redacted]	iot.operator.com	Active	● Online
□	[redacted]	SIM 4	750 MB of 750 MB left	250 of 250 left	[redacted]	iot.operator.com	Active	● Offline

IoT connectivity management platform



MSISDN	ICCD	Alias	IMSI	Product	Status	Connected	IMEI	Manufacturer	Model	SEC
					Filter					
[redacted]	[redacted]	test123456	[redacted]	Pay per use (GPL 5)	ACTIVE	No	[redacted]	Quectel Wireless Solutions Co Ltd	BG95-M3	0
[redacted]	[redacted]		[redacted]	Pay per use (GPL 5)	ACTIVE	No	[redacted]	Quectel Wireless Solutions Co Ltd	Quectel BC68	0

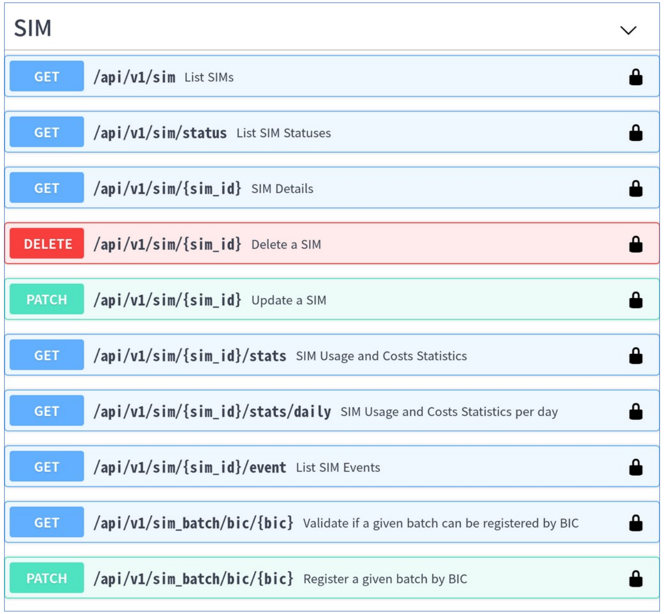
Access to network exposure services via IoT service platforms

IoT service platform

- Service APIs portal (swagger/OpenAPI interface)
- 30 – 100 APIs for IoT device connectivity status, tracking, SMS exchange, IP data exchange (e.g., ping)
- Applications like smart factory, VR, fleet tracking, vehicle telematics
- billing and data plan management, SIM & credential management, device IP address management, roaming policy control, etc.
- API access roles: API administrator, API user, Developer

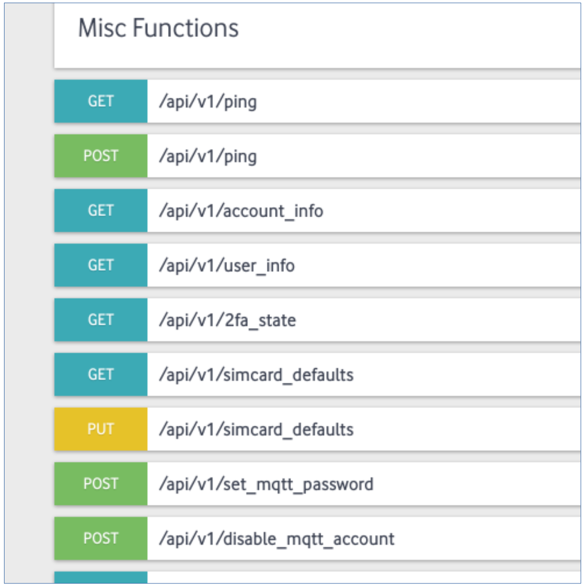
Example platforms and APIs

Service APIs inside IoT Service platform



SIM

GET	/api/v1/sim	List SIMs
GET	/api/v1/sim/status	List SIM Statuses
GET	/api/v1/sim/{sim_id}	SIM Details
DELETE	/api/v1/sim/{sim_id}	Delete a SIM
PATCH	/api/v1/sim/{sim_id}	Update a SIM
GET	/api/v1/sim/{sim_id}/stats	SIM Usage and Costs Statistics
GET	/api/v1/sim/{sim_id}/stats/daily	SIM Usage and Costs Statistics per day
GET	/api/v1/sim/{sim_id}/event	List SIM Events
GET	/api/v1/sim_batch/bic/{bic}	Validate if a given batch can be registered by BIC
PATCH	/api/v1/sim_batch/bic/{bic}	Register a given batch by BIC



Misc Functions

GET	/api/v1/ping
POST	/api/v1/ping
GET	/api/v1/account_info
GET	/api/v1/user_info
GET	/api/v1/2fa_state
GET	/api/v1/simcard_defaults
PUT	/api/v1/simcard_defaults
POST	/api/v1/set_mqtt_password
POST	/api/v1/disable_mqtt_account

API security for Network Exposure

3GPP Standard (recommended) fundamental security mechanisms for exposure services

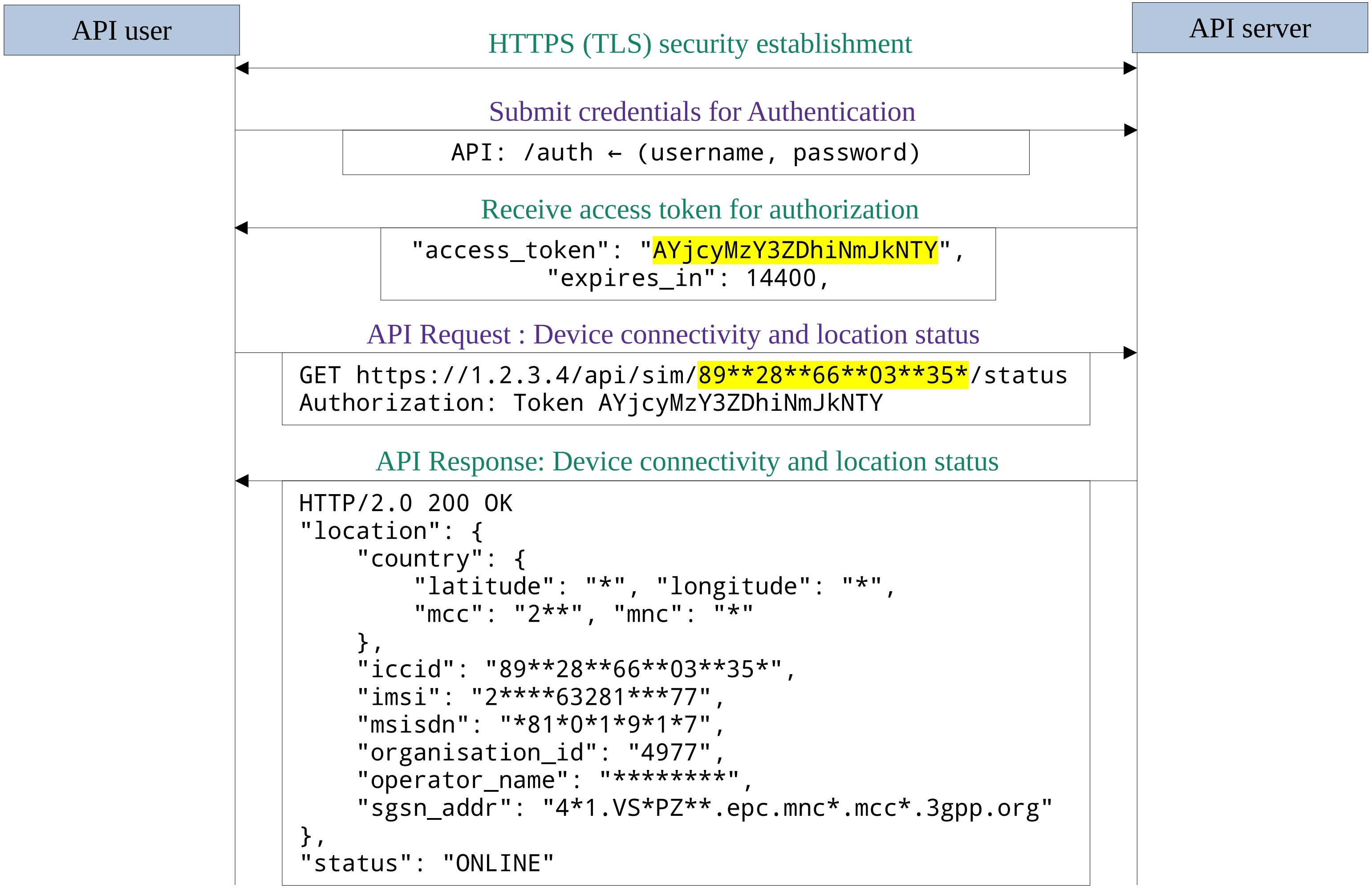
- Authentication & Authorization (OAuth 2.0)
- Confidentiality and integrity protection (TLS)
- Privacy
- Rate limiting*
- Logging and Monitoring*
- Guidelines from GSMA[1,2]

*additional security best-practices

[1] GSM Association. Iot security guidelines for network operators version 2.2 https://www.gsma.com/iot/wp-content/uploads/2020/05/CLP.14-v2.2-GSMA-IoT-Security-Guidelines-for-Network-Operators.pdf

[2] GSM Association. IoT SECURITY GUIDELINES for IoT Service Ecosystems https://www.gsma.com/iot/wp-content/uploads/2016/02/CLP.12-v1.0.pdf

How it works: Get device location



API user

API server

HTTPS (TLS) security establishment

Submit credentials for Authentication

```
API: /auth ← (username, password)
```

Receive access token for authorization

```
"access_token": "AYjcyMzY3ZDhiNmJkNTY",
"expires_in": 14400,
```

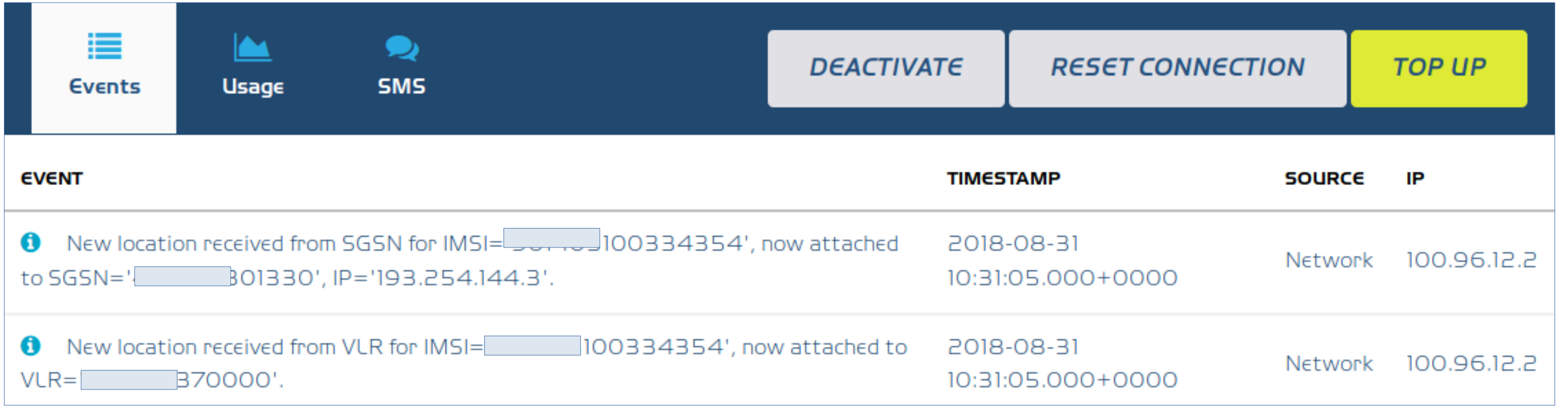
API Request : Device connectivity and location status

```
GET https://1.2.3.4/api/sim/89**28**66**03**35*/status
Authorization: Token AYjcyMzY3ZDhiNmJkNTY
```

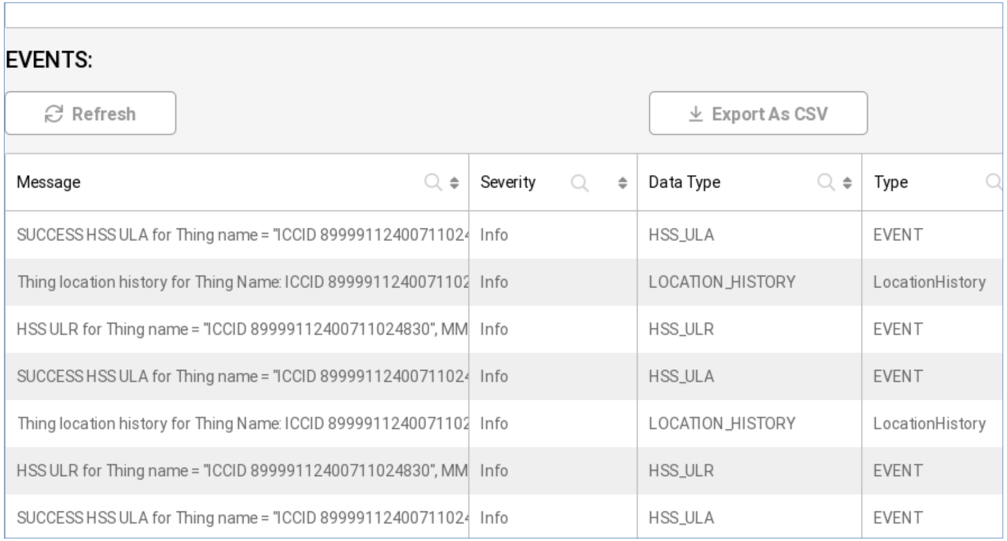
API Response: Device connectivity and location status

```
HTTP/2.0 200 OK
"location": {
    "country": {
        "latitude": "*", "longitude": "*",
        "mcc": "2**", "mnc": "*"
    },
    "iccid": "89**28**66**03**35*",
    "imsi": "2****63281***77",
    "msisdn": "*81*0*1*9*1*7",
    "organisation_id": "4977",
    "operator_name": "********",
    "sgsn_addr": "4*1.VS*PZ**.epc.mnc*.mcc*.3gpp.org"
},
"status": "ONLINE"
```

Device location updates from VLR and HSS



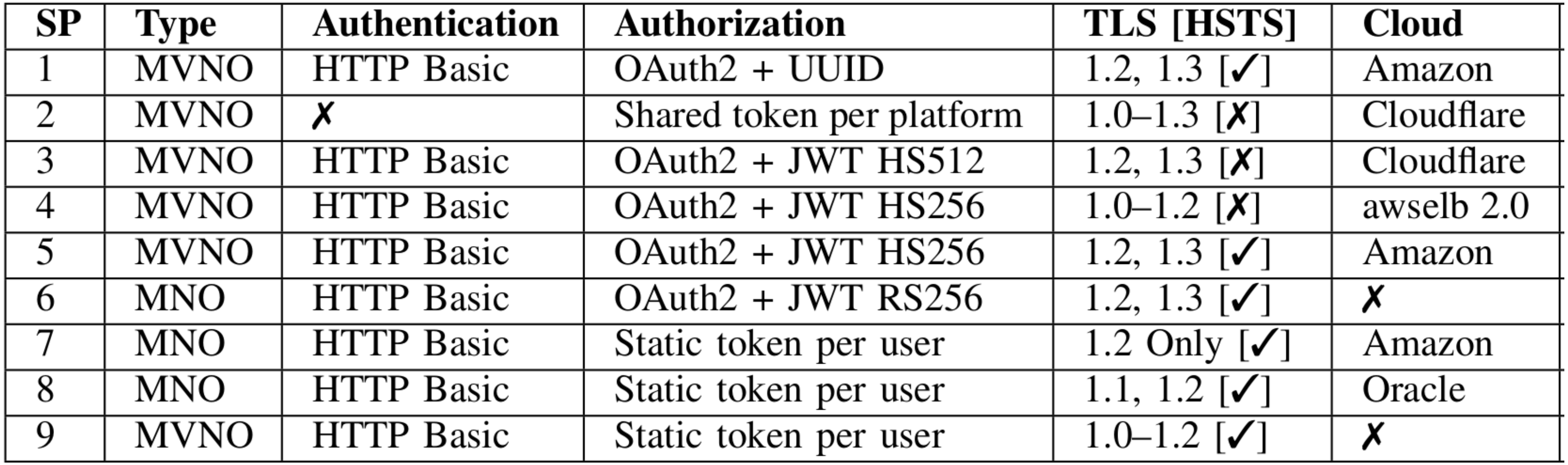
EVENT	TIMESTAMP	SOURCE	IP
New location received from SGSN for IMSI='[redacted]100334354', now attached to SGSN='[redacted]301330', IP='193.254.144.3'.	2018-08-31 10:31:05.000+0000	Network	100.96.12.2
New location received from VLR for IMSI='[redacted]100334354', now attached to VLR='[redacted]370000'.	2018-08-31 10:31:05.000+0000	Network	100.96.12.2



Message	Severity	Data Type	Type
SUCCESS HSS ULA for Thing name = "ICCID 89999112400711024	Info	HSS_ULA	EVENT
Thing location history for Thing Name: ICCID 89999112400711024	Info	LOCATION_HISTORY	LocationHistory
HSS ULR for Thing name = "ICCID 89999112400711024830", MM	Info	HSS_ULR	EVENT
SUCCESS HSS ULA for Thing name = "ICCID 89999112400711024	Info	HSS_ULA	EVENT
Thing location history for Thing Name: ICCID 89999112400711024	Info	LOCATION_HISTORY	LocationHistory
HSS ULR for Thing name = "ICCID 89999112400711024830", MM	Info	HSS_ULR	EVENT
SUCCESS HSS ULA for Thing name = "ICCID 89999112400711024	Info	HSS_ULA	EVENT

```
"pdp_context": {
    "ggsn_ip_addr": "10.70.4.17",
    "rat_type": { "description": "NB-IoT" },
    "sgsn_control_plane_ip_addr": "10.73.4.5",
    "ue_ip_address": "100.96.15.132"
},
```

Commercial IoT service platform features and configuration



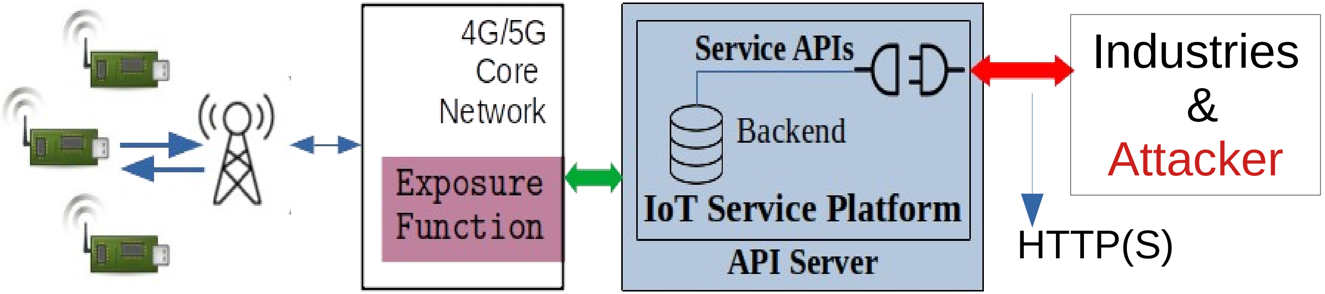
SP	Type	Authentication	Authorization	TLS [HSTS]	Cloud
1	MVNO	HTTP Basic	OAuth2 + UUID	1.2, 1.3 [✓]	Amazon
2	MVNO	✗	Shared token per platform	1.0–1.3 [✗]	Cloudflare
3	MVNO	HTTP Basic	OAuth2 + JWT HS512	1.2, 1.3 [✗]	Cloudflare
4	MVNO	HTTP Basic	OAuth2 + JWT HS256	1.0–1.2 [✗]	awselb 2.0
5	MVNO	HTTP Basic	OAuth2 + JWT HS256	1.2, 1.3 [✓]	Amazon
6	MNO	HTTP Basic	OAuth2 + JWT RS256	1.2, 1.3 [✓]	✗
7	MNO	HTTP Basic	Static token per user	1.2 Only [✓]	Amazon
8	MNO	HTTP Basic	Static token per user	1.1, 1.2 [✓]	Oracle
9	MVNO	HTTP Basic	Static token per user	1.0–1.2 [✓]	✗

HSTS: HTTP Strict-Transport-Security

- SP: Service platform
 Type of exposure: See document by NGMN
- Credentials: Username + Password
- Current network exposure using 4G core (SCEF)

Attack model in service Platforms

- **Requirements**
  - business relationship with the operator or service provider (can forge a tax ID)
    - authentication credentials to get authenticated and authorized
    - access to all service APIs, platform and connectivity management platform
- **Goals**: obtain data of arbitrary IoT service platform users (industries), compromise server and penetrate into mobile core network via the exposure function
- **Privileges**: Web/API knowledge Internet, using HTTP(S), remotely-located, use VPN or tor.



Security problems with IoT platforms?

- Standard security mechanisms. Are they sufficient
- Business logic flaws targeting IoT applications
  - Require manual intensive testing
- Web/API Firewalls or security-by-design
- Security scanners and automated testing
- Limited knowledge on attacks on IoT service platforms

Our interests in the platform

- Dynamic API security analysis on **9 commercial IoT service platforms**
  - To find vulnerabilities in
    - API configuration, input validation, business flow, authentication, access-control, and transport layer security such as encryption.
  - Select APIs that have high impact on business, reputation
    - Billing fraud, DoS, code execution, device hijacking
    - Send SMS or IP messages to arbitrary IoT devices, Reset billing and charging counters, APN manipulation, location tracking, device blacklisting
  - Model a set of Attacks:
    - Inject Malicious payloads, strings, characters, files
    - Guidelines from OWASP web security testing, REST security cheat sheets
    - Tools: Burpsuite, ZAP and other well-known for API testing

Ethical considerations

- Only access or manipulate API data corresponding to our own user/admin accounts.
- Only key API parameters (like IMSI,ICCID, APN, Tariff, topup, MSISDN, SMS) per platform are analyzed for vulnerabilities – to avoid traffic towards API platform
- GET/POST/PUT operations are carried out into our own accounts
- We took measures neither to damage the exposure platform nor interrupt the ongoing API services for other verticals/users.
- Clear guessing strategy is applied rather than a random penetration/function testing
- Noisy attacks such as DoS or bruteforce are ignored

Design risks in IoT service platforms (9)

(access-control, authentication, data exposure)

Forged access?

Procedure to obtain access to IoT service platforms is vulnerable to a social engineering attack

- Attacker registers using a forged company (tax) ID and spoofed email address. Relaxed verification found with many providers
- Receives SIM cards to a private(arbitrary) address and also access to service APIs
- Now attacker has access to IoT platform cloud and data resources hosted on it
- Attacker masquerades as a target company/industry while accessing the platform
- Limitless API operations and probing to find vulnerabilities. No rate-limits in many platforms.
- Lack of (strict) monitoring and logging facilities are added advantage for attacker
- A strict KYC procedure should be implemented by both providers and operators.

Username and password policy for API authentication

Password creation, update, management are not compliant with GSMA guidelines[1,2]:

- Weak passwords are allowed (such a *root, admin, iotadministrator*) for credentials
- Some don’t allow "few dictionary passwords” and have shortcomings"
- Some restrict dictionary passwords during account creation, but allow them during password update
- Fix: comply to best password practices

* asdf1234, qwer1234, qwerty1234 -> weak password, not allowed
* 1qaz2wsx -> top 100 weak password
* iotadmin1 -> Set password error : This is similar to a commonly used password
* iotuser1 -> Set password error : Add another word or two. Uncommon words are better.

*** iotuser10, Password1234, Administrator1 -> allowed**

[1] GSM Association. Iot security guidelines for network operators version 2.2, Section 5.8.4- Secure IoT Connectivity Management Platform
https://www.gsma.com/iot/wp-content/uploads/2020/05/CLP.14-v2.2-GSMA-IoT-Security-Guidelines-for-Network-Operators.pdf

[2] Referring to section 6.11 of GSMA CLP.12 - Never allow a user to utilize a default, weak, or poorly designed password.
https://www.gsma.com/iot/wp-content/uploads/2016/02/CLP.12-v1.0.pdf

Token management

- No OAuth based token generation in several platforms,
- Token expiry
  - Static API token (does not expire), should be revoked for every API user
  - 24 hours to 1 week
- Fix: Use standard approach of Oauth and JSON web tokens for authorization

1. 3GPP. Security aspects of Machine-Type Communications (MTC) and other mobile data applications communications enhancements. Technical Specification (TS) 33.187. Section 4.7 Requirements on T8 reference point
https://www.etsi.org/deliver/etsi_ts/133100_133199/133187/16.00.00_60/ts_133187v160000p.pdf

2. 3GPP. Security aspects of Common API Framework (CAPIF) for 3GPP northbound APIs. Technical Specification (TS) 33.122, 3rd Generation Partnership Project.

Lack of rate limiting for API requests

Only 2 platforms have rate-limits for API requests

- Test: Sending 250/500 valid GET/POST requests in short period
- Using same IP address and user account for all requests
- No backoff period or IP ban was observed from the API gateway
  - Did not receive any HTTP response like : `429 Too Many Requests`
- Some providers specify rate-limits in user manuals, but in practice they are unavailable
- Fix: Rate limiting policies with random/exponential back-off timers

Private identifiers used inside IoT domain

ICCID, IMEI, and IMSI exposed outside of 3GPP domain (can be SUPI in 5G)

- To access/indicate the SIM cards and IoT devices; convenient for developers and API users
- Violates 3GPP privacy requirement [1] for Machine type communications using exposure services
- Enables user/device enumeration
- Fix: an identifier like General Purpose Subscriber Identifier (GPSI[2]) or other custom identifier. Avoid linking to any identifiers used over the radio interface.
  - An alphanumeric proprietary id and its mapping to IMSI is known only to the provider/operator.

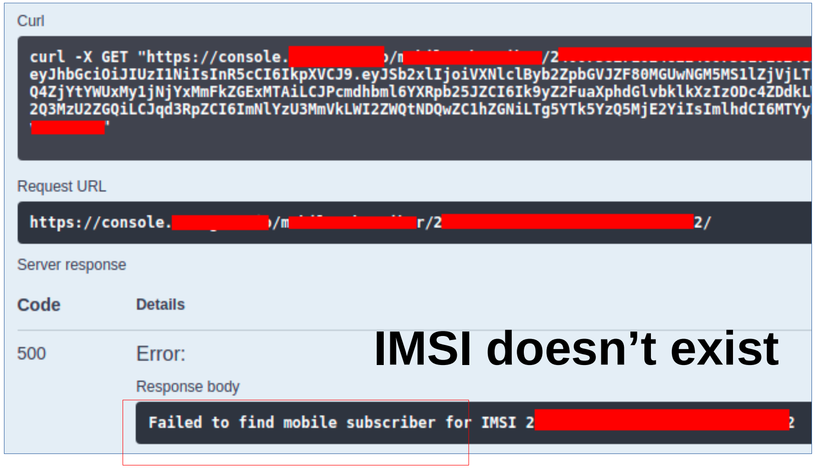
[1] 3GPP. Security aspects of Machine-Type Communications (MTC) and other mobile data applications communications enhancements. Technical Specification (TS) 33.187. Section 4.7 Requirements on T8 reference point https://www.etsi.org/deliver/etsi_ts/133100_133199/133187/16.00.00_60/ts_133187v160000p.pdf

[2] 5G; Procedures for the 5G System (5GS) (3GPP TS 23.502 version 15.4.1 Release 15)

Verbose error messages

Easy user enumeration via probing with IMSI/ICCID/IMEI

- Attacker can find existing and non-existing IMSIs registered on the platform/database from the different API error responses
- Fix: The error can be very generic, such as, *unauthorized*.



Curl

```
curl -X GET "https://console.[redacted]/[redacted]/2[redacted]
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJSb2xlIjoiVXNlclByb2ZpbGVJZF80MGUwNGM5MS1lZjVjLT
Q4ZjYtYWUxMy1jNjYxMmFkZGExMTAiLCJPcmdhbml6YXRpb25JZCI6Ik9yZ2FuaXphdGlvbklkXzIzODc4ZDdkL
2Q3MzU2ZGQiLCJqd3RpZCI6ImNlYzU3MmVkLWI2ZWQtNDQwZC1hZGNiLTg5YTk5YzQ5MjE2YiIsImlhdCI6MTYy
'[redacted]'
```

Request URL

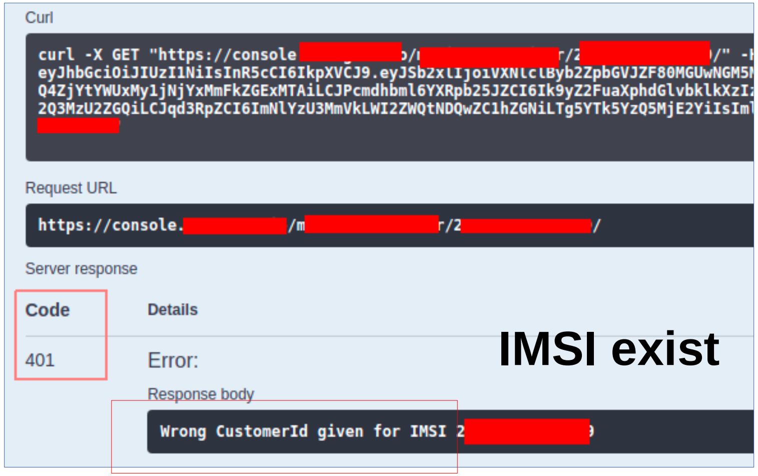
```
https://console.[redacted]/[redacted]r/2[redacted]2/
```

Server response

Code	Details
500	Error: Response body

```
Failed to find mobile subscriber for IMSI 2[redacted]2
```

IMSI doesn't exist



Curl

```
curl -X GET "https://console[redacted]/[redacted]/2[redacted]/" -H
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJSb2xlIjoiVXNlclByb2ZpbGVJZF80MGUwNGM5M
Q4ZjYtYWUxMy1jNjYxMmFkZGExMTAiLCJPcmdhbml6YXRpb25JZCI6Ik9yZ2FuaXphdGlvbklkXzIz
2Q3MzU2ZGQiLCJqd3RpZCI6ImNlYzU3MmVkLWI2ZWQtNDQwZC1hZGNiLTg5YTk5YzQ5MjE2YiIsIml
[redacted]
```

Request URL

```
https://console.[redacted]/[redacted]r/2[redacted]/
```

Server response

Code	Details
401	Error: Response body

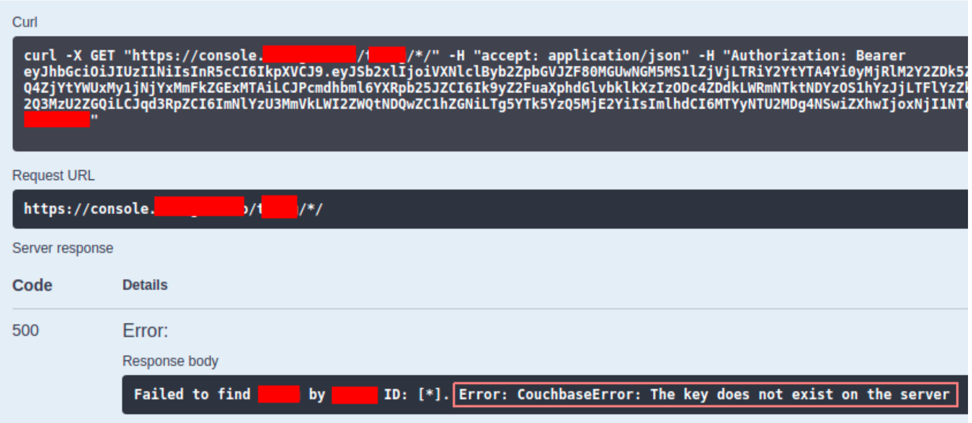
```
Wrong CustomerId given for IMSI 2[redacted]
```

IMSI exist

Internal software information exposed

Database software information exposed via error messages: Couchbase, Jboss

- Platform deployment details are also exposed such cloud provider and etc.
- Deprecated TLS versions are negotiable (TLS v1.2/1.0)



Curl

```
curl -X GET "https://console.[redacted]/[redacted]/*/" -H "accept: application/json" -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJSb2xlIjoiVXNlclByb2ZpbGVJZF80MGUwNGM5MS1lZjVjLTRiY2YtYTA4Yi0yMjRlM2Y2ZDk5
Q4ZjYtYWUxMy1jNjYxMmFkZGExMTAiLCJPcmdhbml6YXRpb25JZCI6Ik9yZ2FuaXphdGlvbklkXzIzODc4ZDdkLWRmNTktNDYzOS1hYzJjLTFlYzZk
2Q3MzU2ZGQiLCJqd3RpZCI6ImNlYzU3MmVkLWI2ZWQtNDQwZC1hZGNiLTg5YTk5YzQ5MjE2YiIsImlhdCI6MTYyNTU2MDg4NSwiZXhwIjoxNjI1NT
[redacted]"
```

Request URL

```
https://console.[redacted]/[redacted]/*/
```

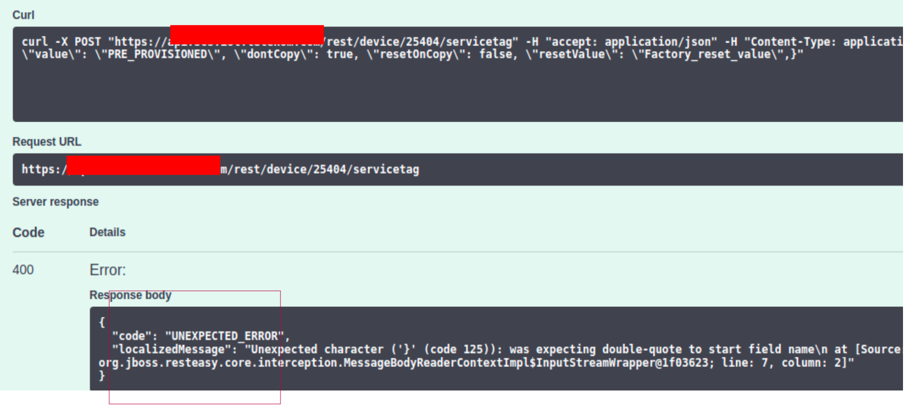
Server response

Code Details

500 Error:

Response body

```
Failed to find [redacted] by [redacted] ID: [*]. Error: CouchbaseError: The key does not exist on the server
```



Curl

```
curl -X POST "https://[redacted]/rest/device/25404/servicetag" -H "accept: application/json" -H "Content-Type: applicati
\"value\": \"PRE_PROVISIONED\", \"dontCopy\": true, \"resetOnCopy\": false, \"resetValue\": \"Factory_reset_value\",}"
```

Request URL

```
https:/[redacted]m/rest/device/25404/servicetag
```

Server response

Code Details

400 Error:

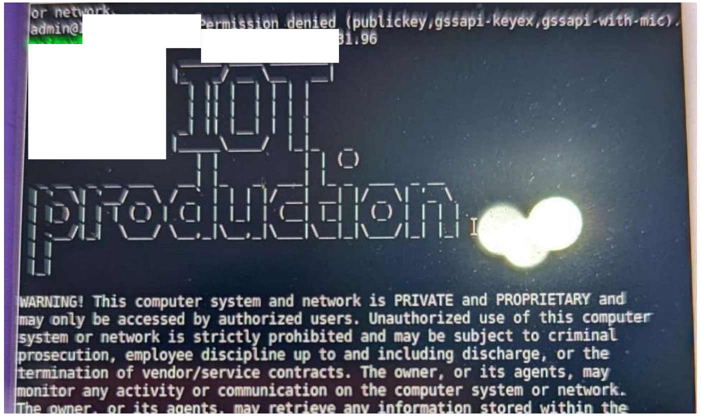
Response body

```
{
  "code": "UNEXPECTED_ERROR",
  "localizedMessage": "Unexpected character ('}' (code 125)): was expecting double-quote to start field name\n at [Source
org.jboss.resteasy.core.interception.MessageBodyReaderContextImpl$InputStreamWrapper@1f03623; line: 7, column: 2]"
}
```

Internal node exposure

APIs leak Core network elements/gateway exposes internal SSH ports/interface

- SSH Login attempt are made to an internal IoT node
- Forged attacker can launch a bruteforce
- Fix: configuration control and reduce exposure

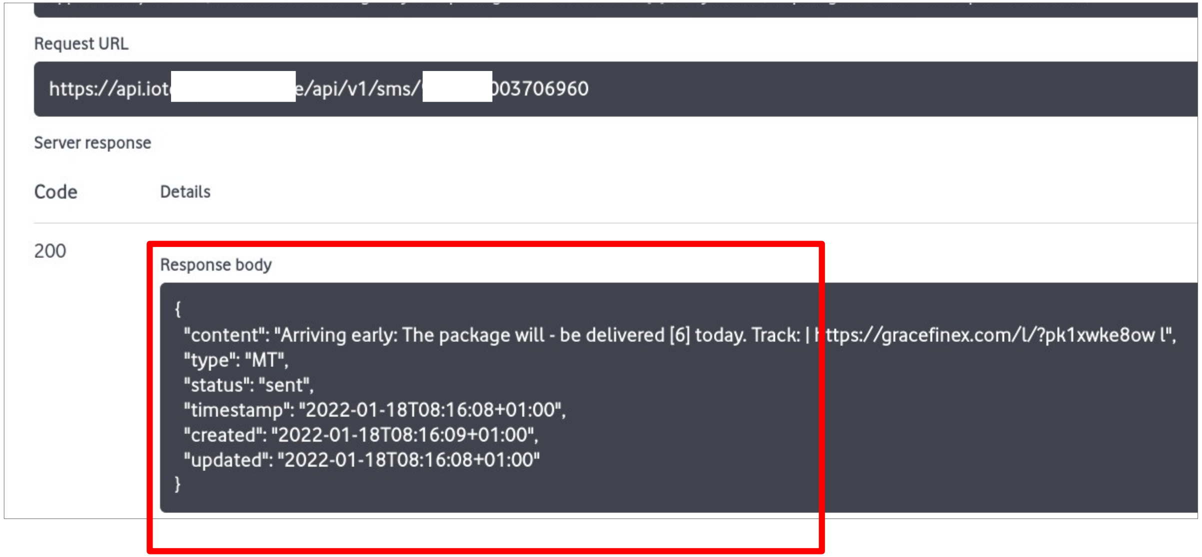


```
or network.
admin@[redacted]Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[redacted]31.96

IoT.
production

WARNING! This computer system and network is PRIVATE and PROPRIETARY and
may only be accessed by authorized users. Unauthorized use of this computer
system or network is strictly prohibited and may be subject to criminal
prosecution, employee discipline up to and including discharge, or the
termination of vendor/service contracts. The owner, or its agents, may
monitor any activity or communication on the computer system or network.
The owner, or its agents, may retrieve any information stored within the
```

Malware propagation inside user plane

Allows malicious data[1] (popular malware and binaries)

- Inside 100 SMS, and IP payload
- malware, spam and phishing content is allowed to propagate inside the mobile network and delivered to IoT devices
- No spam detection filters
- Malware[1] can be sent to arbitrary IoT devices with authorization bypass
- Operators argue that SMS and data against law in some countries



Request URL

```
https://api.iot[redacted]e/api/v1/sms/[redacted]03706960
```

Server response

Code	Details
200	Response body

```
{
  "content": "Arriving early: The package will - be delivered [6] today. Track: | https://gracefinex.com/l/?pk1xwke8ow l",
  "type": "MT",
  "status": "sent",
  "timestamp": "2022-01-18T08:16:08+01:00",
  "created": "2022-01-18T08:16:09+01:00",
  "updated": "2022-01-18T08:16:08+01:00"
}
```

[1] https://www.kaspersky.com/resource-center/threats/sms-attacks

Vulnerabilities in IoT service platforms (5)

(authorization, injection and code execution)

Broken authorization while sending downlink message

IP address not validated for `/ping` API

- The IoT user can send PING message using `/ping` API to communicate with IoT devices over IP layer.
  - User inputs `Ipaddress` of the target device that is assigned internally by the 4G/5G core
- Due to an authorization bug in the platform, an attacker can insert a victim's `IPaddress` in the `/ping` API request and send to the IoT device
  - Required that target/victim device is hosted on the same IoT service platform
- IoT device responds to ping operation (IPV4) with a ping reply. (upto 200 devices available)
- Similarly, port scans can be performed on target device and inject malicious IP packets into the device.
- Impact:
  - increase data consumption over radio interface, billing and charging to victim's account
  - battery drain for low-powered IoT devices, and eventually a DoS.
- Fix: Strict authorization checks for every API parameter/object level.

Private details of SIM and customer are exposed over webhook

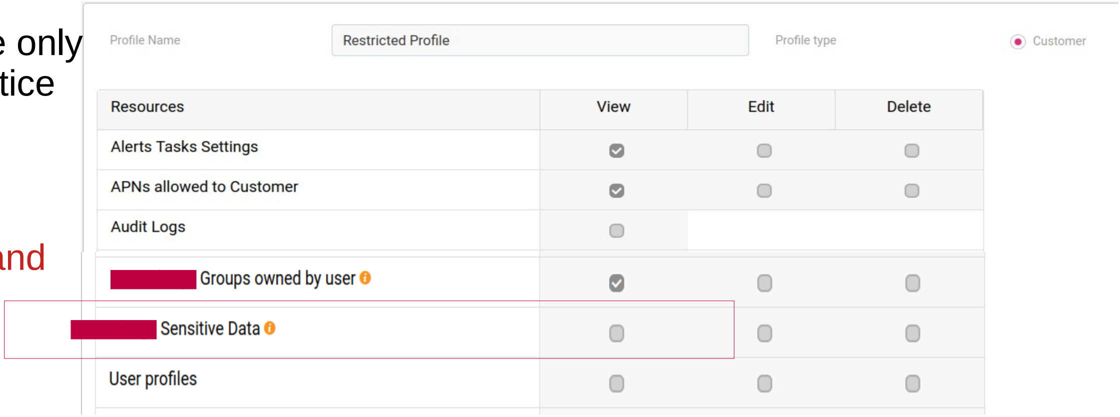
SIM PIN, PUK and subscriber details exposed

- While sending SMS using API, the HTTP response sent to a user-defined Webhook (URL) exposes user's private information
  - Private info: Billing details, subscriber plan and many other sensitive details linked to SIM card (identities, PIN1,PIN2, PUK, Opc, SQN, location, HLR ID).
  - Providers argue that some business cases require such sensitive information in the response
- BGP hijacking[1] to steal all the data exposed over a HTTP Webhook
- Fix: use only HTTPS webhook, and eliminate sending SIM card private info to customer over the Internet

[1] What is bgp hijacking? https://www.cloudflare.com/ko-kr/learning/security/glossary/bgp-hijacking

Access control misconfiguration

- Sensitive Data (like SGSN IP address)
  - Visible to API user in restricted profile (even though view permissions unchecked by administrator)
  - API manual says sensitive data is accessible only to administrator, but fail to implement in practice
  - Other parameters may also be affected with access-control bug, but not verified
  - Discrepancies between API documentation and software implementation.



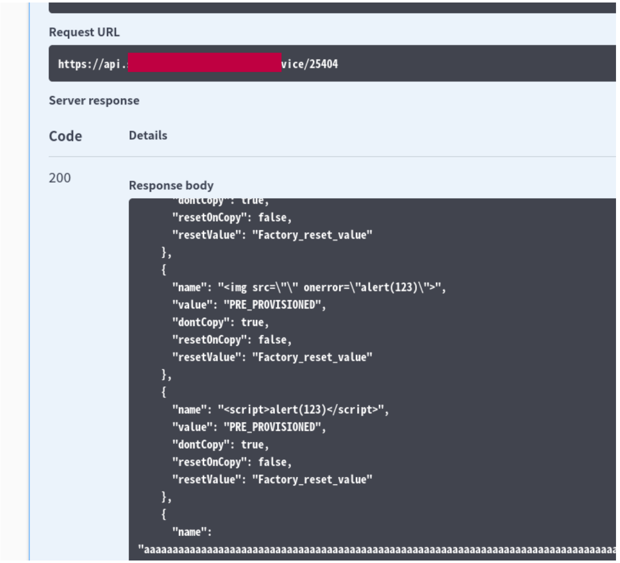
Profile Name: Restricted Profile

Profile type: ● Customer

Resources	View	Edit	Delete
Alerts Tasks Settings	☑	□	□
APNs allowed to Customer	☑	□	□
Audit Logs	□		
[redacted] Groups owned by user ⓘ	☑	□	□
[redacted] Sensitive Data ⓘ	□	□	□
User profiles	□	□	□

Script Injection

- High probability for a code execution attack
  - Many parameters accept tampered and malicious inputs
  - Accepts commands and scripts as API objects
    - <script>Alert(123)</script>
  - This may lead to persistent XSS and injection attacks
  - The injected values gets stored in backend DB
    - Can be called by another backend process
    - Or Customer management web application
  - Fix: strict input sanitization for each and every parameter



Request URL

```
https://api.[redacted]vice/25404
```

Server response

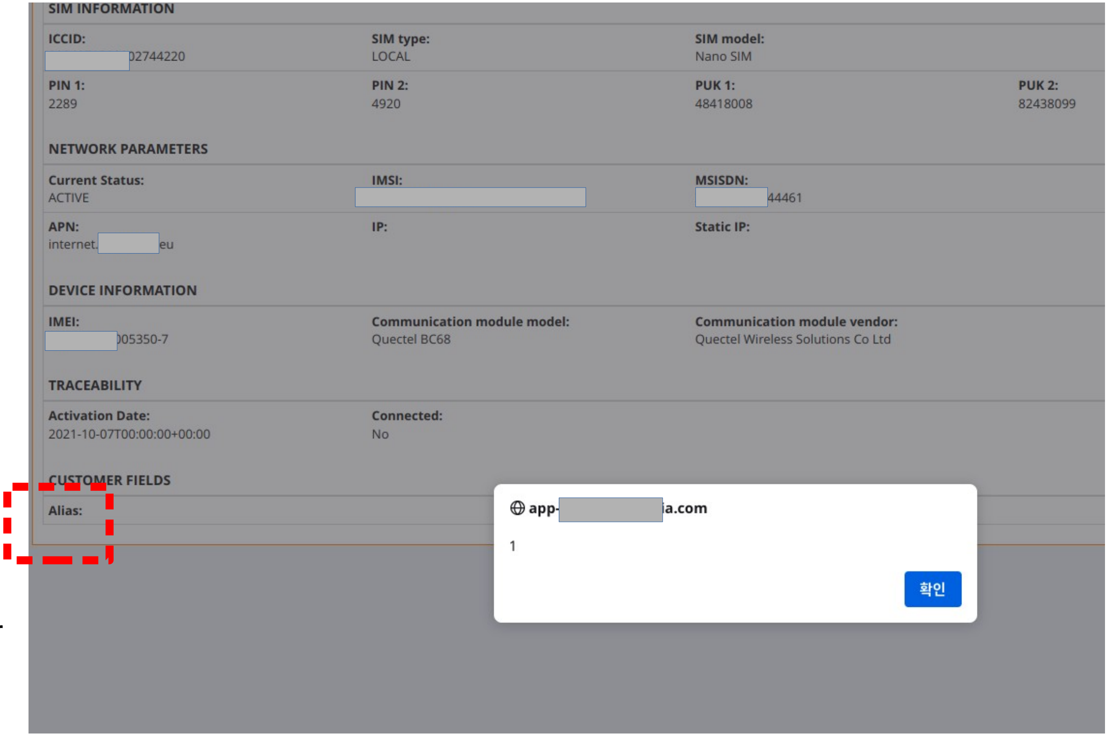
Code Details

200 Response body

```
      "dontCopy": true,
      "resetOnCopy": false,
      "resetValue": "Factory_reset_value"
    },
    {
      "name": "<img src=\"\" onerror=\"alert(123)\">",
      "value": "PRE_PROVISIONED",
      "dontCopy": true,
      "resetOnCopy": false,
      "resetValue": "Factory_reset_value"
    },
    {
      "name": "<script>alert(123)</script>",
      "value": "PRE_PROVISIONED",
      "dontCopy": true,
      "resetOnCopy": false,
      "resetValue": "Factory_reset_value"
    },
    {
      "name":
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

XSS execution

- Code Injection
  - Via API on the service platform
  - e.g., the *`Alias`* is an alternate name of the SIM card and can be given as input from the user
  - Allows script and arbitrary code
- Code Execution
  - via the *IoT connectivity management platform*
  - *`Alias`* parameter is shared between both platforms and inject script is triggered on the web interface leading to code execution
  - With authorization bypass, attacker can inject code into another customer's platform and trigger it



SIM INFORMATION

ICCID: [redacted]02744220
SIM type: LOCAL
SIM model: Nano SIM

PIN 1: 2289
PIN 2: 4920
PUK 1: 48418008
PUK 2: 82438099

NETWORK PARAMETERS

Current Status: ACTIVE
IMSI: [redacted]
MSISDN: [redacted]44461

APN: internet.[redacted]eu
IP:
Static IP:

DEVICE INFORMATION

IMEI: [redacted]05350-7
Communication module model: Quectel BC68
Communication module vendor: Quectel Wireless Solutions Co Ltd

TRACEABILITY

Activation Date: 2021-10-07T00:00:00+00:00
Connected: No

CUSTOMER FIELDS

Alias:

app-[redacted]ia.com

1

Responsible disclosure

- Responsibly disclosed our findings to the affected IoT service providers and operators
- Received positive acknowledgments and confirmation of the vulnerabilities, and appreciation for our efforts to make the exposure services more secure.
- Operators confirmed that our testing methods never caused any damage to their services and infrastructure.
- Three of the tested service providers indicated that, injection vulnerabilities discovered in our findings remained hidden during their internal penetration testing exercise.
- We do not disclose any of the API and provider/operator names

Summary of security analysis

- Oauth and TLS is used in majority of platform (5/9) but not all of them.
- Only 2 out of 9 IoT platforms are not affected with serious vulnerabilities and API risks
- IMSI is exposed outside of 3GPP network, same practice may apply for 5G IMSI (SUPI)
- Lack of rate-limits, strong password policies
- Internal software information and core network IP addresses are exposed
- Authorization vulnerability can destroy the IoT devices and the network
- Script/code injection vulnerability found in many platforms, and is missed when a internal pen-testing
- SMS and IP content inspection is not present in mobile and IoT networks
- Attacker can easily obtain access to IoT service platforms and service APIs with forged identity

Security measures

- KYC – strict Know Your Customer check before issue access to IoT service platforms
- Customized API design : limit the number of APIs available for each use-case or business partner – reducing attack surface
- Reduced data exposure over several zones
  - Private identifiers like IMSI and SUPI should be replaced with random identifiers
  - Information sent over Webhook, in API responses, and error messages
- Rate limits should be mandatory and smart algorithms to detect malicious behavior
- Strict Input validation and sanitization for each every parameter taken as input from user
- Analytics-based security including logging and real-time monitoring

Key takeaways

- Opening new door on mobile networks – strict identity and access control, zero-trust
- Standard Oauth and TLS mechanisms wont help achieve full security
- Insecure API Design/Configuration **=** risk for mobile core and IoT devices
- Telecom exposure API risks are new: application **logic flaws** – require rigorous application specific tests (not using general API security scanners)
- Firewalls won’t always help – need security-by-design and testing into CI/CD pipelines
- APIs in Telecom is new **and require a Telecom API top 10** to help developers and operators understand the security risks

Questions? Concerns? Comments?

Write me:

(altaf.shaik@fastiot.org)