SIM Cards

The new frontier of mobile exploitation, back from the past

Giovanni De Luca

DEFCON ROME

About me

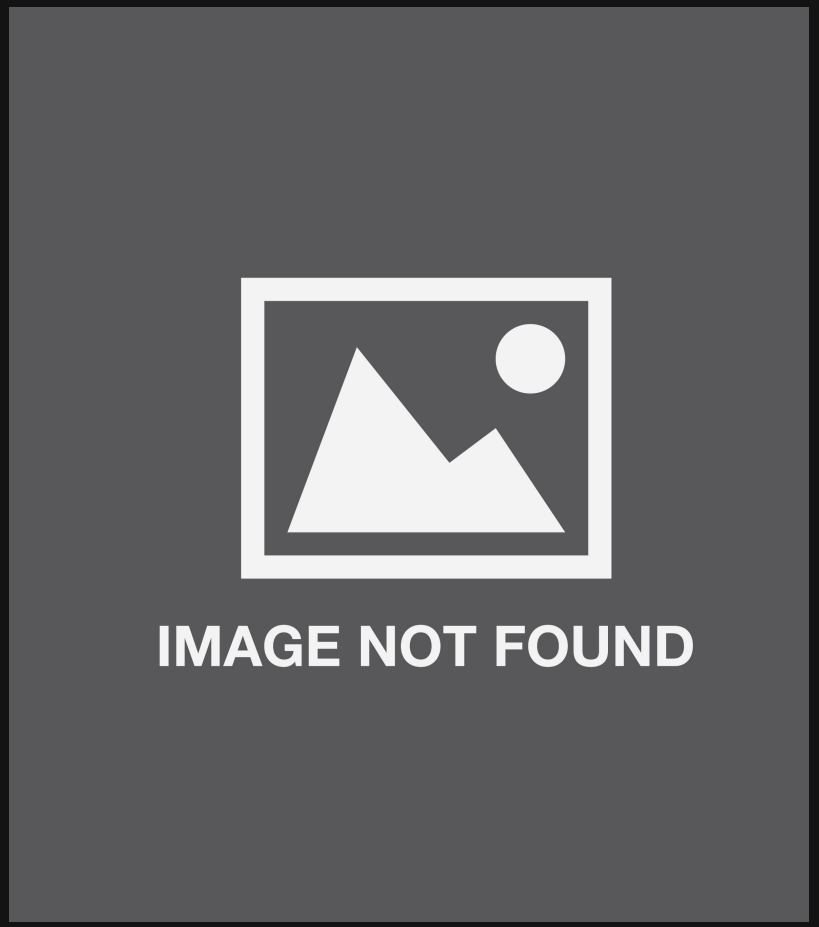
Giovanni De Luca

Student, Engineering in Computer Science Master's Degree @Sapienza.

Mozilla Italia volunteer developer and occasional speaker at tech events.

Infosec, AI, computer graphics, gamedev, operating systems enthusiast, occasional contributor to several open source software projects.

Guitarist, 3D artist.



jotaro-sama

@gdl_jotaro_sama

Outline

- SIM cards 101
- SIM card security
- Rooting (old) SIM cards
- Simjacker – Next generation spying over mobile

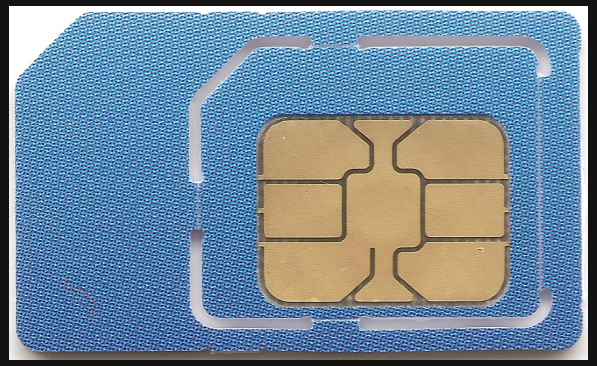
Disclaimer

The contents of this talk are potentially DANGEROUS!

Attacks are shown here **ONLY** for the purposes of self-defense and public awareness

I do **NOT** take responsibility for your own actions

SIM cards 101



- SIM stands for Subscriber Identity Module
- Intended to securely store the international mobile subscriber identity (IMSI) number and its related key (K_i)
- The physical smart card is actually called UICC (universal integrated circuit card), the SIM is theoretically just the GSM (or whatever network) applet running on it
- For simplicity, we'll use SIM to refer to the whole card

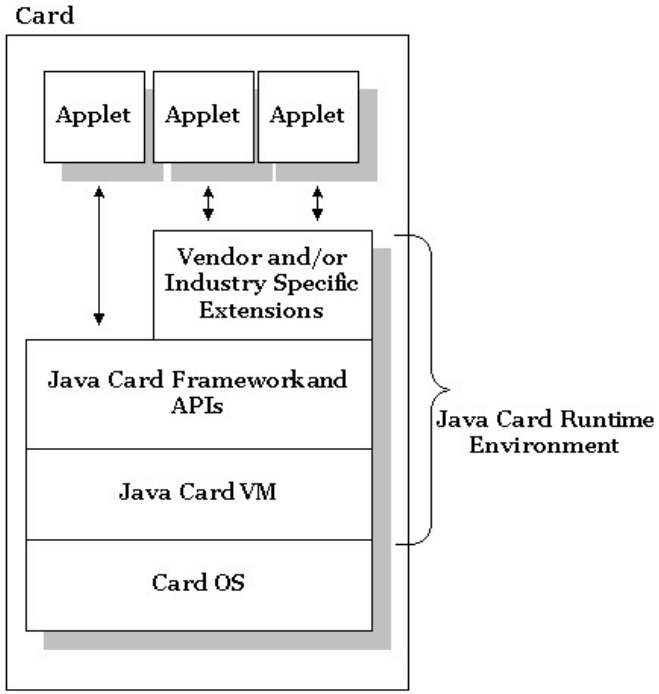
What’s on a SIM card?

- Identity (IMSI) and symmetric key (K_i)
- A CPU (not unlike what you can find on some Arduino boards), ROM to store OS and apps, EEPROM for storage (with limited write/erase cycles), RAM
- A filesystem (storing session keys, contacts, SMS)
- A JVM and custom Java apps
  - Wait, what?

Yes, SIM Cards are fully programmable computer systems

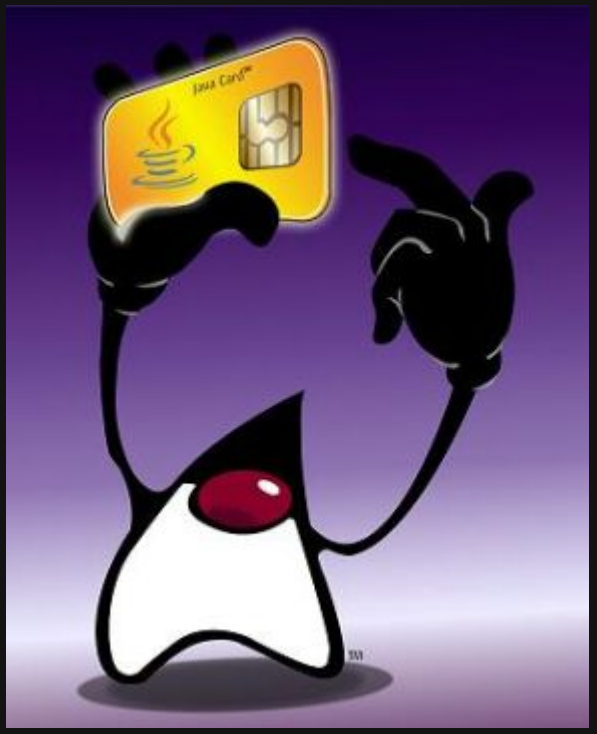
Therapist: Java on SIM cards does not exist, it can't hurt you

Java on SIM cards:



SIM Applications

- Programmed in Java Card (a limited version of Java)
- There is a master card manager app, sometimes it’s the GSM (which is just one of the applets)
- They can display a simple UI to phones, launch URLs, send SMSs and do other things (more on this later).
- They talk to the phone UI via STK API, and are loaded and managed according to the GlobalPlatform standard
- They can be silently remotely installed by carrier

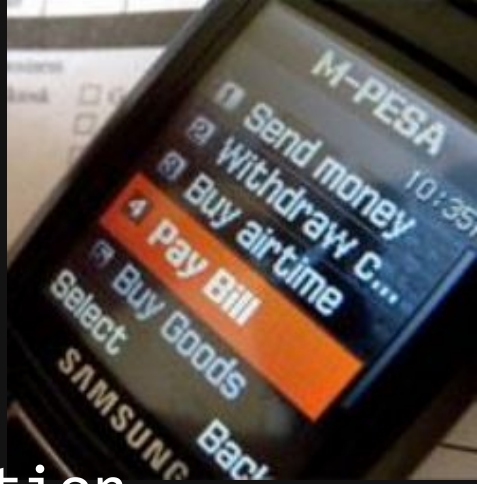


SIM card security

SIM cards have many protection mechanisms:

- Through physical smartcard security mechanisms, modern ones cannot have their contents read. They are thought as «black boxes», with which you can communicate only via packets called APDUs
- PIN and PUC numbers
- SIM authentication by cryptographic hash function
- Secure communication with OTA servers (with encryption and/or signature)
- Separation of apps (Java sandboxing)

SIM card security



- SIMs were intended for secure storage and computation (in some countries there are SIM-based banking apps)
- The SIM can access the networking hardware without necessarily going through the phone itself
- Carriers can send «silent» SMS to install new apps, and to avoid malicious use, messages between the OTA server and the SIM are either encrypted or signed. The most common legacy way was to just sign them with DES and an OTA key.
- Note: OTA keys and all secrets generated on the SIM are based on K_i

Rooting (old) SIM cards

- In 2013, Karsten Nohl showed at Black Hat how easy it could be to exploit a couple weaknesses present in many cards
- By sending a badly signed request to a SIM, in some cases the SIM would reply with a signed error message
- The DES signature can be cracked with rainbow tables in a reasonable time to retrieve the OTA key
- Encrypting solves nothing if still done with DES
- 3DES (which many carriers are upgrading to) solves the issue but only if used properly

Weaknesses in 3DES-using cards

- Using only one key: Some carriers keep the column in the OTA server DB able to only store one key
- Using the default developer key for all cards (yes some do)
- Downgrade attacks

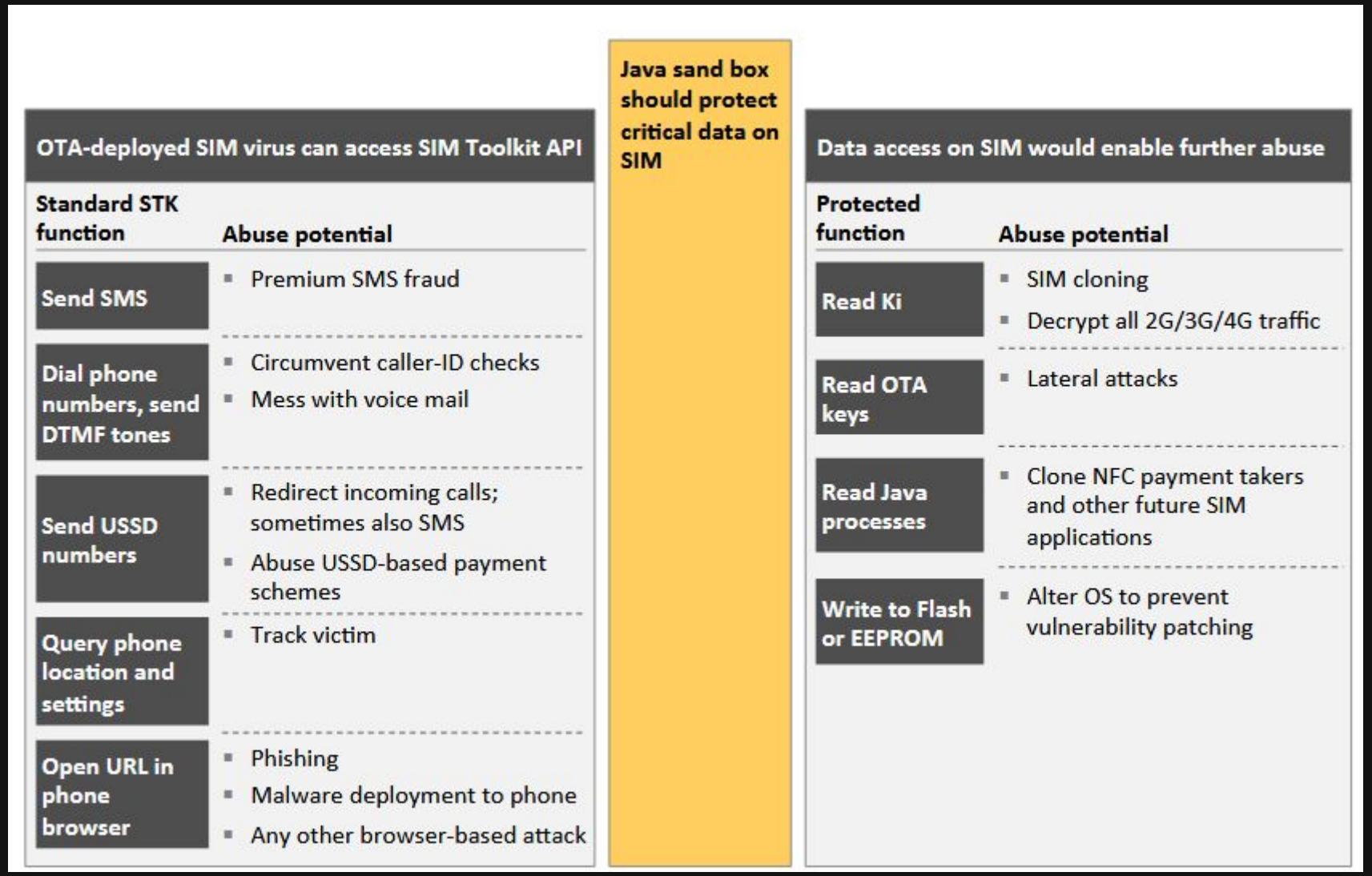
Downgrade attacks

- Some cards, if requested to use a certain security mechanism and it’s below their default security level, will answer no BUT will use it to sign the «No I’m not gonna use it» message. You ask to use DES and get the first of the 3 keys. Then you ask to use 2-key 3DES and get the second (no rainbow tables allowed), then you ask to use 3-key 3DES and crack the last one.

Exfiltrating data

- A Java virus loaded once you have the OTA keys already has a lot of potential, but through sandbox escaping (in 2013, available on 2 JVM implementations making 80% of the market share) everything on the SIM can be accessed by a Java app on the SIM, and all sort of stuff can be done
- This does not include data on the phone, or access to the Android/iOS system, the SIM is a separated machine

Exfiltrating data



Mitigation and industry response

- There were many ways to mitigate this:
  - Deactivate OTA (why can’t the user decide to opt-out?)
  - Use 3DES or OTA keys
  - Don’t send crypto texts as reply to badly formed requests
- The industry seemed to take interest in working together to fix this, and the talk had a good response

Simjacker:
Next generation spying over mobile

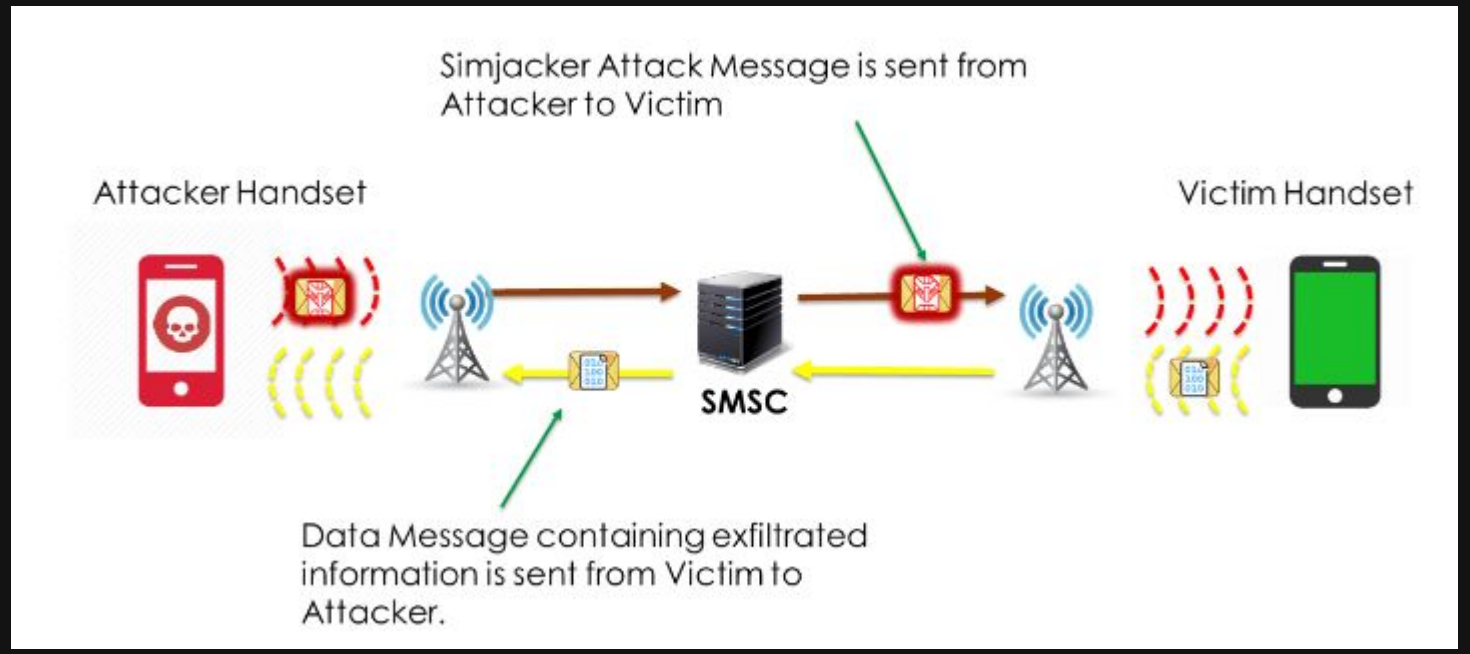
- New and recent attack
- Exploits the S@T Browser application on SIMs
- Its default security does not require any authentication
- Attacker can execute functionalities unbeknownst to the mobile phone user

Simjacker:
Next generation spying over mobile

- Tracked thousands of mobile numbers over the months (possibly years)
- Executed with many variations in sending the attack, exfiltrating data, structure in the request and extracted information, techniques used to avoid detection

The Attack

- Attack stage: An attack message containing STK instructions requesting current Cell-ID and IMEI is sent to the device
- Exfiltration stage: A data SMS is sent to the exfiltration address
- All of this goes unnoticed by the phone user



S@T Browser

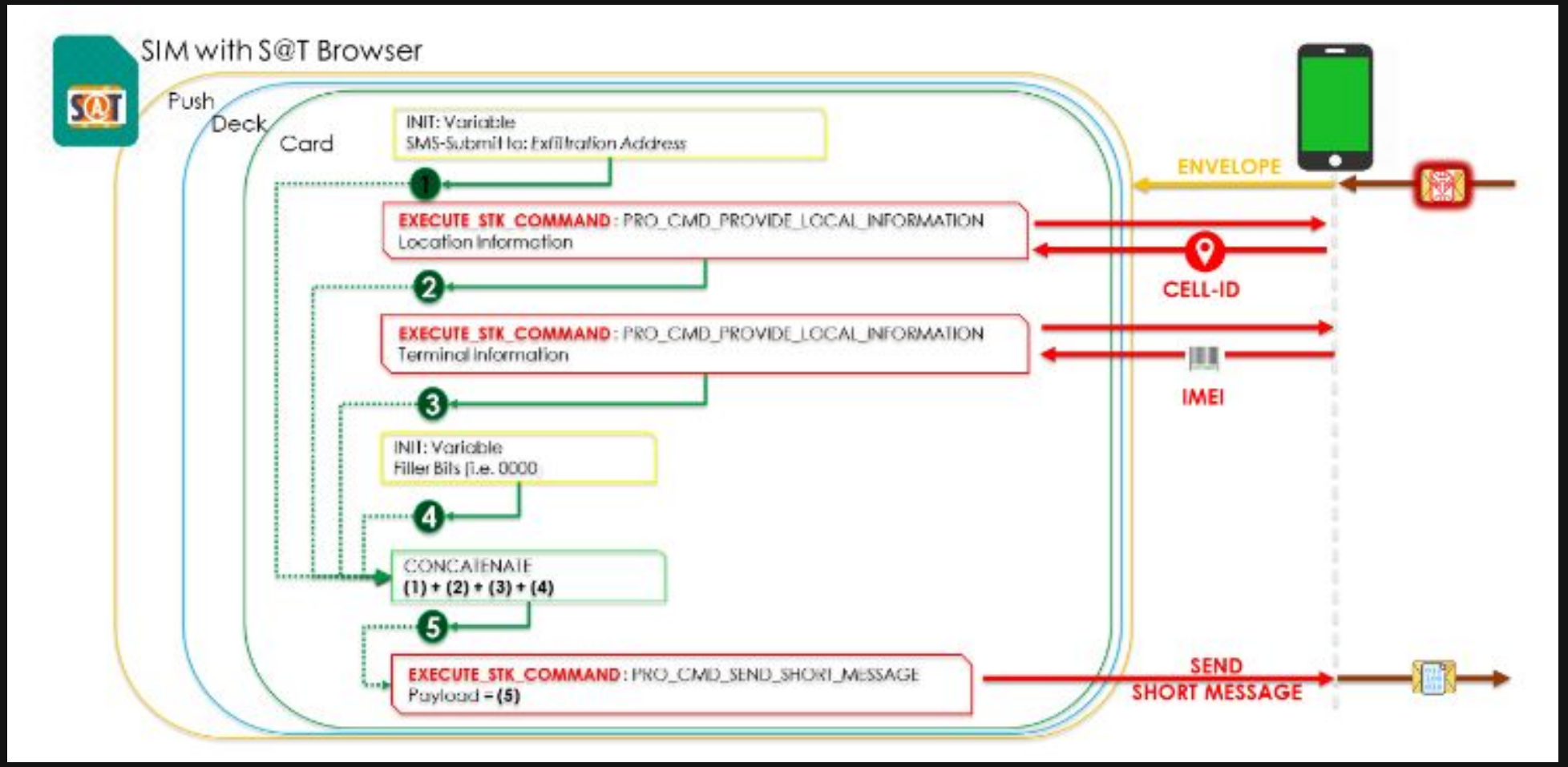
- For the attack to succeed, device needs to be able to receive SIM OTA SMS, and must have the S@T Browser technology on the SIM
- S@T Browser supports two MSL: No security and 3DES
- No recommended security level for push messages!
- If “no security” is used, anyone can send push messages to the devices without any cryptographic authentication

The following security levels shall be supported by the S@T browser:

SPI	*KIc*	*KID*	*DESCRIPTION*	*NOTES*
0x0000	0x00	0x00	No security applied	Shall be supported for incoming (MT) and outgoing (MO) messages. This security level is not recommended for Administration protocol.
0x1200	0x00	0xX5	Triple DES Cryptographic Checksum (8-byted MAC); counter higher	Shall be supported for incoming (MT) messages. This security level is not recommended for Pull protocol.

Typical Simjacker message

Those in red are STK commands, the other S@T commands



Variables management

Variables in the Simjacker messages are set as temporary, meaning they are cleared when:

- S@T Browser goes into IDLE state
- S@T Browser starts a card with ResetVar flag
- A High priority push is received

Variables management

When multiple messages are sent in a quick succession, high priority push is used from the 2nd one on for this reason, and ResetVar is often specified when creating the cards.

Low priority push messages however were used the 99.23% of the time in the time period the researchers observed

Variants of the attack

- In the massive usage of the attack, over 1000 encoding combinations in the header were tried (probably to try avoid mobile operator defences)
- Over 860 sub-variants in the actual SMS packet

Potentially affected countries

These are the countries using S@T with the no security settings:

Additional functionalities

- Other than the ones exploited in Simjacker, the S@T Browser has access to many other STK commands, which could result in:
- Fraud Applications, Advanced Location Tracking, Assistance in Malware Deployment, Denial of Service, Information Retrieval, Misinformation

- REFRESH
- MORE TIME
- POLL INTERVAL
- POLLING OFF
- SETUP EVENT LIST
- SET UP CALL
- SEND SS
- SEND USSD
- SEND SMS
- SEND DTMF
- LAUNCH BROWSER
- PLAY TONE
- DISPLAY TEXT
- GET INKEY
- GET INPUT
- SELECT ITEM
- SET UP MENU
- PROVIDE LOCAL INFO
- TIMER MANAGEMENT
- SETUP IDLE MODE TEXT

Other vulnerable applets

- In theory, any SIM card application with poor security specifications could be targeted by attacks as S@T browser did
- Lately, the WIB (Wireless Internet Browser) was found to be able to be exploited

Wireless Internet Browser

- Technology specified by SmartTrust for SIM based browsing
- Specification not generally available, but implementation documents of some companies can be found
- Most documents at least state that “no security MSL should be used for testing only, since it provides no protection whatsoever”
- However, some countries are using no security OTA for WIB

Mitigation

- Filtering on a network level (even though attackers were smart enough to evade those in place until now)
- Improvements in the security of the affected S@T and WIB applications, as the filtering for incoming and outgoing messages does not depend only on the MSL for the application but on the application’s configuration

References

- https://media.blackhat.com/us-13/us-13-Nohl-Rooting-SIM-cards-Slides.pdf
- https://www.defcon.org/images/defcon-21/dc-21-presentations/Koscher-Butler/DEFCON-21-Koscher-Butler-The-Secret-Life-of-SIM-Cards-Updated.pdf
- https://simjacker.com/downloads/technicalpapers/AdaptiveMobile_Security_Simjacker_Technical_Paper_v1.01.pdf#page=4&zoom=auto,-47,324
- https://ginnoslab.org/2019/09/21/wibattack-vulnerability-in-wib-sim-browser-can-let-attackers-globally-take-control-of-hundreds-of-millions-of-the-victim-mobile-phones-worldwide-to-make-a-phone-call-send-sms-to-any-phone-numbers/