Assessing the Mobile Device Platform to Assure DFS Security

Kevin Butler, Director, Florida Institute for Cybersecurity Research, University of Florida

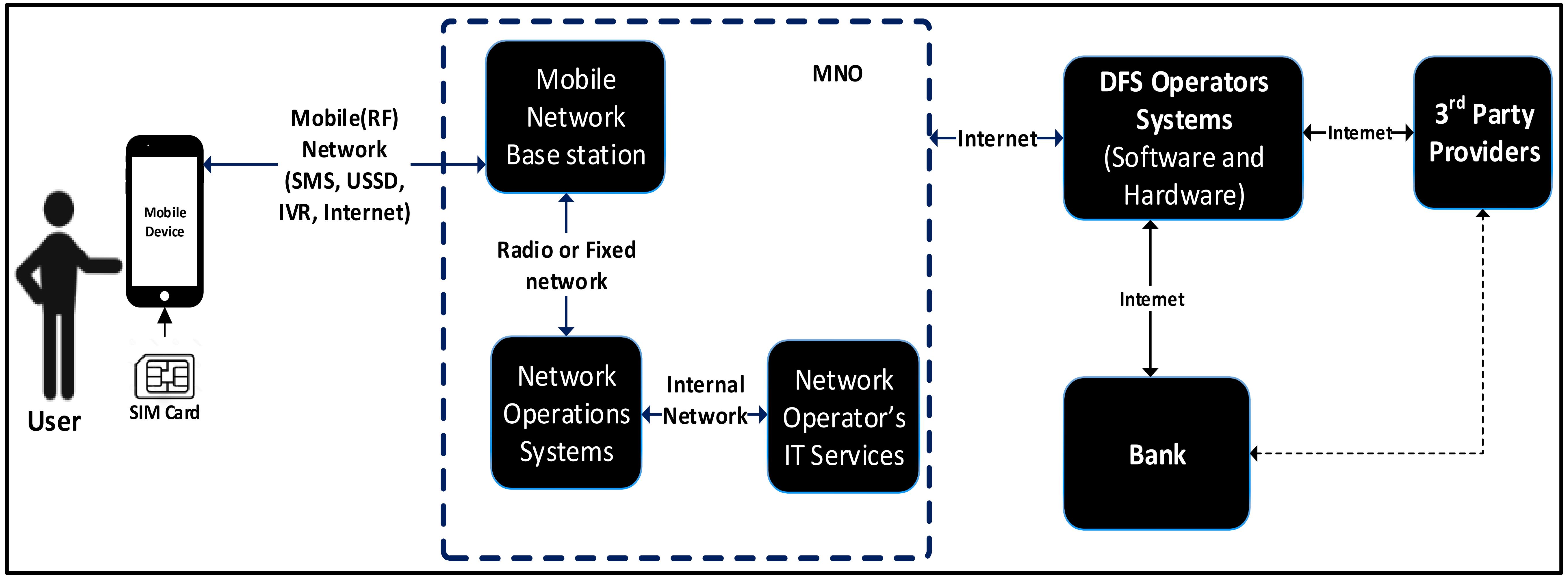
ITU Regional Digital Financial Services Security Clinic for Asia-Pacific Region

Seoul, Republic of Korea

April 24, 2024

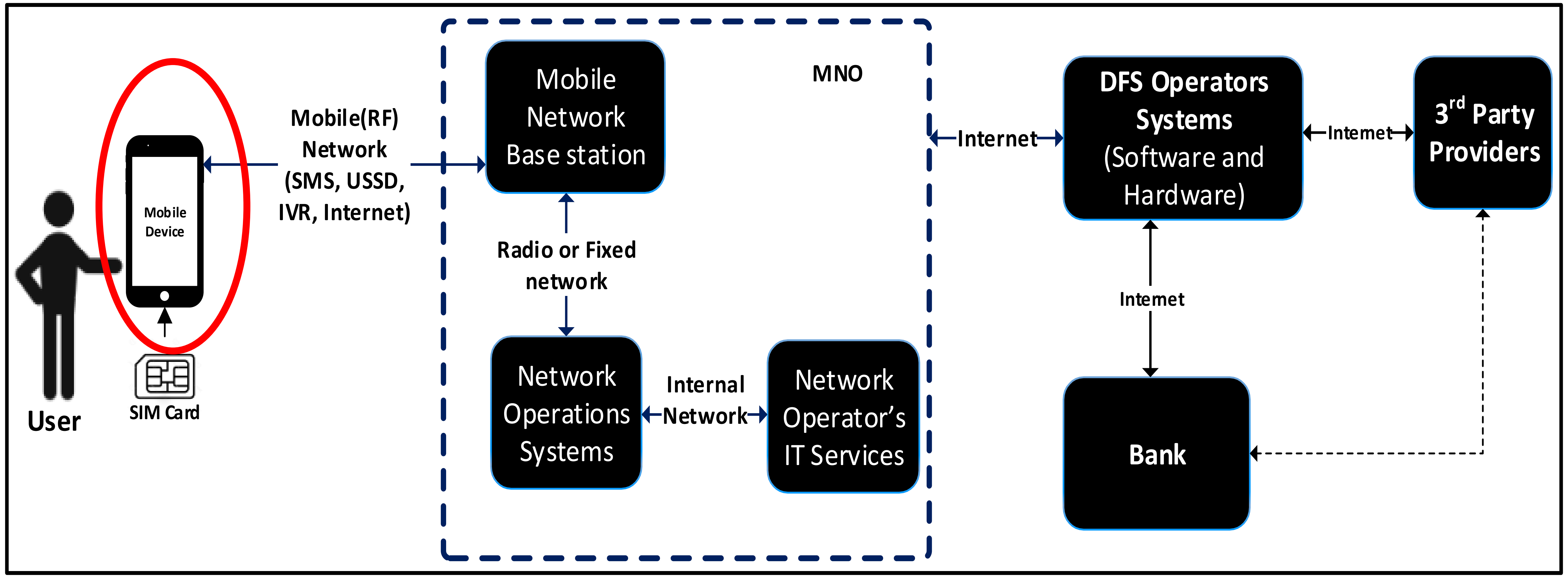
DFS Ecosystem Stakeholders

- Regulators
- Mobile network operators
- DFS providers
- Customers
- External service providers


User
Mobile Device
SIM Card
Mobile(RF) Network (SMS, USSD, IVR, Internet)
MNO
Mobile Network Base station
Radio or Fixed network
Network Operations Systems
Internal Network
Network Operator's IT Services
Internet
DFS Operators Systems (Software and Hardware)
Intemet
3rd Party Providers
Intemet
Bank

DFS Ecosystem Stakeholders

- Regulators
- Mobile network operators
- DFS providers
- Customers
- External service providers



User
Mobile Device
SIM Card
Mobile(RF) Network (SMS, USSD, IVR, Internet)
MNO
Mobile Network Base station
Radio or Fixed network
Network Operations Systems
Internal Network
Network Operator's IT Services
Internet
DFS Operators Systems (Software and Hardware)
Intemet
3rd Party Providers
Intemet
Bank

X.805 Security Dimensions

Access control: protection against unauthorized use of network resources.

Authentication: methods of confirming the identities of communicating entities.

Non-repudiation: methods to prevent an individual or entity from denying having performed a particular action.

Data confidentiality: protection of data from unauthorized disclosure.

Communication security: assurance that information only flows between authorized endpoints.

Data integrity: protection of the correctness and accuracy of data.

Availability: prevention of denial of authorized access to network elements and data.

Privacy: protection of data information that might be derived from observing network activity.

Analyzing External Attack Surfaces

What modalities can an adversary use to create vulnerabilities in a device?

Three case studies:

Analyzing External Attack Surfaces

What modalities can an adversary use to create vulnerabilities in a device?

Three case studies:

1. Unanticipated use of device commands

Analyzing External Attack Surfaces

What modalities can an adversary use to create vulnerabilities in a device?

Three case studies:

1. Unanticipated use of device commands

2. Exploiting inconsistencies amongst access control mechanisms used within the device

Analyzing External Attack Surfaces

What modalities can an adversary use to create vulnerabilities in a device?

Three case studies:

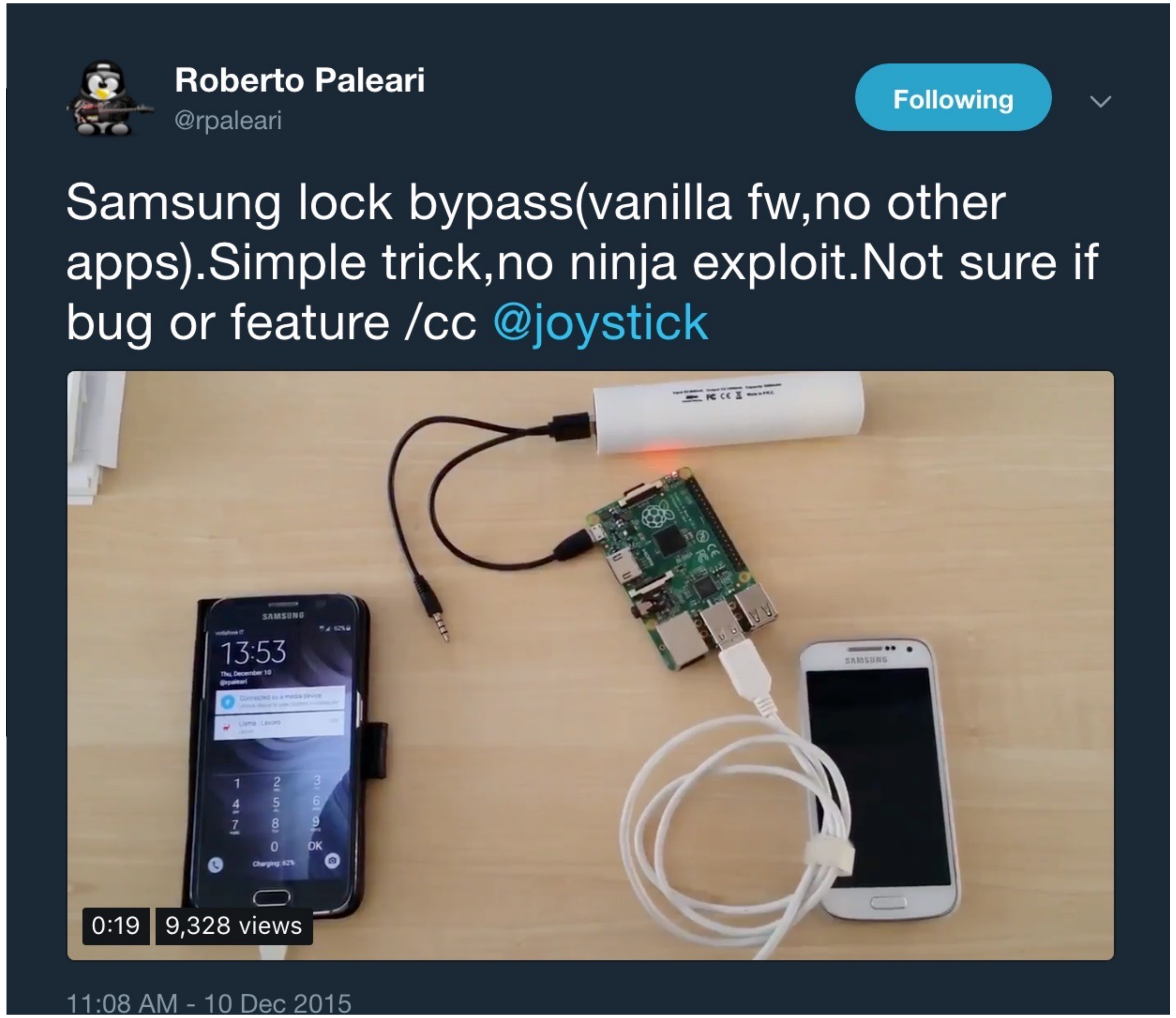
1. Unanticipated use of device commands
2. Exploiting inconsistencies amongst access control mechanisms used within the device
3. Attacks against device elements that are not well understood

AT Commands in Smartphones

AT commands aren’t new

Previous work on smartphones shows that a select few AT commands have an impact

- But we still have no idea…
  - How many commands exist?
  - What their security impact is?
  - What the commands do?



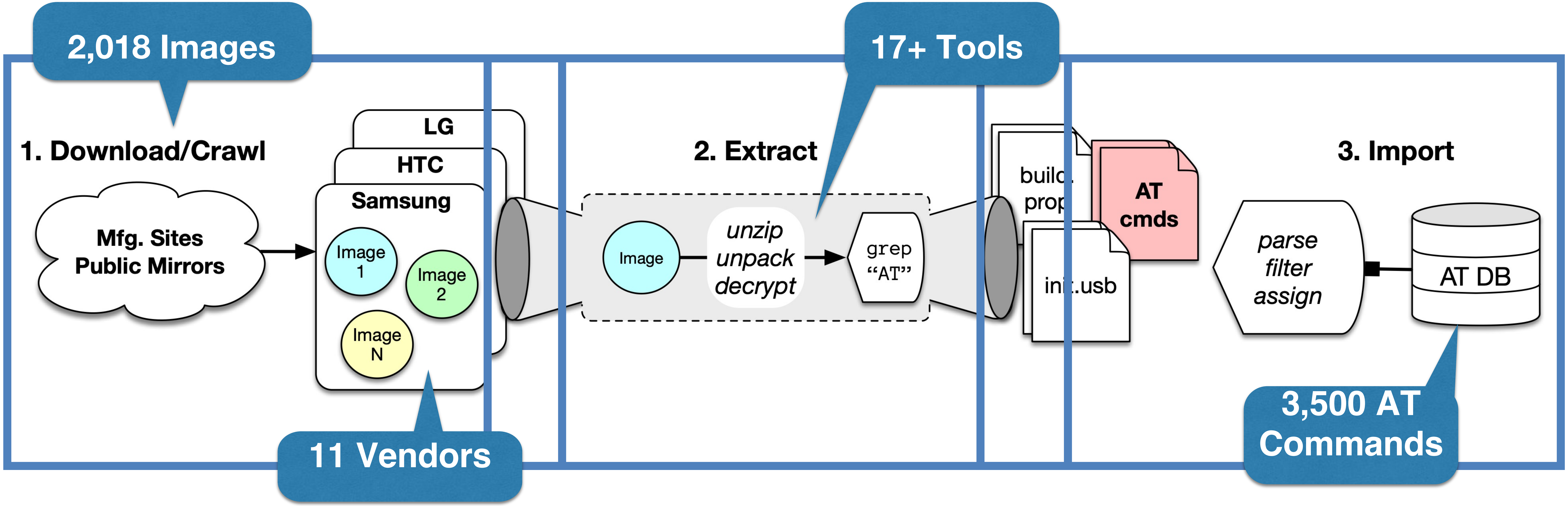
Roberto Paleari
@rpaleari

Following

Samsung lock bypass(vanilla fw,no other apps).Simple trick,no ninja exploit.Not sure if bug or feature /cc @joystick

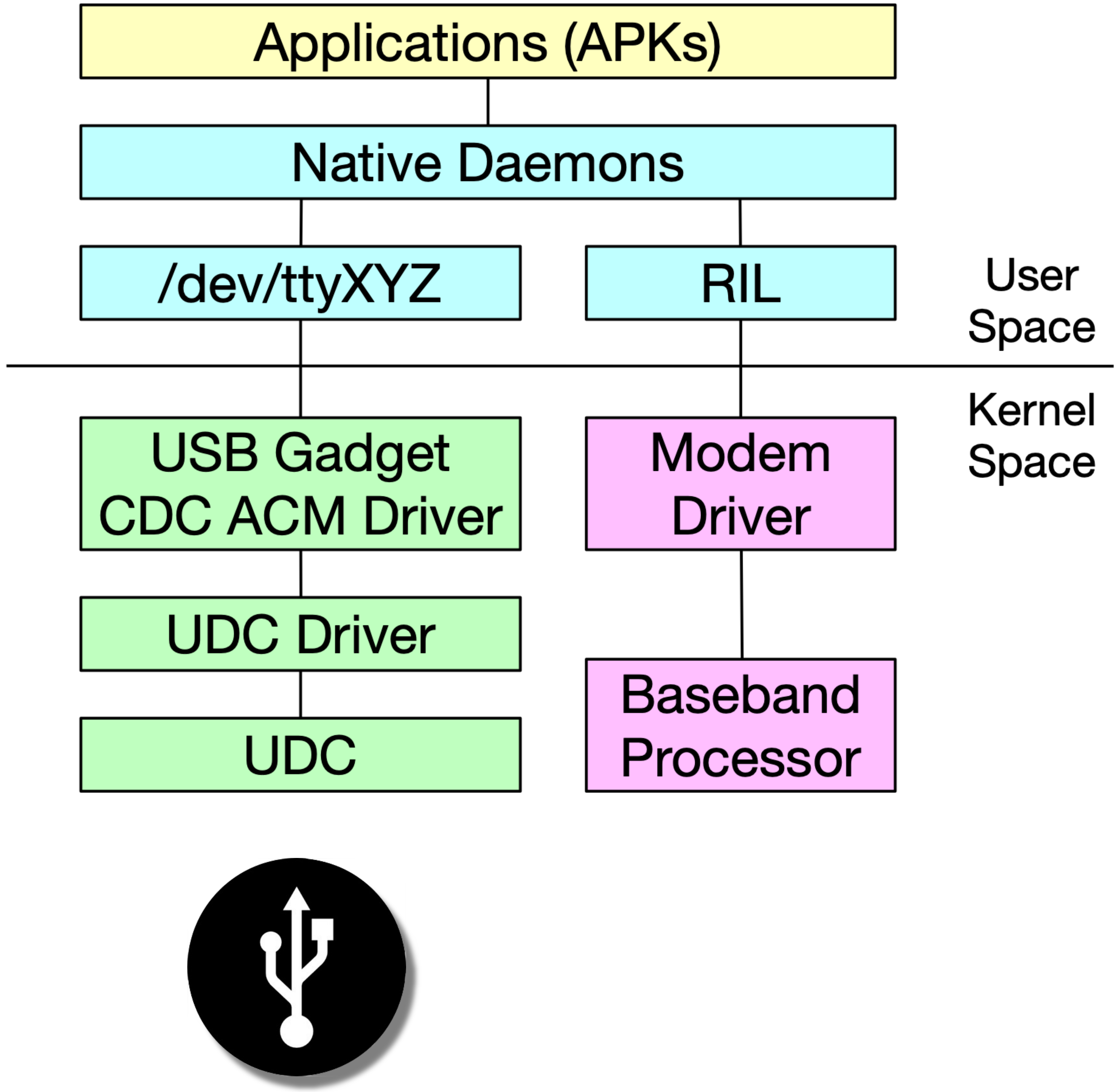
11:08 AM - 10 Dec 2015

Analysis Pipeline



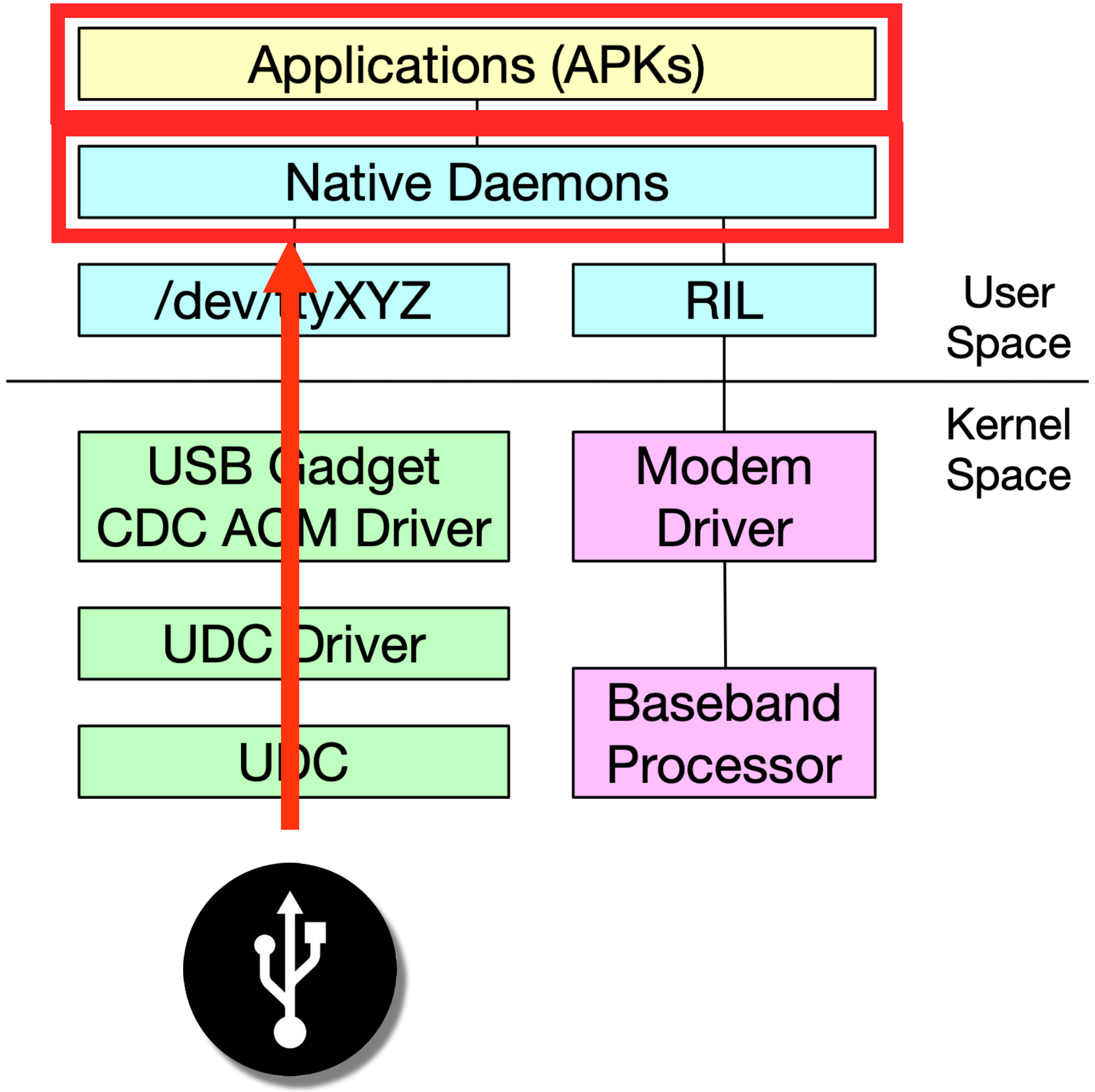
Attack Vector: Modem Interface

- Some smartphones expose "modem interface" over USB



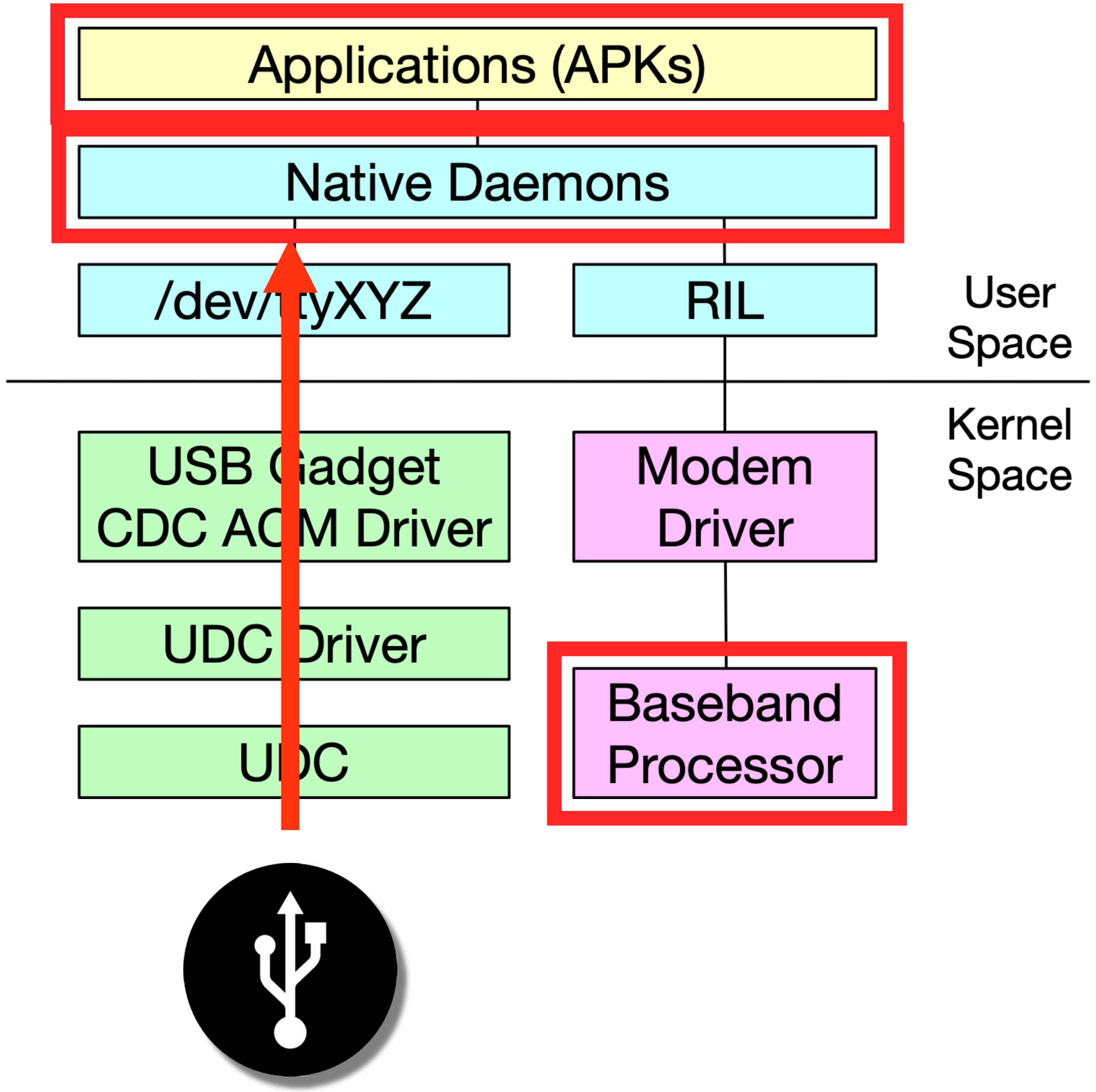
Attack Vector: Modem Interface

- Some smartphones expose "modem interface" over USB
- Commands flow from the USB port to a listening native daemon and either go to the modem or the Android system



Attack Vector: Modem Interface

- Some smartphones expose “modem interface” over USB
- Commands flow from the USB port to a listening native daemon and either go to the modem or the Android system
- Some phones have a “hidden” modem configuration that can be activated externally with usbswitcher



Android Security Bypassing

Make Calls

```
ATD3521174567
```

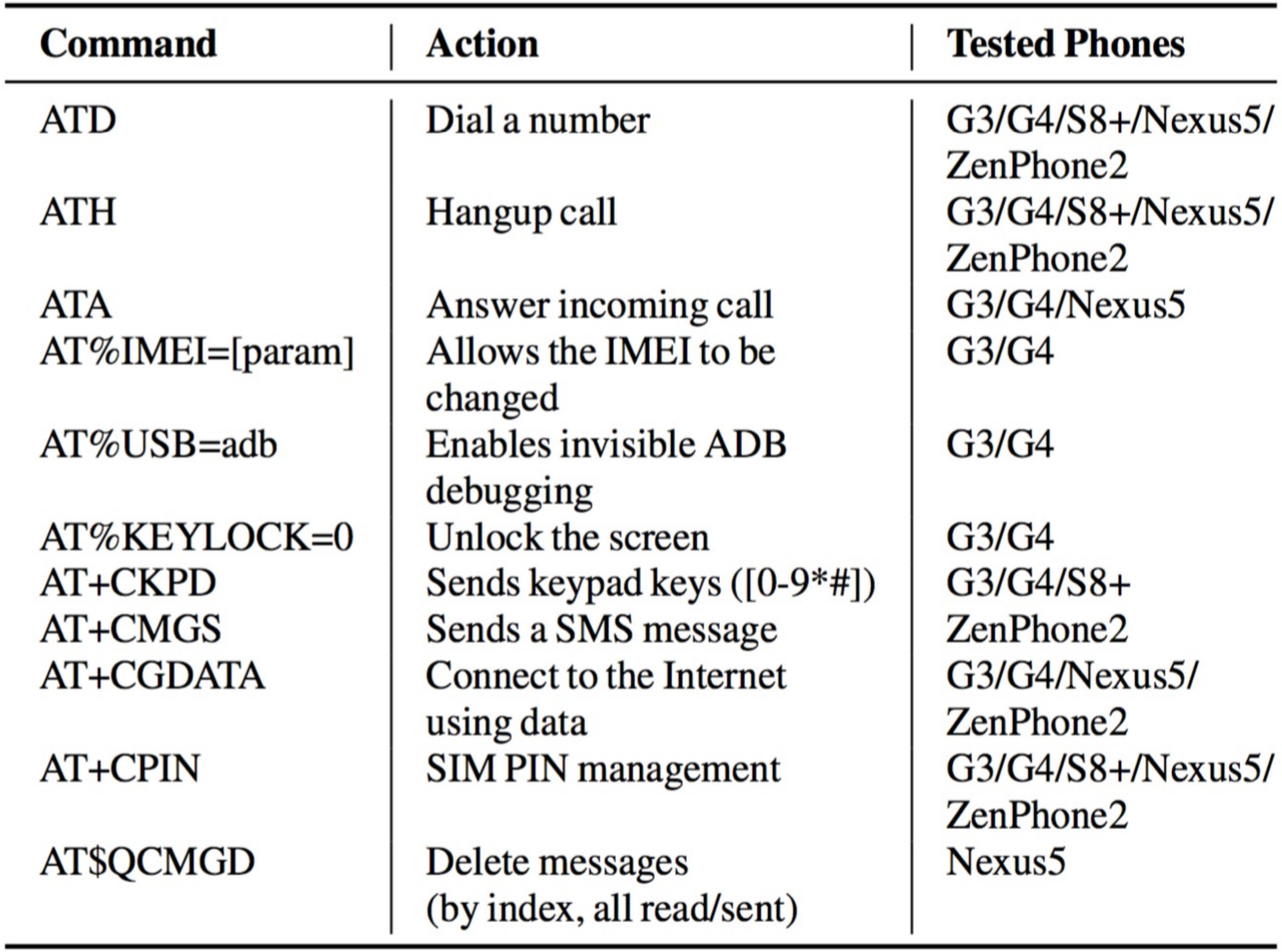
Bypass the lock screen

```
AT%KEYLOCK=0
```

Inject Touch Events

```
AT+CTSA=EVENT,X,Y
```

Results reported to multiple smartphone vendors



Command	Action	Tested Phones
ATD	Dial a number	G3/G4/S8+/Nexus5/ ZenPhone2
ATH	Hangup call	G3/G4/S8+/Nexus5/ ZenPhone2
ATA	Answer incoming call	G3/G4/Nexus5
AT%IMEI=[param]	Allows the IMEI to be changed	G3/G4
AT%USB=adb	Enables invisible ADB debugging	G3/G4
AT%KEYLOCK=0	Unlock the screen	G3/G4
AT+CKPD	Sends keypad keys ([0-9*#])	G3/G4/S8+
AT+CMGS	Sends a SMS message	ZenPhone2
AT+CGDATA	Connect to the Internet using data	G3/G4/Nexus5/ ZenPhone2
AT+CPIN	SIM PIN management	G3/G4/S8+/Nexus5/ ZenPhone2
AT$QCMGD	Delete messages (by index, all read/sent)	Nexus5

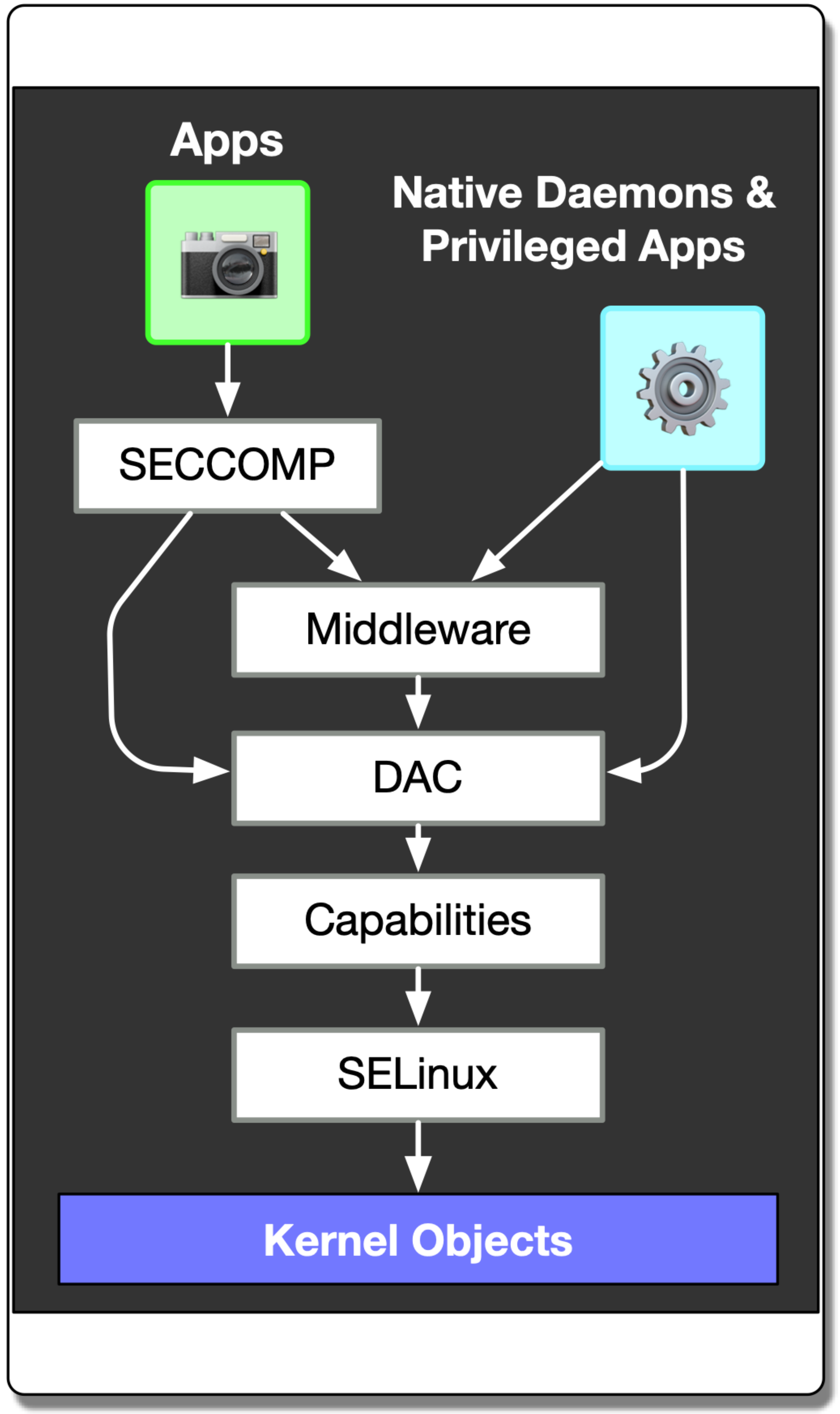
Android Security

Access Control heavyweights

- Linux DAC
- Linux Capabilities
- SELinux / SEAndroid (MAC)

Other

- SECCOMP
- Android Middleware

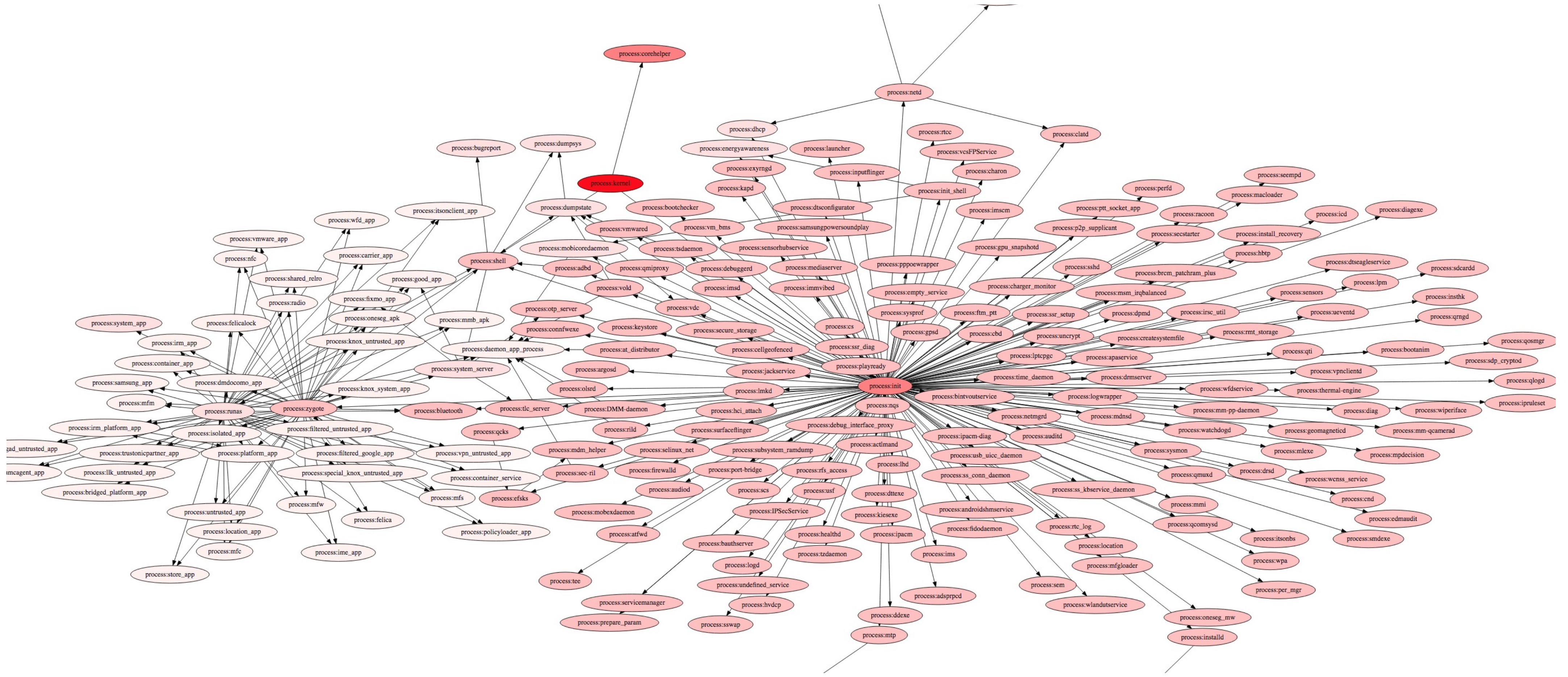


Access control attack surface

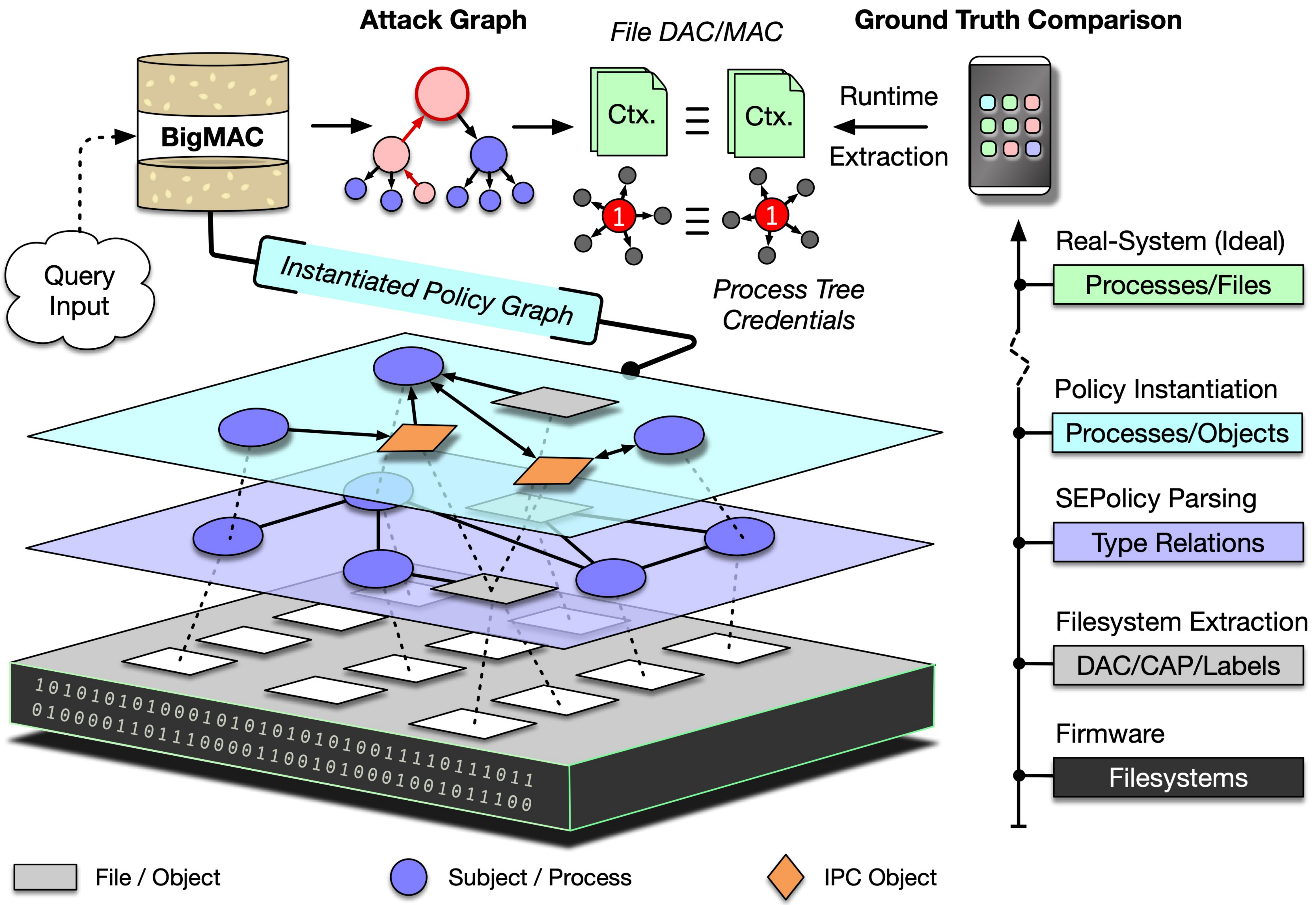
Combine the whole system security model into a unified graph

Query the graph to find attack paths

Example: what objects and processes can an untrusted app talk to?



BigMAC



Attack Graph

File DAC/MAC

Ground Truth Comparison

BigMAC

Query Input

Instantiated Policy Graph

Runtime Extraction

Process Tree Credentials

Real-System (Ideal)

Processes/Files

Policy Instantiation

Processes/Objects

SEPolicy Parsing

Type Relations

Filesystem Extraction

DAC/CAP/Labels

Firmware

Filesystems

File / Object

Subject / Process

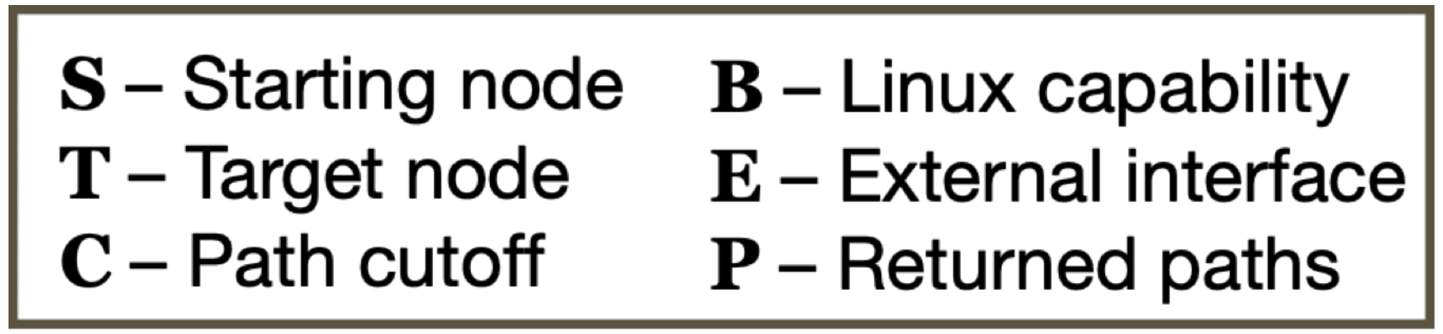
IPC Object

Query Interface

We developed a Prolog query engine to find attack-paths

with MAC, DAC, CAP, and external attack surface filtering

```
query_mac(S,T,C,P).
query_mac_dac(S,T,C,P).
query_mac_dac_cap(S,T,C,B,P).
query_mac_dac_cap_ext(S,T,C,B,E,P).
```



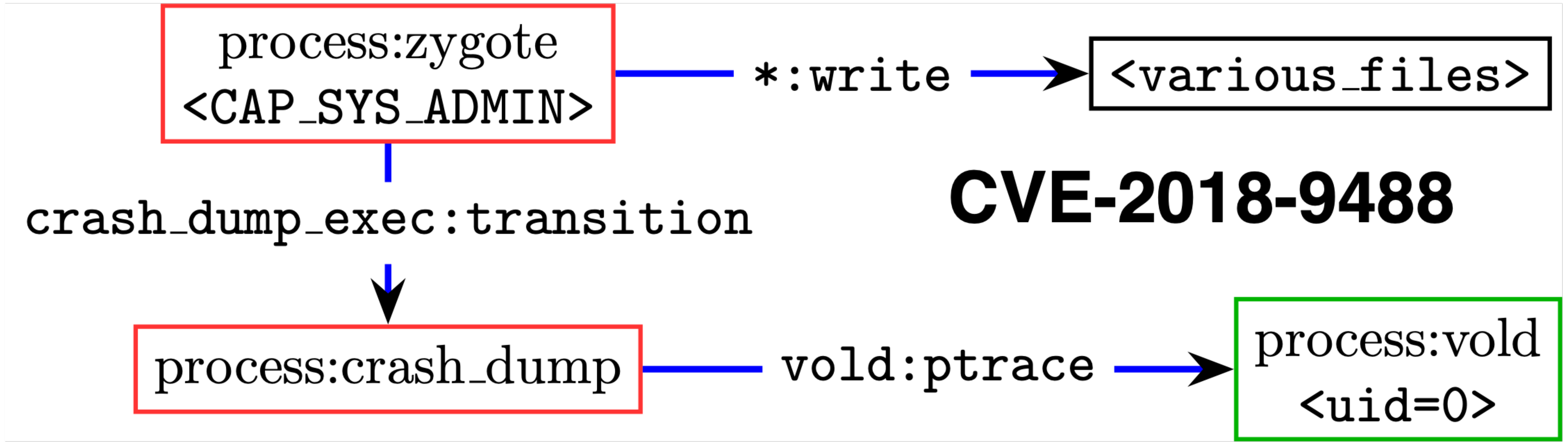
S – Starting node	**B** – Linux capability
T – Target node	**E** – External interface
C – Path cutoff	**P** – Returned paths

As a case study, we ran queries against a 1.3 million edge

Samsung S8+ and a ~2 million edge LG G7 image

Privilege Escalation Analysis

```
#1 query_mac_dac(zygote,vold,3,P).
```

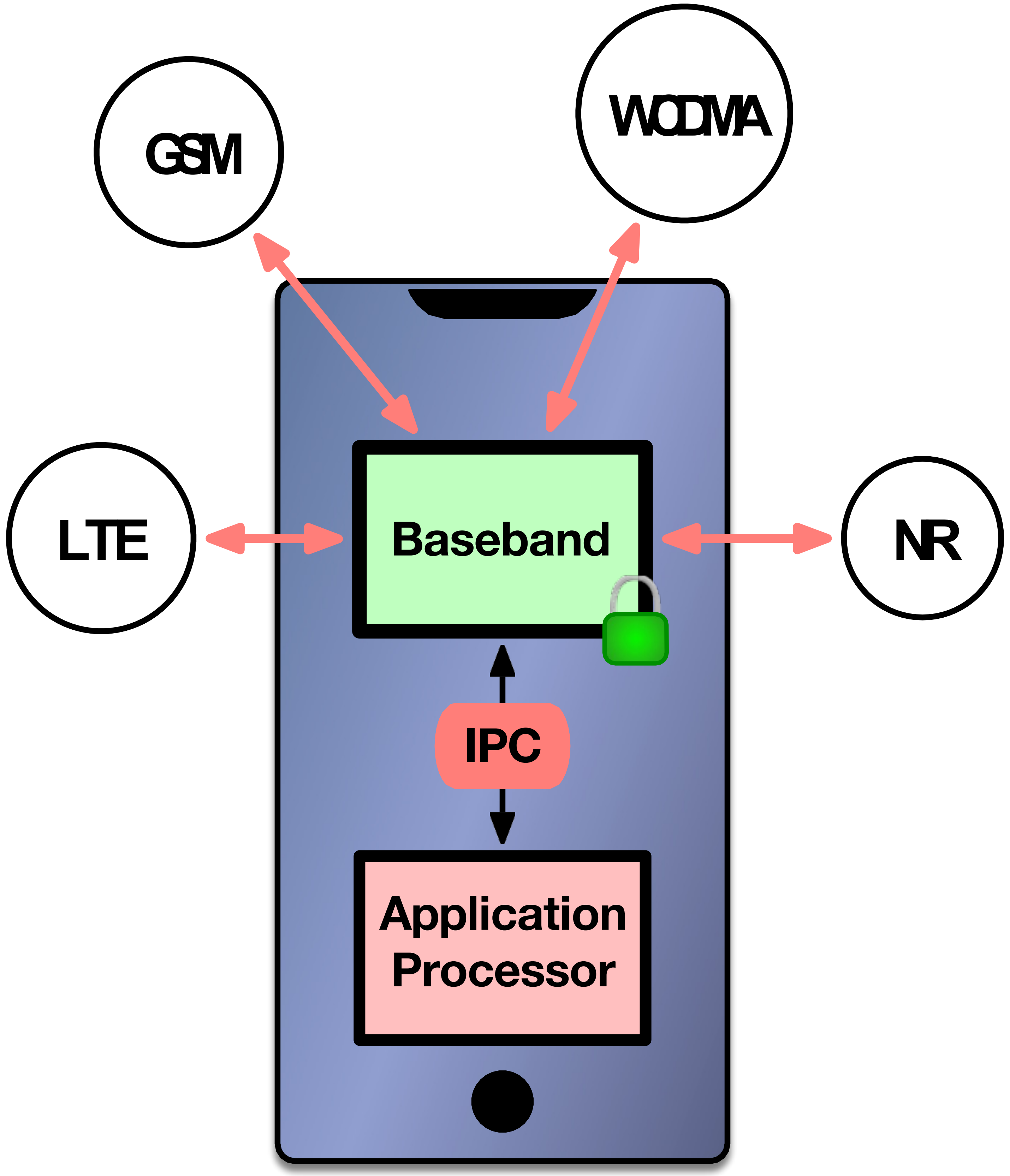


```
#2 query_mac_dac_cap(_,crash_dump,1,CAP_SYS_ADMIN,P).
```

22 additional processes beyond zygote could escalate

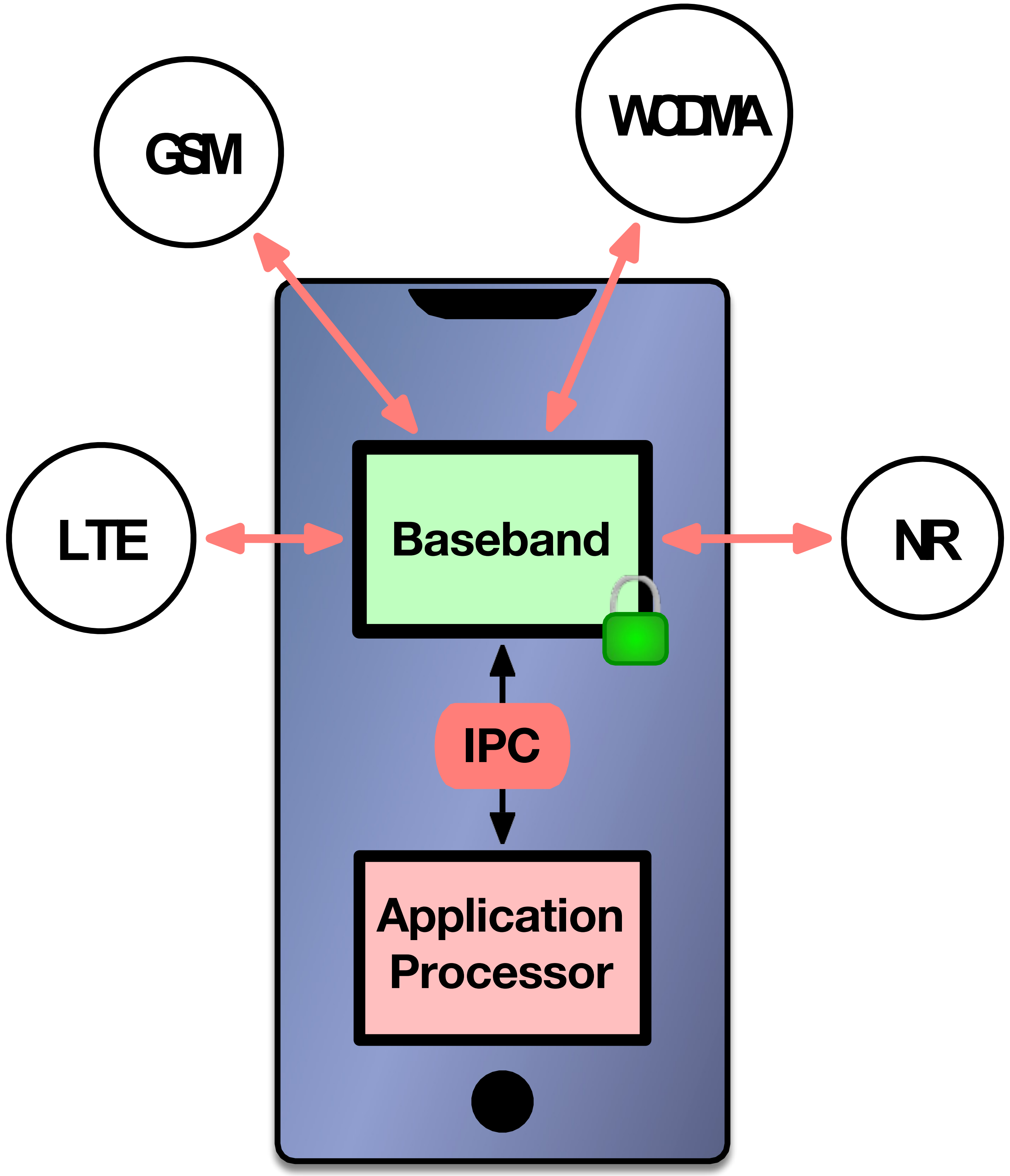
Baseband processors

- Basebands implement multiple generations of 3GPP (and, for now, 3GPP2) cellular standards



Why basebands?

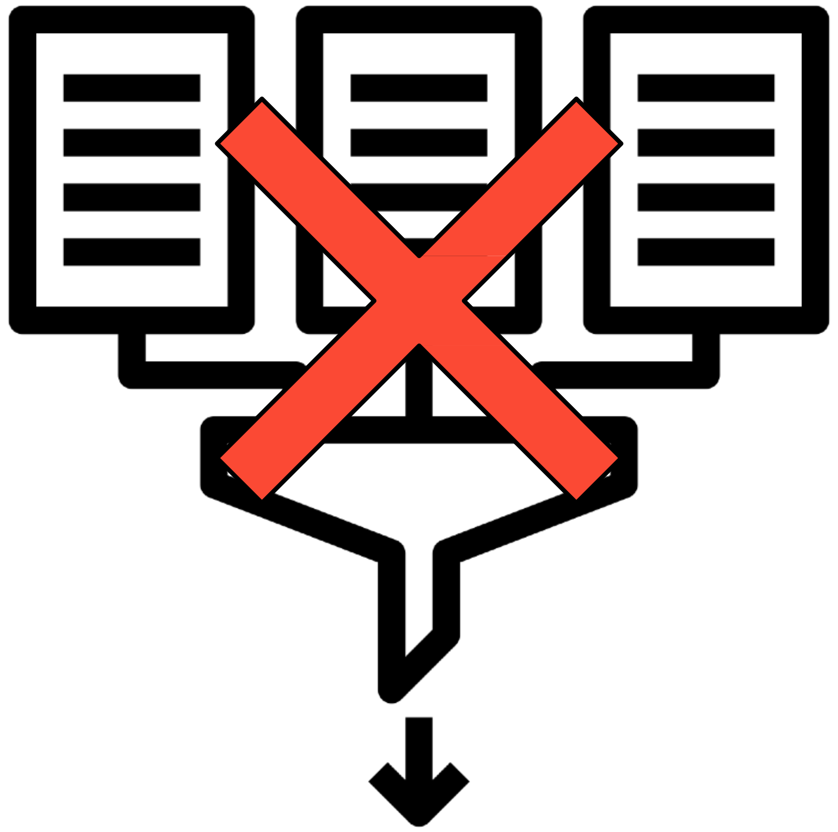
- Basebands implement multiple generations of 3GPP (and, for now, 3GPP2) cellular standards
  - More standards → more implementation bugs
  - More bugs → more security vulnerabilities
  - More vulnerabilities means more exploitable bugs
- Today, basebands are comparatively "easier" targets. Android/iOS userspace, kernel, and browsers are hard targets to exploit
  - But baseband functionality has been largely hidden



Baseband Testing Strategies

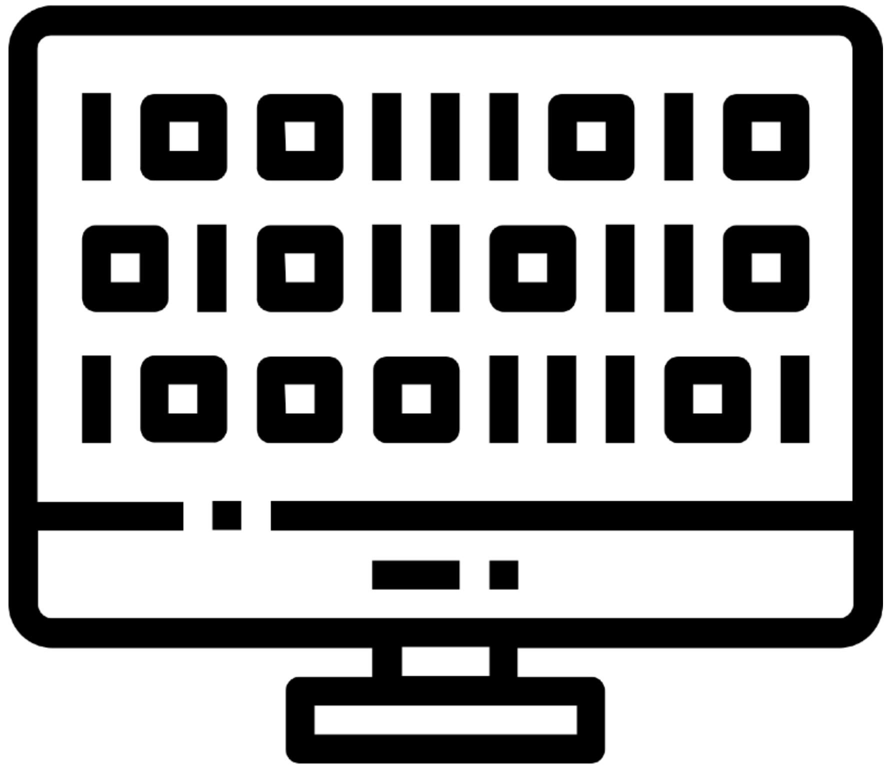
Over-the-air testing

Manual & non–deterministic
Lack of crash details



Binary Static Analysis

Many complex protocols and
firmware versions to analyze



Emulation

```
                 ___            __      __
-.     .-.     | __|(+) _ _ _ _\ \    / /(+) _ _ ___    .-.     .-
  \   /   \    | _|  | | '_| '  \ \/\/ /  | | '_/ -_)  /   \   /
   '-'     '-|_|   | |_| |_|_|_|\_/\_/   | |_| \___|-'     '-'
               ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- FirmWire is the first dynamic analysis platform to support emulating Samsung and MediaTek baseband firmware from boot
- Built on PANDA (QEMU emulator derivative) and allows for binary-only, coverage-guided fuzzing and memory inspection
- Mostly written in Python with Avatar2 device orchestrator as an underlying framework

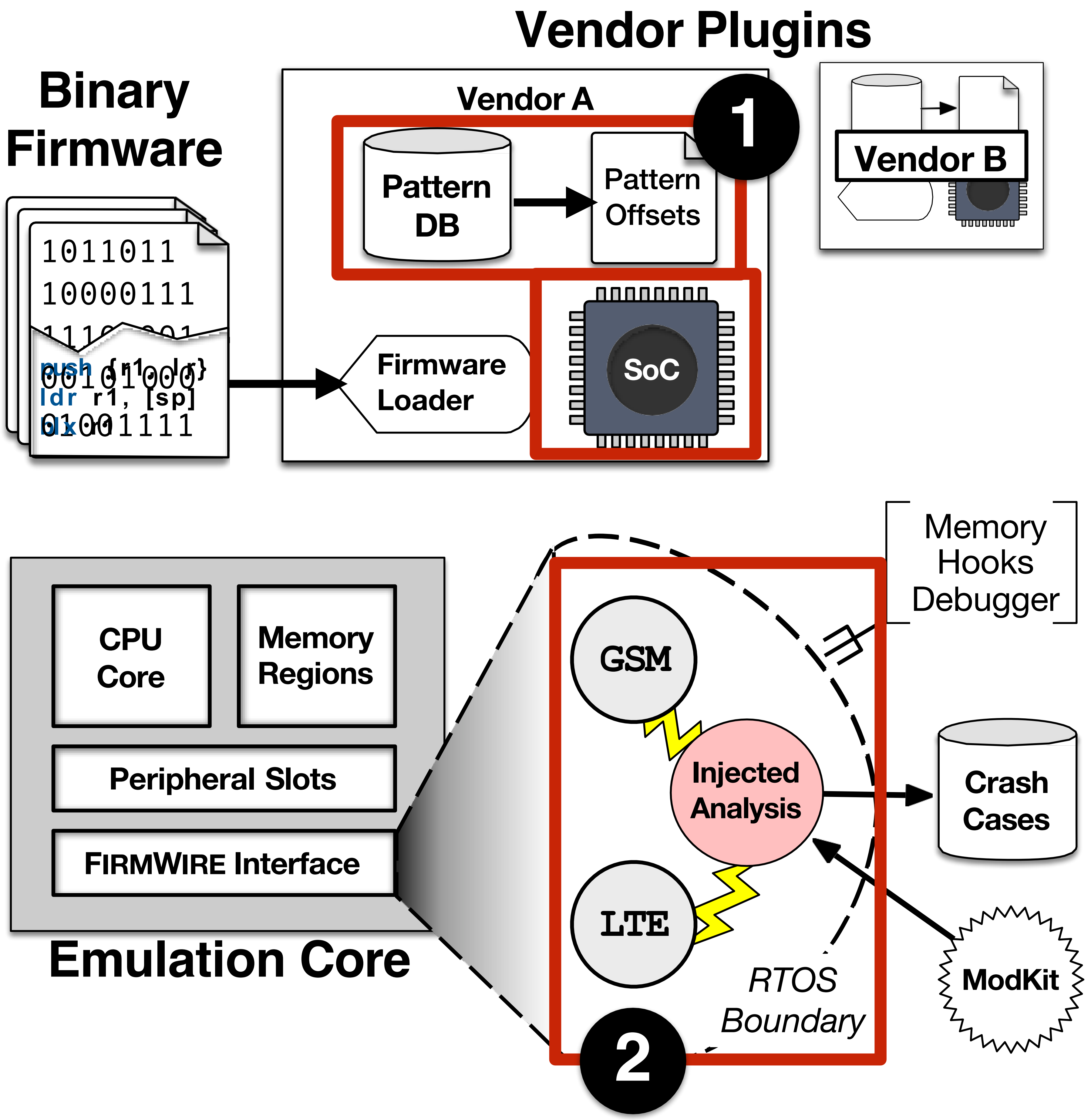
MEDIATEK

MIPS

arm

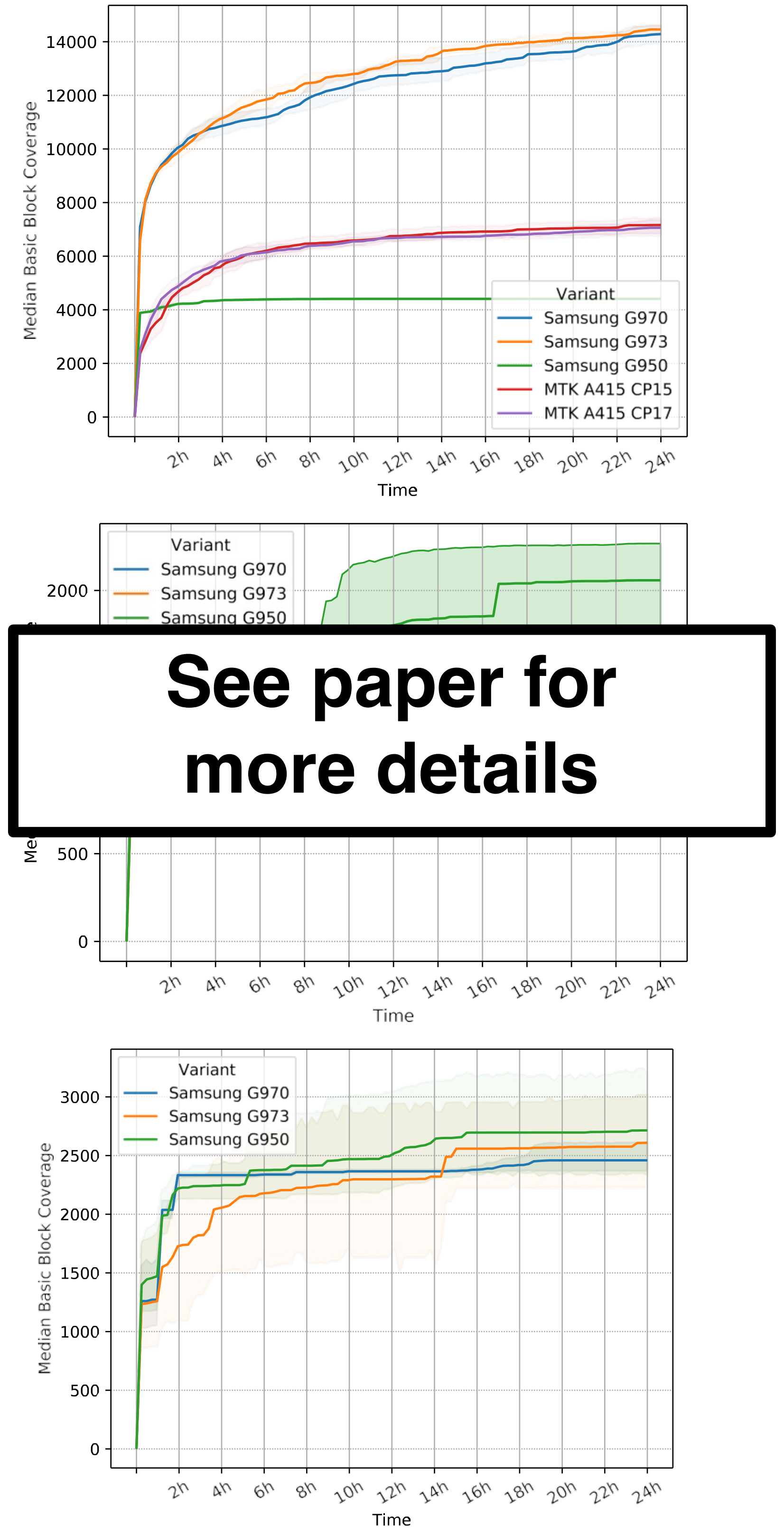
FirmWire Features

- It supports multiple platforms, chipsets, and phone models through *vendor plugins*
  - MTK: support for MIPS16e2
  - Shannon: support for ARM Cortex-R
- It offers cross-platform RTOS introspection and task injection
- We built fuzzing frameworks to assess security of GSM SM, GSM CC, and LTE RRC protocols



Fuzzing results

- **Discovered 7 crashes, 4 of which were previously unknown**
  - LTE RRC - 2 **critical**, and 1 **high**
  - GSM CC - 1 **critical**
  - GSM SM (ground-truth)
- Ratings given by Samsung
- Highest CVE - CVE-2020-25279 (9.8 critical, CC SETUP)



See paper for more details

OTA Crash Reproduction

- We replayed crashing fuzz inputs over-the-air modifying open source base stations

- <u>No SIM credentials were required, making all attacks pre-authentication</u>

LTE RRC (OpenLTE)

- Modified the RRCConnectionReconfiguration encoder to instead throw the fuzzed RRC packets

GSM (YateBTS)

- **SM** - Changed Protocol Configuration Options (PCO) encoder

- **CC** - Changed Call Setup encoder & initiated call

The basebands crashed with each message

DFS Provider Recommendations

- It is necessary, but not sufficient, to assure the security of DFS applications

- Mobile platforms are a vital part of the DFS ecosystem and also need to be assured

- Ensure that threats are enumerated against devices

- Make use of tools to assure security against attack, or ensure that manufacturer/DFS provider/integrator/regulator is using such tools

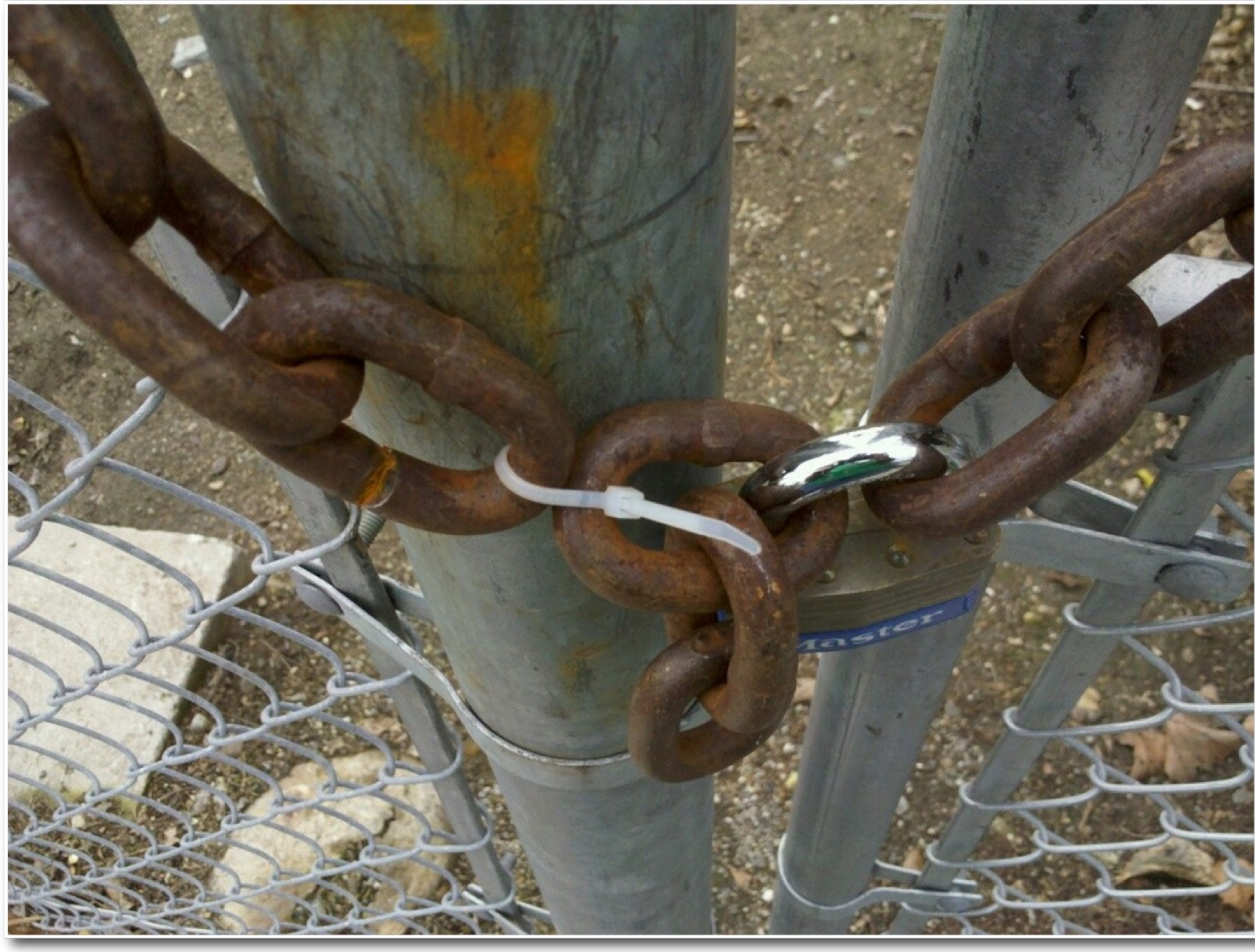
DFS Provider Recommendations

- It is necessary, but not sufficient, to assure the security of DFS applications

- Mobile platforms are a vital part of the DFS ecosystem and also need to be assured

- Ensure that threats are enumerated against devices

- Make use of tools to assure security against attack, or ensure that manufacturer/DFS provider/integrator/regulator is using such tools



DFS Provider Recommendations

- It is necessary, but not sufficient, to assure the security of DFS applications
- Mobile platforms are a vital part of the DFS ecosystem and also need to be assured
- Ensure that threats are enumerated against devices
- Make use of tools to assure security against attack, or ensure that manufacturer/DFS provider/integrator/regulator is using such tools

Related Papers

Tian, D. (Jing), Hernandez, G., Choi, J.I., Frost, V., Ruales, C., Traynor, P., Vijayakumar, H., Harrison, L., Rahmati, A., Grace, M., Butler, K.R.B., 2018. {ATtention} Spanned: Comprehensive Vulnerability Analysis of {AT} Commands Within the Android Ecosystem. Presented at the 27th USENIX Security Symposium (USENIX Security '18), pp. 273–290.

Hernandez, G., Tian, D. (Jing), Yadav, A.S., Williams, B.J., Butler, K.R.B., 2020. BigMAC: Fine-grained policy analysis of Android firmware, in: Proceedings of the 29th USENIX Security Symposium (USENIX Security'20). USENIX Association, pp. 271–287.

Elgharabawy, M., Kojusner, B., Mannan, M., Butler, K.R.B., Williams, B., Youssef, A., 2022. SAUSAGE: Security Analysis of Unix domain Socket usAGE in Android, in: Proceedings of the 7th IEEE European Symposium on Security and Privacy (EuroS&P'22), pp.572-586.

Hernandez, G., Muench, M., Maier, D., Milburn, A., Park, S., Scharnowski, T., Tucker, T., Traynor, P., Butler, K., 2022. FirmWire: Transparent Dynamic Analysis for Cellular Baseband Firmware, in: Proceedings of the 2022 Network and Distributed System Security Symposium (NDSS'22). DOI://10.14722/ndss.2022.23136

Contact

Kevin Butler, FICS Research Director: butler@ufl.edu

FICS Research: https://fics.institute.ufl.edu

github.com/FirmWire/FirmWire

Funding acknowledgemnts: National Science Foundation CNS-1815883, Office of Naval Research ONR-OTA N00014-20-1-2205, Semiconductor Research Corporation,Air Force Office of Scientific Research, Dutch Research Council NOW 628.001.0303 (“TROPICS”), Natural Science and Engineering Council of Canada

https://atcommands.org