DFS Application Best Practices for Mobile Application Developers

Dr. Kevin Butler
Director, Florida Institute for Cybersecurity Research
Professor, Computer and Information Science and Engineering
University of Florida, Gainesville, FL, USA

March 2025

Motivation

Digital technology has spurred financial access to millions

- 55% of account owners in high-income economies and 30% of account owners in developing world economies have made at least one direct payment using a mobile money account, a mobile phone, or the Internet

The digital financial services (DFS) ecosystem is uniquely vulnerable to a variety of security threats

- Interconnectedness of system entities
- Extended security boundaries due to reliance on numerous parties
- Mobile ecosystem itself is increasingly complex – devices, OSes

Question: What are the actual security threats and controls (mitigations) for stakeholders within the DFS ecosystem?

Security Framework Goals

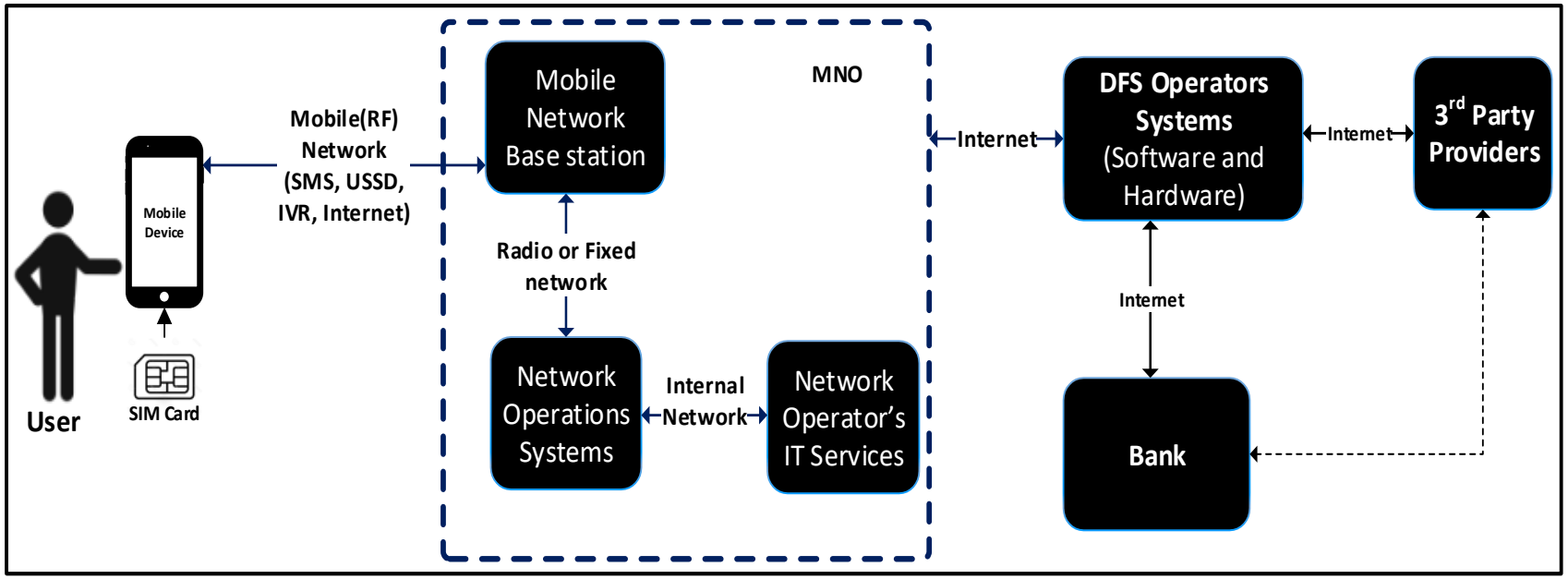
The X.1150 Security Assurance Framework aims to bridge the knowledge gap and recommends a structured methodology for risk management

How can the framework be used?

- Enhance customer trust and confidence in DFS
- Clarify roles and responsibilities for each stakeholder in the ecosystem
- Identify security threats and vulnerabilities within the ecosystem
- Establish security controls to provide end-to-end security
- Strengthen management practices with respect to security risk management in a manner that is inclusive to all shareholders

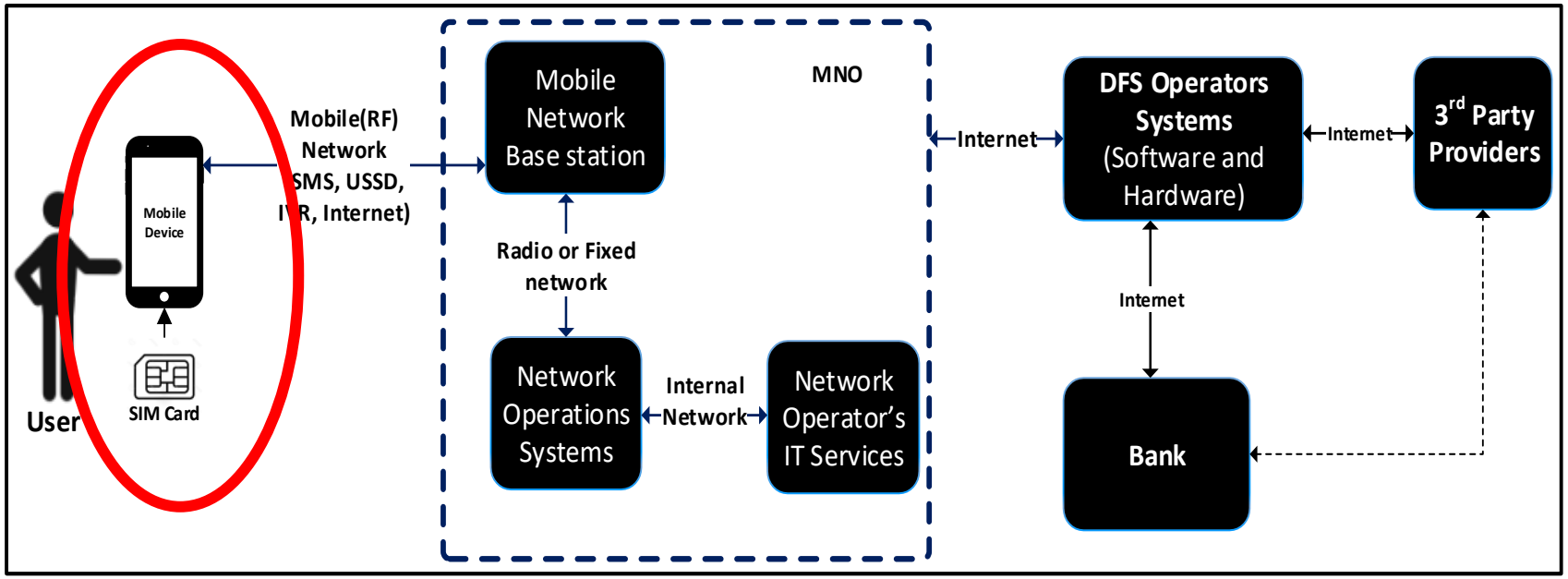
DFS Ecosystem Stakeholders

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



DFS Ecosystem Stakeholders

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



Security Assurance Framework Overview

Draws on principles from ISO/IEC 27000 security management systems standards, PCI/DSS v3.2, PA-DSS, NIST 800-53, CIS controls version 7, OWASP top-10 vulnerabilities, GSMA application security best practices

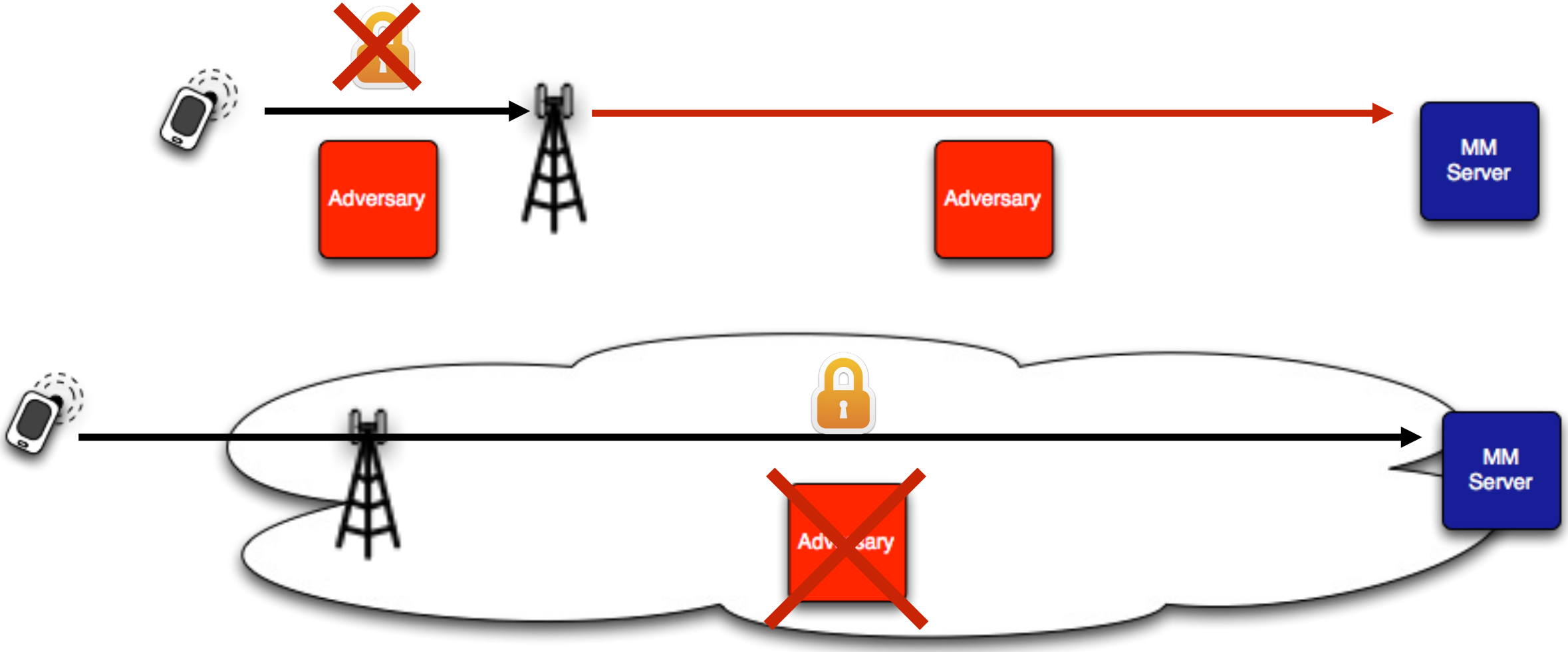
Contains the following components:

- Security risk assessment based on ISO/IEC 27005
- Assessment of threats and vulnerabilities to underlying infrastructure, DFS applications, services, network operators, third-party providers
- Discussed in terms of standardized threat then by stakeholder
- Identification of vulnerabilities enabling the threats
- Security control measures and the x.805 security dimension they represent (119 controls identified)

Security Advantages of Smartphones

Featurephones on 2G networks face vulnerabilities in a number of areas throughout the MNO's network.

Security Advantages of Smartphones



Smartphones support end-to-end security

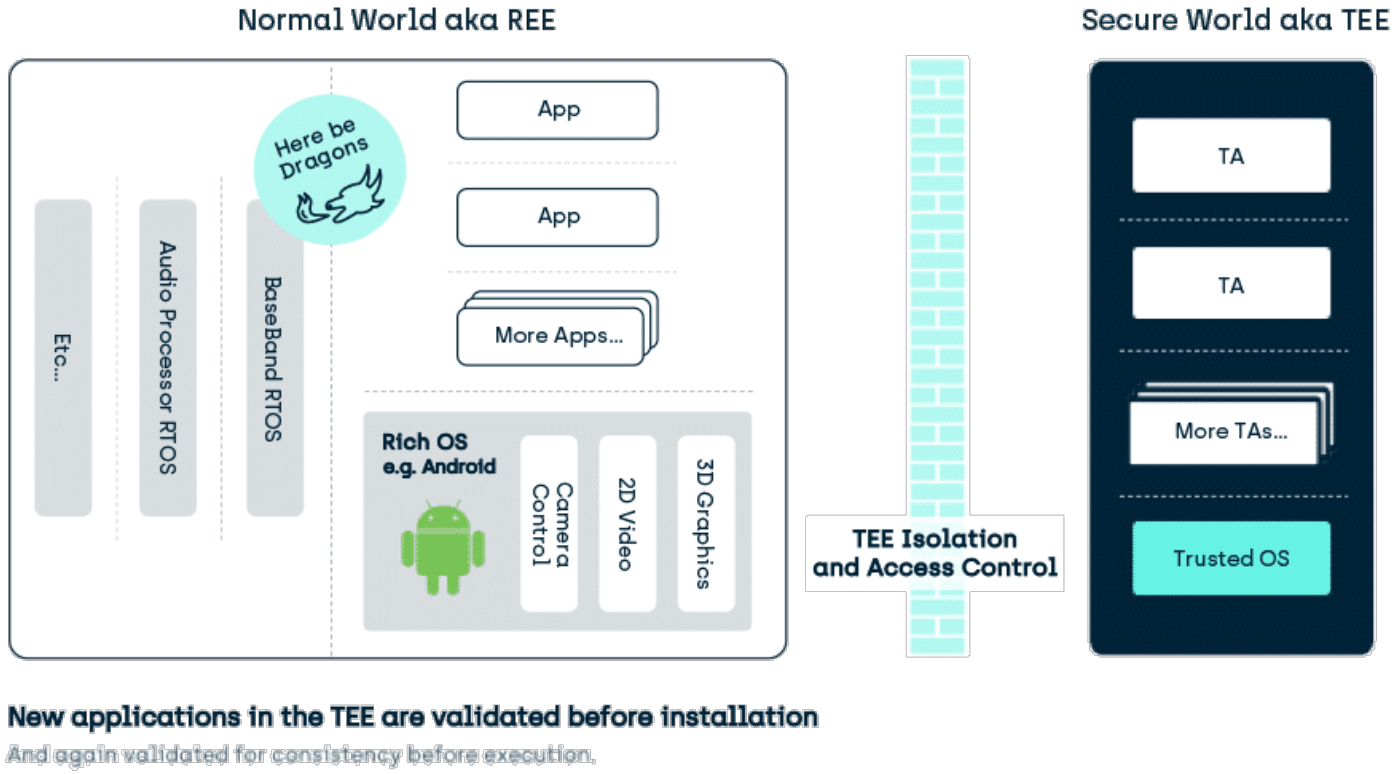
Other Smartphone Security Features

Secure hardware

- Secure boot
- Platform measurement
- Trusted execution environments

Strong authentication

- Biometric security



https://www.trustonic.com/technical-articles/what-is-a-trusted-execution-environment-tee/

Template for Application Security Best Practices

- General best practices for a mobile money smartphone application security framework
- Draws upon GSMA study on mobile money best practices, ENISA smartphone security development guidelines, State Bank of Pakistan mobile payment applications security framework
- Template can be used as input to an app security policy by DFS providers
- Considerations: device and application integrity, communication security and certificate handling, user authentication, secure data handling, secure application development

Device and Application Integrity

- Assure the integrity of mobile devices prior to engaging in sensitive data transactions
  - The safest devices are those that have not been “jailbroken” or “rooted” since it may not be possible to assess their security
  - Circumvents platform integrity routines such as secure boot
  - Applications should use mobile platform services to determine that they and the underlying platform have not been modified
- Remove extraneous code that may have been added to the application during testing or features not designed for the target platform prior to deploying app in production
- Assure the integrity of the app on the server side

Communication Security

- Ensure that apps are using standardized encryption libraries
- Current best practice is minimum TLS 1.2

  Why TLS? Widely-deployed, protocol openly available, not patent-encumbered, free libraries in every OS
- Older libraries have issues

  Susceptibility to attacks (Heartbleed, DROWN, POODLE, BEAST, etc)
- TLS v1.3 is becoming broadly available

  Better performance and security and steadily growing in adoption

TLS Ciphersuites

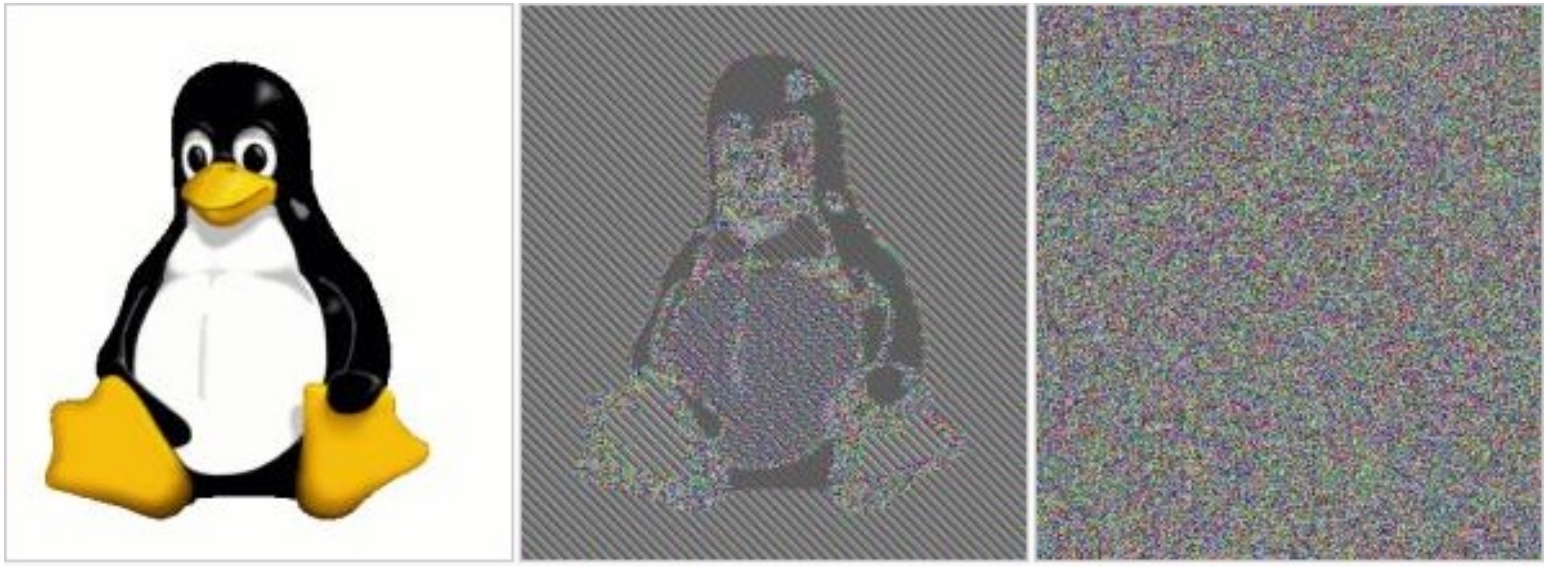
- There are many different algorithms used for authentication, encryption, and integrity
- Not all of them are considered secure for a modern deployment – especially if the server supports TLS < 1.2
- Modes of operation: the way that you encrypt information is important to consider
  - Encryption ciphers are generally “block ciphers” that break data into chunks (blocks)
  - The way that these blocks are linked together can be good or bad for security
- One recommended mode: `TLS_DHE_RSA_WITH_AES_128_CBC_SHA`

Example: ECB Electronic Code Book Cipher

- ECB is fast and parallelizable; blocks are encrypted independently of each other
- Why is this bad?

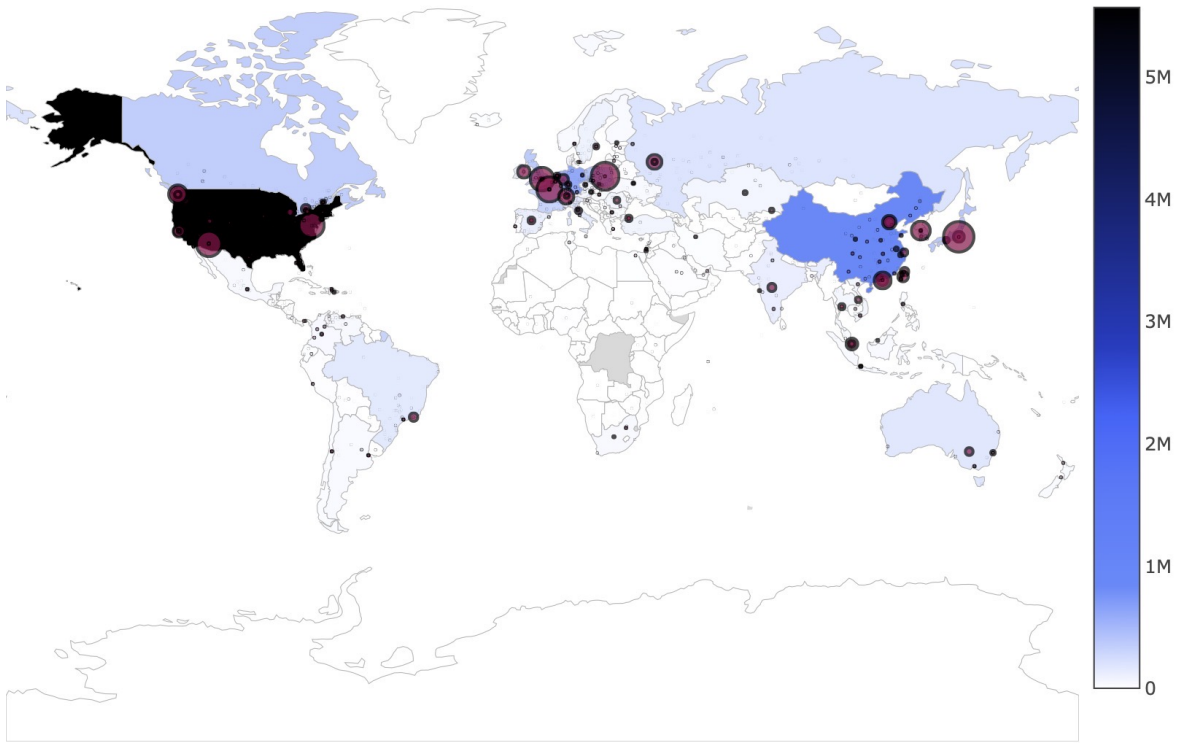
Example: ECB Electronic Code Book Cipher

- ECB is fast and parallelizable; blocks are encrypted independently of each other
- Why is this bad?

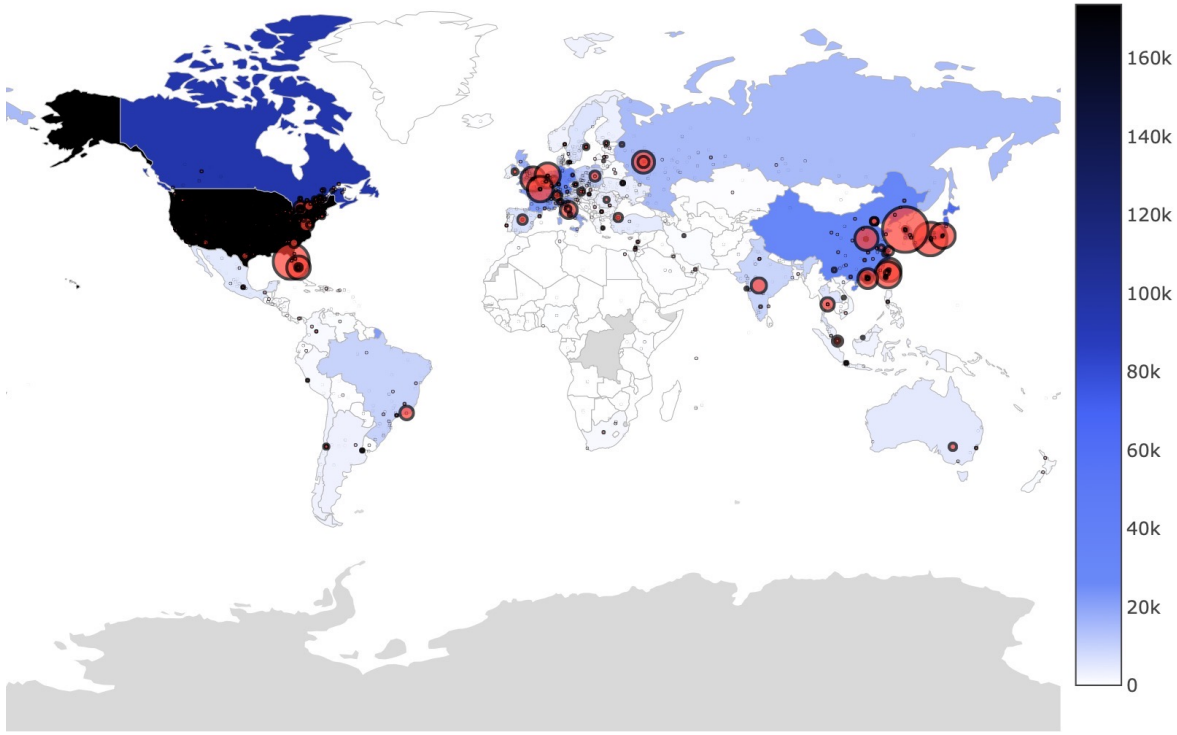


Original image | Encrypted using ECB mode | Modes other than ECB result in pseudo-randomness

Weak Cipher Existence



3DES



DES-56

- Our 2019 study found existence over 40% of queried TLS servers supported some form of DES, even DES-40 on some servers

TLS Ciphersuites

- There are many different algorithms used for authentication, encryption, and integrity
- Not all of them are considered secure for a modern deployment – especially if the server supports TLS < 1.2
- Modes of operation: the way that you encrypt information is important to consider
  - Encryption ciphers are generally “block ciphers” that break data into chunks (blocks)
  - The way that these blocks are linked together can be good or bad for security
- One recommended mode: `TLS_DHE_RSA_WITH_AES_128_CBC_SHA`

TLS on Clients

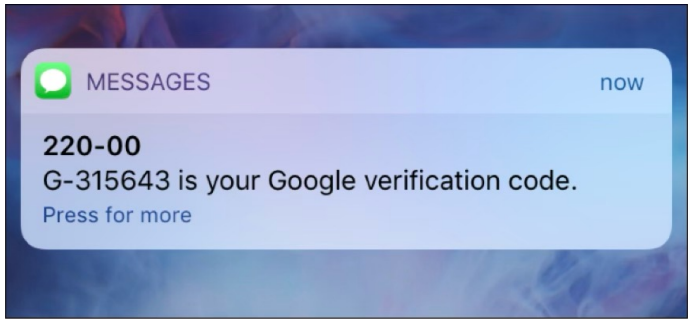
- All versions of Android beginning with API level 16+ have support for TLS 1.2
  - Corresponds to Android 4.1 Jelly Bean, released July 2012
- TLS 1.2 enabled by default for API level 20+
  - Corresponds to Android 5.0 Lollipop, released November 2014
  - SHA-256 only supported as API option for API level 20+
- Optimally at least this level of Android should be supported to ensure TLS 1.2 is employed by clients

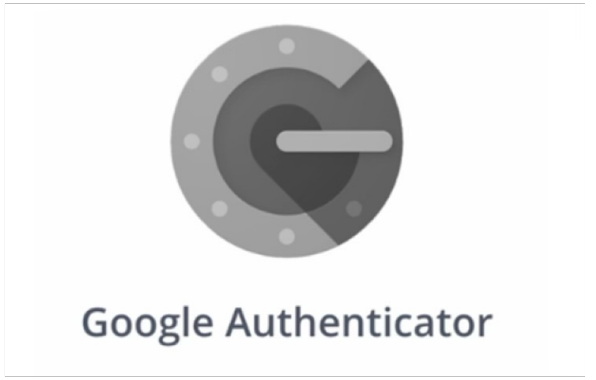
TLS Certificate Handling

- Smartphone OSes properly check to see that new TLS connections include a valid X.509 certificate
  - But many applications **override** this check
  - Likely to silence errors during testing
- Bypassing certificate validation and server authentication ***should never*** be done in production code
- Trusted CAs should be stored securely by the client
  - E.g., in an Android `KeyStore` object for storing the set of trusted Cas and a `TrustManagerFactory` object to store them in
- Certificate pinning can prevent being presented a fake server name later
  - Use APIs such as Android's `CertificatePinner`

Two-Factor Authentication

- Two-Factor Authentication (2FA) simply adds a second credential.
- 2FA comes in many forms
  - Something I know, Something I have, Something I am…
- Often a tradeoff between **security** and **usability**

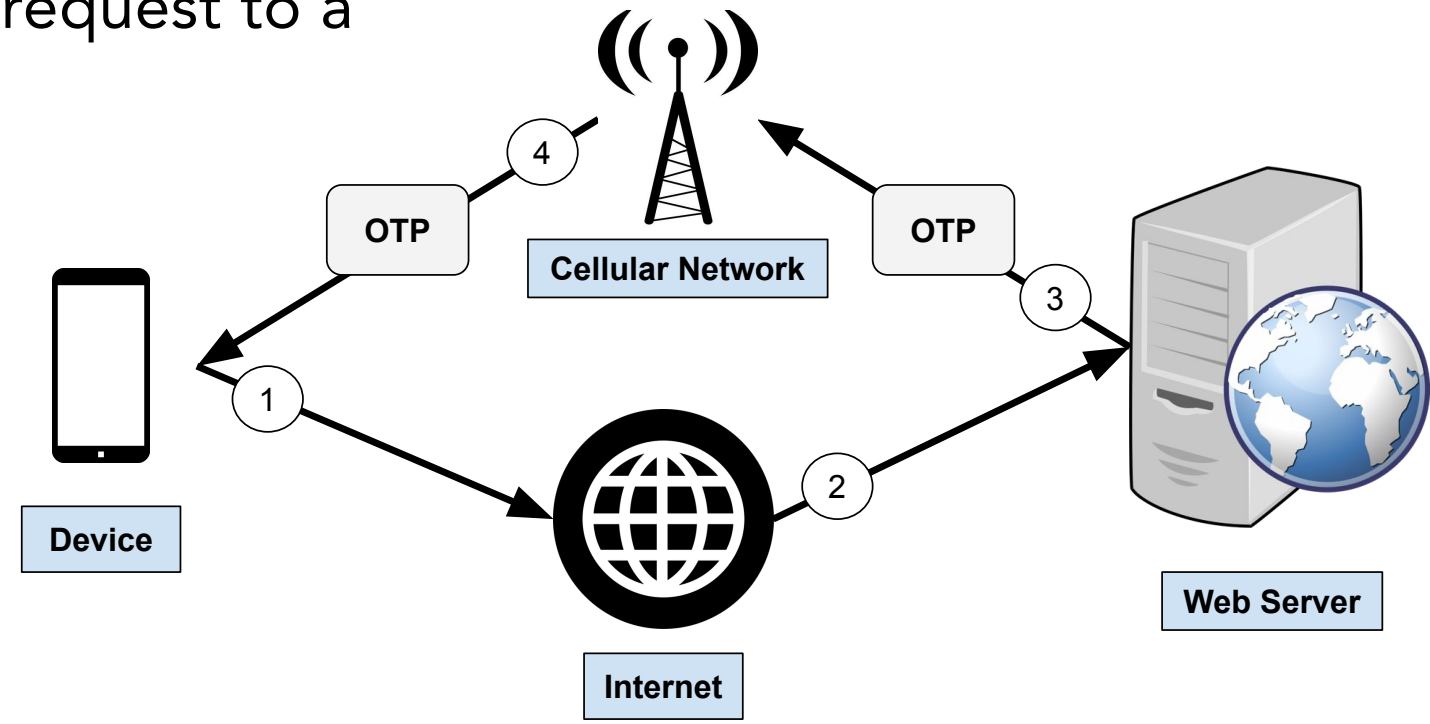




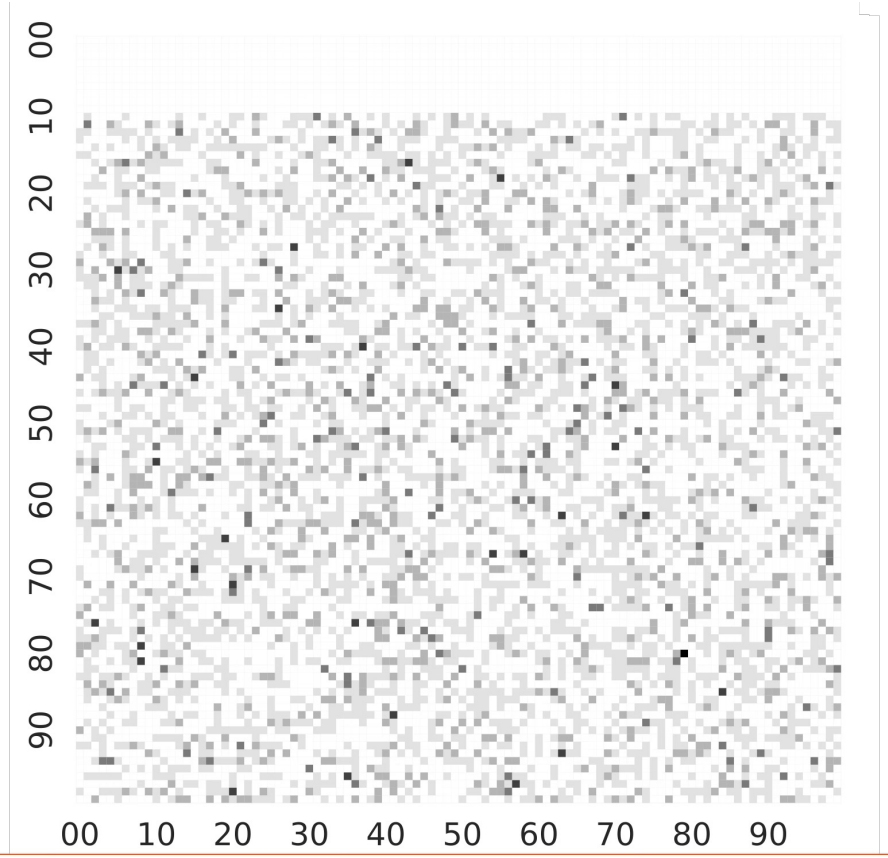
2FA Principles

- At their core, most 2FA variants are similar
- A device makes a verification request to a server via the internet
- The server responds with a One Time Password (OTP)
- Two stage process:

1. Registration
2. Authentication



OTP Selection



WeChat:
rand()∗16 mod10000

Talk2:
?

LINE:
No leading 0s

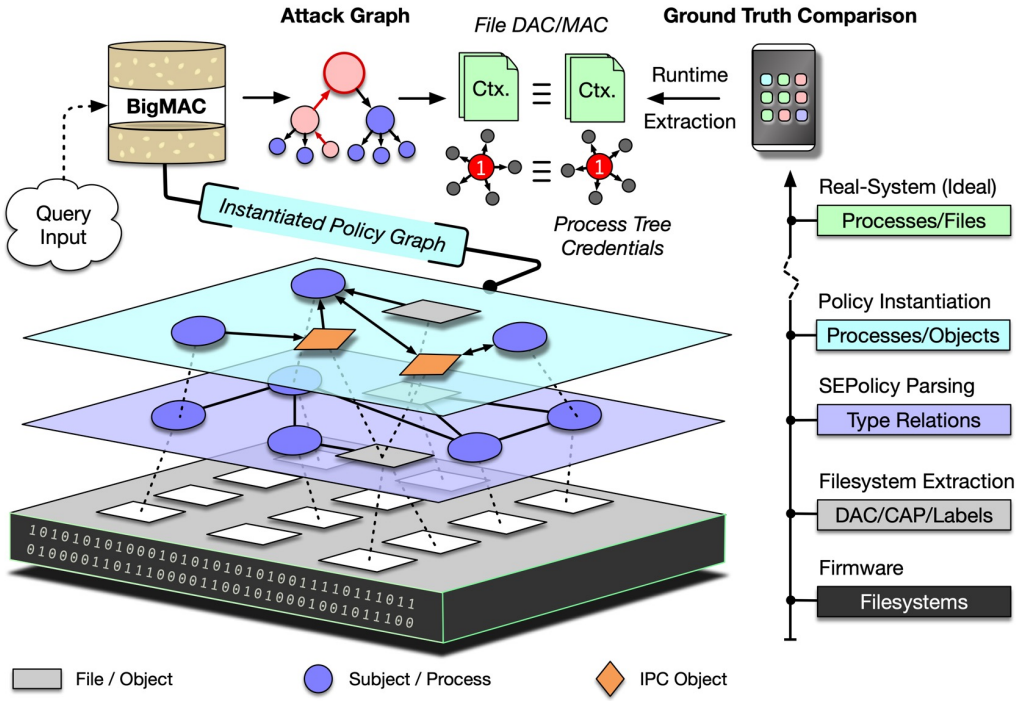
Do you know what your OTP generator is sending out?

Secure Data Handling Recommendations

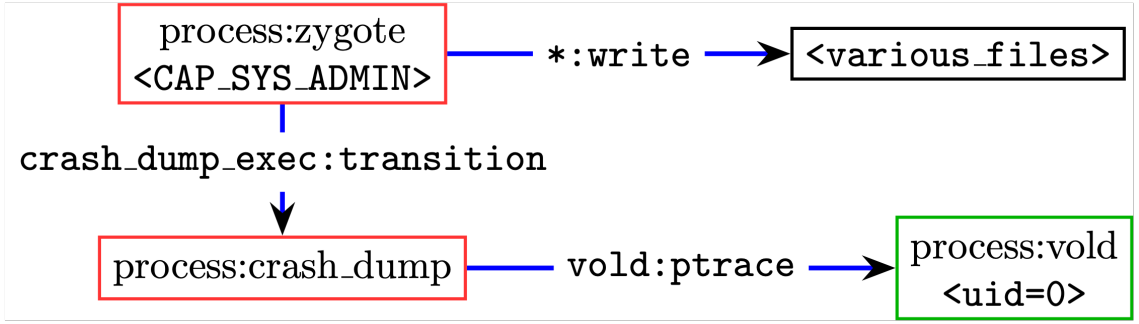
- Use the Android KeyStore framework
- Use TEEs when they are available on devices
- Delete confidential information from caches and memory and avoid general exposure of information

  E.g., storing the secret key on the stack
- Use fine-grained permissions to restrict data sharing
- Do not hard-code sensitive information such as passwords or keys into the application source code

Platform Tools for Access Control Checking




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

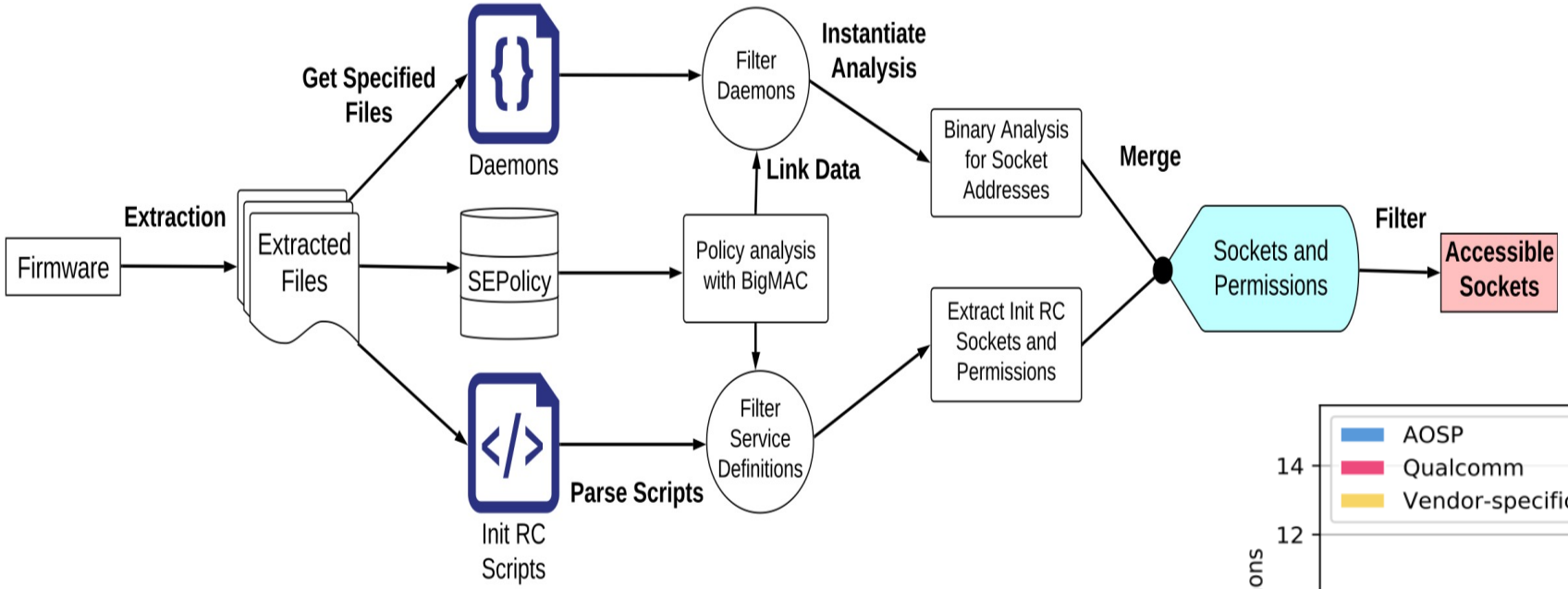



```
#2 query_mac_dac_cap(_,crash_dump,1,CAP_SYS_ADMIN,P).
```

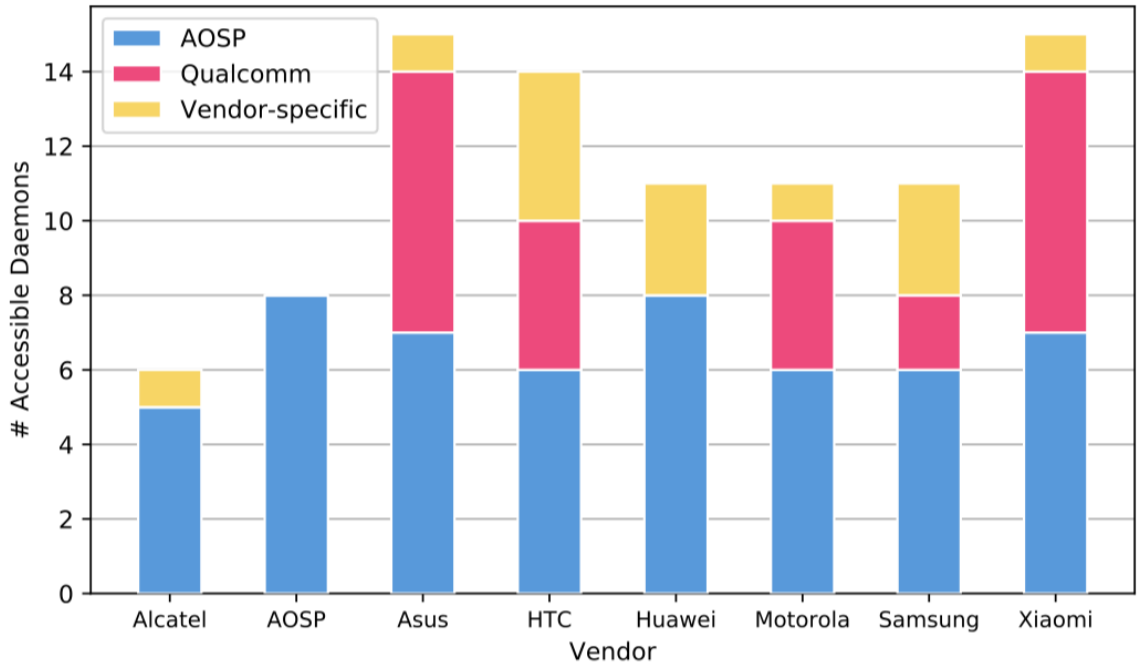
22 additional processes beyond zygote could escalate

- BigMAC tool allows permission checking of access control rules for potential privilege escalation on a variety of smartphone platforms

Platform Tools for Access Control Checking



- SAUSAGE tool extends BigMAC to allow analysis of UNIX socket IPCs and potential privilege escalations



Related Research [1]

- Reaves, B., Scaife, N., Bates, A., Traynor, P. and Butler, K.R.B. 2015. Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World. *Proceedings of the USENIX Security Symposium* (2015).
- Reaves, B., Tian, D., Scaife, N., Blue, L., Traynor, P. and Butler, K. 2016. Sending out an SMS: Characterizing the security of the SMS ecosystem with public gateways. *Proc. 2016 IEEE symposium on security and privacy,* May 2016.
- Frost, V., Tian, D. (Jing), Ruales, C., Prakash, V., Traynor, P. and Butler, K.R.B. 2019. Examining DES-based Cipher Suite Support within the TLS Ecosystem. *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security* (New York, NY, USA, Jul. 2019), 539–546
- Hernandez, G., Tian, D. (Jing), Yadav, A.S., Williams, B.J. and Butler, K.R.B. 2020. BigMAC: Fine-grained policy analysis of android firmware. *Proceedings of the 29th USENIX security symposium (USENIX security'20)* (Aug. 2020), 271–287.
- Elgharabawy, M., Kojusner, B., Mannan, M., Butler, K.R.B., Williams, B. and Youssef, A. 2022. SAUSAGE: Security Analysis of Unix domain Socket usAGE in Android. *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)* (Genoa, Italy, Jun. 2022), 572–586.

Summary

- Good practices revolve around basic security principles
- The X.1150 assurance framework and template provide a starting point for ensuring principles of authentication, access, control, integrity, and confidentiality are maintained
- Technologies change so specific recommendations might also change but the long-term trends are clear
- Be vigilant!

- For more information:
- Email: butler@ufl.edu
- https://kevinbutler.org
- https://fics.institute.ufl.edu