An Investigation of the Android Kernel Patch Ecosystem

Zheng Zhang¹, Hang Zhang¹, Zhiyun Qian¹, Billy Lau²
¹UC Riverside, ²Google Inc
Openness of Linux and Android

Linux distros
- Ubuntu
- Fedora
- CentOS

Linux mainline

IoT & CPS
- Routers
- Camera
- Cars

Android
- Android AOSP
- Qualcomm Android
- Samsung
- Xiaomi

1500 bugs reported in Linux kernel 2018*

* Syzbot and the Tale of Thousand Kernel Bugs - Dmitry Vyukov, Google, 2018
Android kernel ecosystem

Goal:
How long and why?

Android Zero-Day Panic as Ancient Linux Flaw Forgotten

by Richi Jennings on October 7, 2019
Measurements: Goals

• The relationship between the upstream and downstream
e.g., initial patch? how does it propagate?

• The timeliness of patch propagation
  e.g., typical delay in each step? bottleneck?

• The factors that influence the patch propagation
  e.g., current best practices? how to improve?
Challenges

• Challenge 1: Not completely open source
  - OEM provides mostly binary kernel images
    - **Fiber**: H. Zhang and Z. Qian. Precise and accurate patch presence test for binaries. USENIX Security, 2018
  - Good accuracy with some improvements (~95%)

• Challenge 2: Decision making a blackbox
  - Complex ecosystem/Poor documentation
  - Information from multiple sources
  - Measurements/Inference
Measurements: overall

Dataset:
- 20 phone models, 500+ instances
- 400 Linux/Qualcomm CVEs:
  Android/Pixel security bulletin
  From the beginning to 5/2019

Bottleneck (Linux CVEs):
- Qualcomm mainline
- => Qualcomm stable/OEMs

(30, 0.4): Delay is less than 30 days for 40% of CVEs
Knowledge of security patches

• Lack of knowledge
  – Patches == fixing serious security flaws?

• Knowledge generation
  – Time-consuming triage to determine security impact

• Knowledge propagation
  – Not efficient
Delay in each layer

- Linux mainline
- Linux LTS
- Android common
- Qualcomm mainline
- Qualcomm stable
- OEM phones

- Fork
- Cherry-pick
- Merge
- Notification

Linux
Google
ASB
OEM
Qual
Delay in each layer

- LTS cherry-pick from Linux mainline
- No distinguish between security patch and bug fix
Delay in each layer

Linux mainline

- Fork

Linux LTS

- Merge
  - Notification

Android common

Qualcomm mainline

Qualcomm stable

OEM phones

- Cherry-pick

- Frequently merges
- No requirements for knowledge

Linux

Google

OEM

Qual

ASB
Delay in each layer

- Google’s key role in notification
- With much delay
Delay in each layer

- Linux mainline
  - Linux LTS
    - Android common
      - Qualcomm mainline
      - Qualcomm stable
        - OEM
        - OEM phones
  - Linux
    - Google
      - ASB
      - Qual

- One or two months before ASB release
Delay in each layer

- **Linux mainline**
  - **Linux LTS**
  - **Android common**
  - **Qualcomm mainline**
  - **Qualcomm stable**
  - **OEM phones**

- **Fork**
- **Cherry-pick**
- **Merge**
- **Notification**

- Released in ASB
- Have not patched in OEMs
Delay in each layer

- **Linux mainline**
- **Linux LTS**
- **Android common**
- **Qualcomm mainline**
- **Qualcomm stable**
- **OEM phones**

- **Fork**
- **Cherry-pick**
- **Merge**
- **Notification**

- **Patched in OEMs**
Improvements

• Automated patch triage (security or not)
  – Extracting security knowledge
  – Useful for all kernels

• More efficient knowledge propagation
  – Linux: develop its own notification mechanisms
  – OEM: Pay attention to Qualcomm notifications

• Redesign the mechanism: more merges/forks
  – Follow Linux LTS directly in downstreams (Ongoing effort @ Google)
  – still require more efforts