Simple Spyware
Androids Invisible Foreground Services and How to (Ab)use Them

December 2019
Who am I

→ **ZHAW**: Research Assistant in Information Security @ Zurich University of Applied Sciences

→ **Student**: Master of Science in Engineering

→ **Contact**: suth@zhaw.ch or via Twitter @Me7e0r232
Let’s start with the latest privacy changes
LIVE DEMO
But that’s nothing new!

So what’s new?
First step
How to run stuff in Background?

Schedulers
Alarm Manager

Job Scheduler

Started by an app, but lives outside the app lifecycle.
public void scheduleJob(){
    long interval = 1000 * 60L;
    ComponentName serviceComponent = new ComponentName(this, JobScheduler.class);
    JobInfo.Builder builder = new JobInfo.Builder(JOB_ID, serviceComponent);
    → builder.setPeriodic(interval) // Minimum is 15 minutes
    → builder.setOverrideDeadline(interval * 2);  // Sets the maximum scheduling latency
    → builder.setMinimumLatency(interval);         // Run after delay
    JobScheduler jobScheduler = this.getSystemService(JobScheduler.class);
    jobScheduler.schedule(builder.build());       // Schedule the job
}
JobInfo.builder

→ setPersisted(true);
→ setRequiredNetwork(NetworkRequest networkRequest)
→ setRequiredNetworkType(int networkType)
→ setRequiresBatteryNotLow(boolean batteryNotLow)
→ setRequiresCharging(boolean requiresCharging)
Second step
How to access the data?
Foreground Service
Forground Services

→ Needs to show a sticky **notification**

→ **Notification design** is set by the app

→ Can be started from **background** job

→ Do **not** have sensor limitations

→ Has to be started within a **5 seconds**
How to get rid of the notification?

We just don’t…
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
  // ~4.9999.. seconds to call startForeground(...)
  Notification notification = createCustomNotification();
  this.startForeground(1, notification) // Sensor access not restricted anymore.
  accessCamera();
  accessMicrophone();
  // ... some malicious code
  stopForeground(true); //Stop the service before notification is loaded
  return START_STICKY;
}
Long Running Tasks

→ MediaPlayer API
→ Apps do not run recording in their own lifecycle context.
Does this work on Android 10 (Q)?
New permission level:
“Allow only while using the app”
Let’s track from “background*”

*sorry, I meant foreground*
Simple Spyware

Spy Camera
Spy Microphone
Spy Location

Select Data:

Select Persistence:
- Use Job Scheduler
- Use Alarm Manager

Select interval (120 seconds recommended)

Job Scheduler Status: Stopped

START SPY
STOP SPY
Conclusion

It’s a bug… no, it’s a feature!
Limitations

- Timing
- Access
- Visibility
- Permissions
Mitigation

→ CVE-2019-2219 – Patch is coming soon

→ Probably hard to patch, since as you have seen it’s a kind of a design problem as well.

→ Security by visibility might not be a good idea?

→ Some vendors have permission minitors
No Silver Bullet

- Anti-Virus
- Permission
- Monitoring
- Revocation

Secure Design & Transparency

- Traffic Analysis
- Honey Pots
- Fuzzers
- ML
- Execution
- Graphs
- Statistics & Heuristics
Takeaways

- Stop Apps
- Revoke Access
Demo

→ **Test:** Test yourself! Don’t worry, it’s safe :-D

→ **Code:** [https://github.com/7thomasSutter/SimpleSpyware](https://github.com/7thomasSutter/SimpleSpyware)

→ **APK:** [https://github.com/7thomasSutter/SimpleSpyware/releases](https://github.com/7thomasSutter/SimpleSpyware/releases)
I hope you enjoyed this short talk...

Thank you!

there is some more info in the appendix and the whitepaper
Contact: suth@zhaw.ch or via Twitter @Me7e0r232