An SELinux-based Intent Manager for Android

Simone Mutti, Enrico Bacis, Stefano Paraboschi

DIGIP — Università degli Studi di Bergamo, Italy
{simone.mutti, enrico.bacis, parabosc}@unibg.it

Goal

We propose *SEIntentFirewall*, an SELinux intent manager that provides fine-grained access control over Intent objects, permitting to cover within MAC policies the use of intents.

Scenario

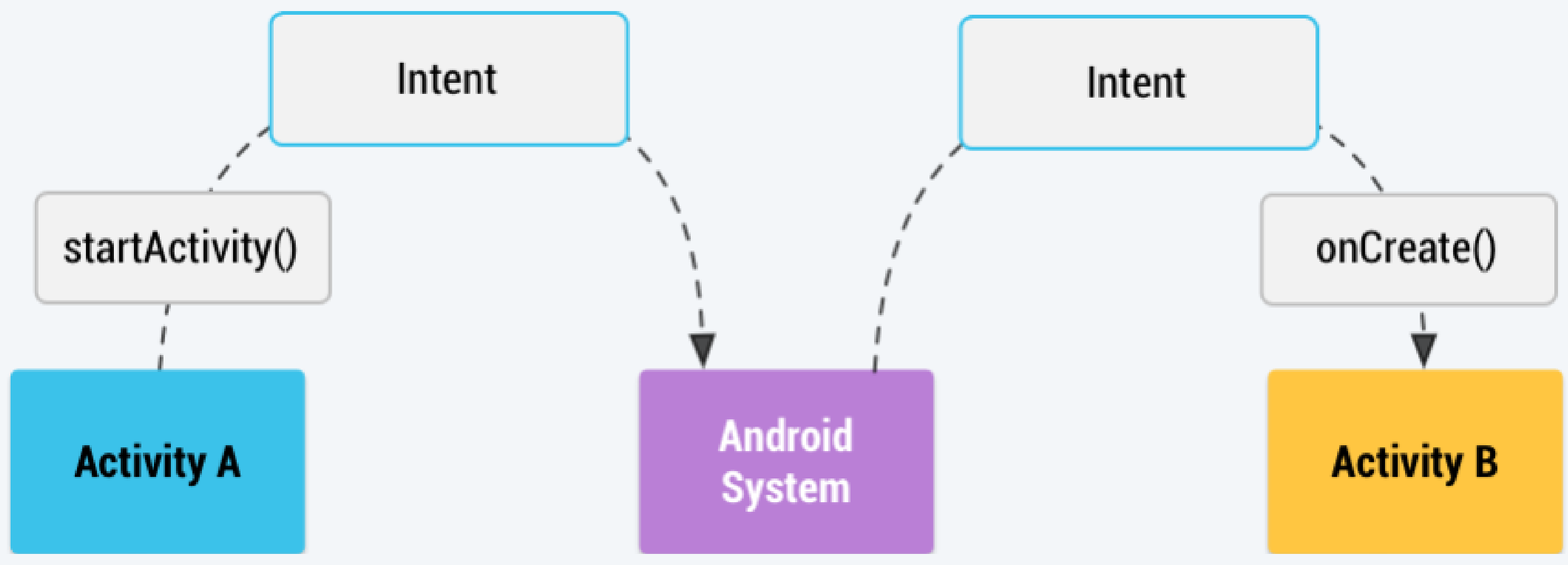

Figure 1: Abstract representation of Intent mechanism

Android provides two types of Intent:

- **Implicit intent**: it specifies the action that should be performed and optionally data that is provided for the action;
- **Explicit intent**: it explicitly defines the component that should be called by the Android system.

Problem

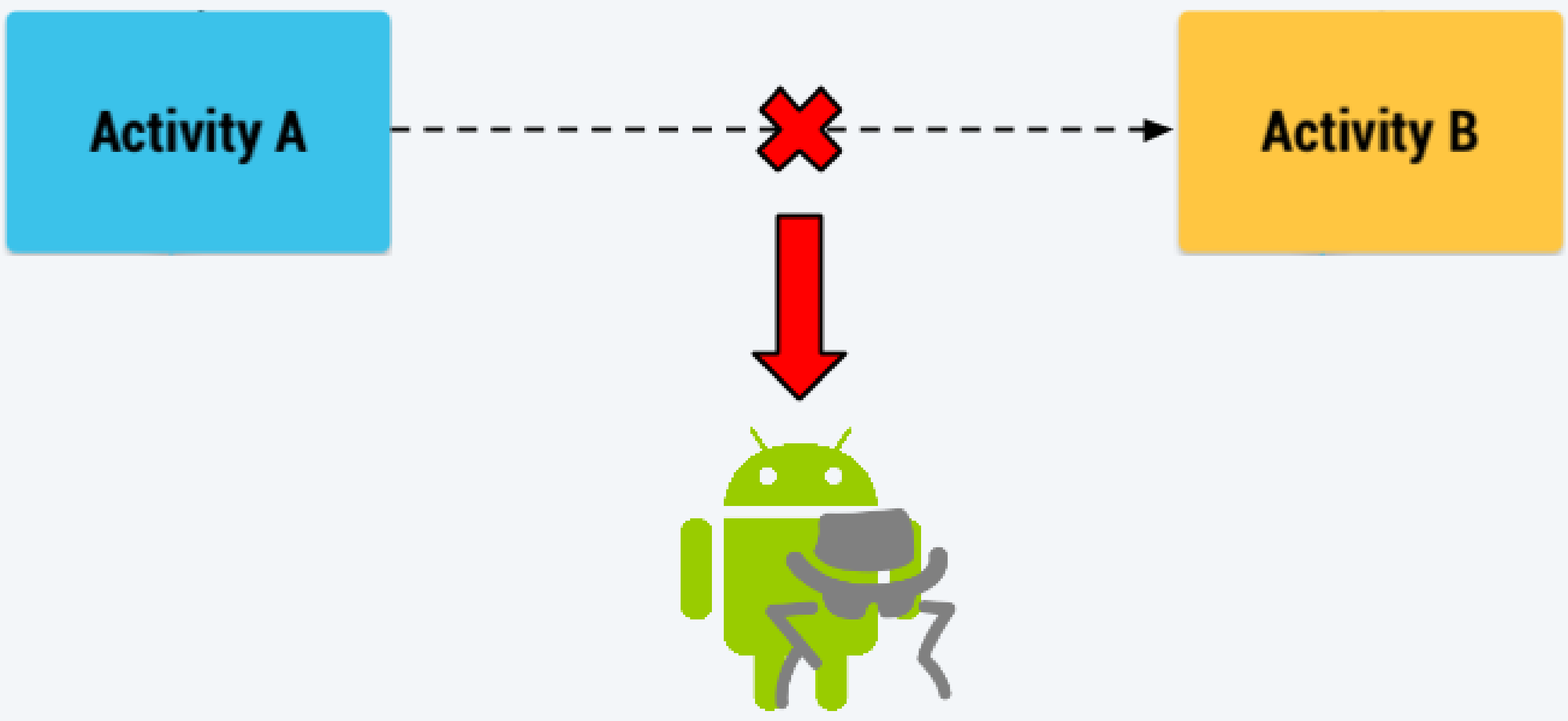

Figure 2: Abstract representation of an hijacking attack

The exchange of intents represents an application attack surface[1]:

- **Activity hijacking attack**: a malicious Activity is launched in place of the intended Activity;
- **Service Hijacking attack**: a malicious Service intercepts an Intent meant for a legitimate Service;
- **Intent spoofing attack**: a malicious application sends an Intent to an exported component that is not expecting Intents from that application.

Current solution

To address this problem, Google has introduced the *Intent Firewall* component, since Android 4.3. The *Intent Firewall* is a security mechanism that regulates the exchange of *Intents* among apps, by analyzing the type of data exchanged.

Limitations of the current solution

- Only the root user can modify the *Intent Firewall* policy;
- The introduction of a new policy language and its own Policy Decision Point (PDP) increase the **Policy Fragmentation** problem.

SEIntentFirewall

SEIntentFirewall is a built-in enhancement of *IntentFirewall*, providing fine-grained Mandatory Access Control (MAC) for Intent objects.

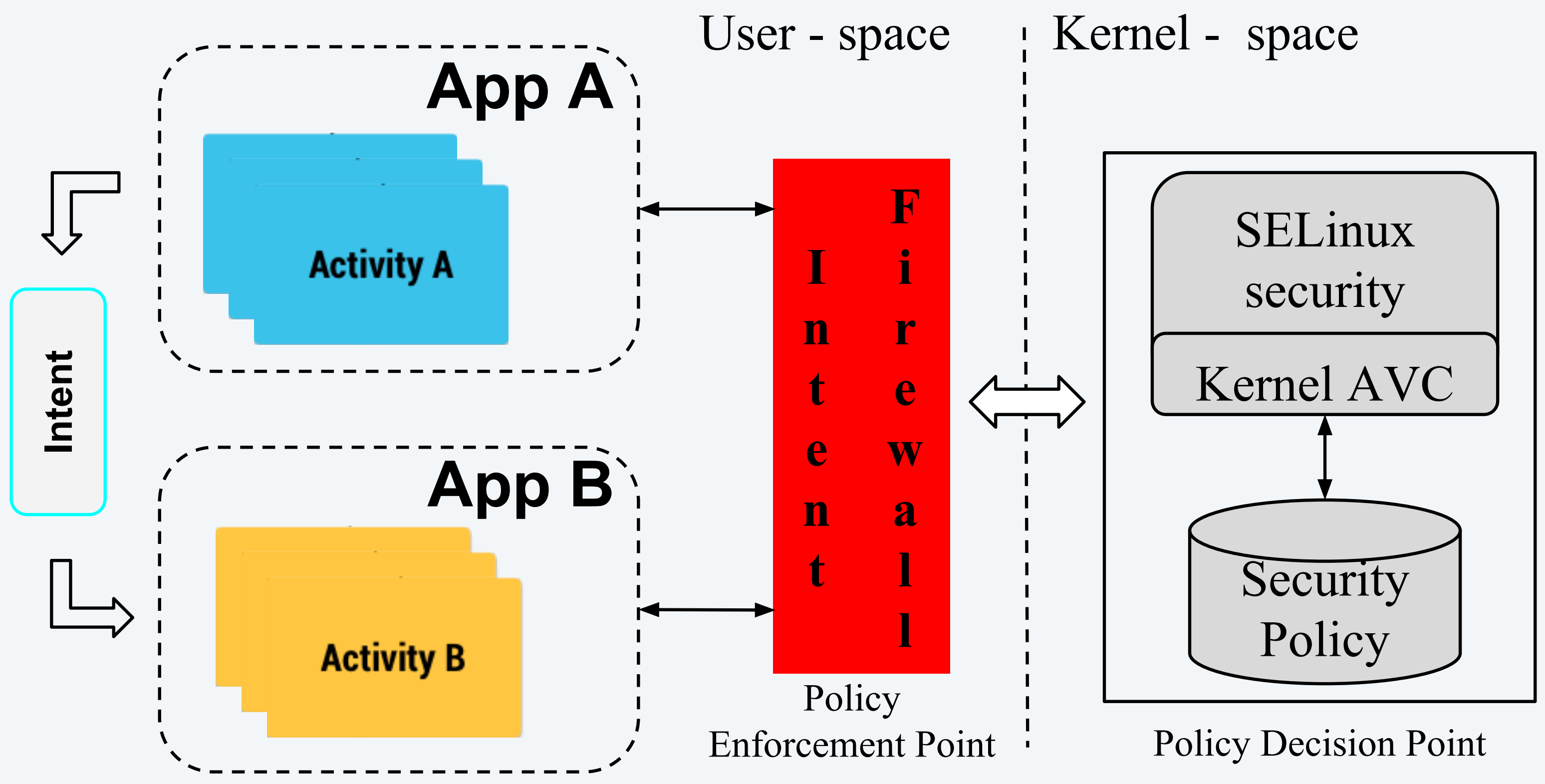

Figure 3: Overview of the *SeIntentFirewall* architecture

- *SEIntentFirewall* takes access control decisions based on a SELinux security policy;
- The SELinux decision engine will then operate as the Policy Decision Point;
- No need to modify apps source code. The *SEIntentFirewall* will be obtained with an adaptation of the services provided by *AppPolicyModules* [2].

Conclusion

- The integration of SELinux into Android is a significant step toward the realization of more robust and flexible security services;
- The potential of an SELinux-based solution like *SEIntentFirewall* leads to a significant improvement in access control enforcement and app isolation.

References

[1] Erika Chin, Adrienne Porter Felt, Kate Greenwood, and David Wagner.
Analyzing inter-application communication in Android.
In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 239–252. ACM, 2011.

[2] Enrico Bacis, Simone Mutti, and Stefano Paraboschi.
AppPolicyModules: Mandatory Access Control for Third-Party Apps.
In *Proc. of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 309–320. ACM, 2015.

[3] Stephen Smalley and Robert Craig.
Security Enhanced (SE) Android: Bringing Flexible MAC to Android.
In *Network and Distributed System Security Symposium (NDSS 13)*, 2013.

This work was partially supported by a Google Research Award (winter 2014), by the Italian Ministry of Research within the PRIN project “GenData 2020” and by the EC within the 7FP and H2020 program, respectively, under projects PoSecCo (257129) and EscudoCloud (644579).