Diagnosing intrusions in Android operating system using system flow graph

Radoniaina Andriatsimandefitra
Valérie Viet Triem Tong
Ludovic Mé

EPC SUPELEC/INRIA CIDRE, Rennes, France

Workshop Interdisciplinaire sur la Sécurité Globale

Supélec

Introduction

Android

- 500 million Android devices activated in the third quarter of 2012
- Google Play : $700,000$ available applications
- Target of malicious applications
- Google solution : analyse applications published on Google Play (no real host-based solution)

Our approach

- Monitor how pieces information from a third-application flow within the system
- Build a system flow graph based on observed flows to diagnose the attacks

Supélec

Information flow

Definitions

Information flow : information transfer from one entity to an other one

To monitor information flow : to survey all information transfer between entities of the monitored environment

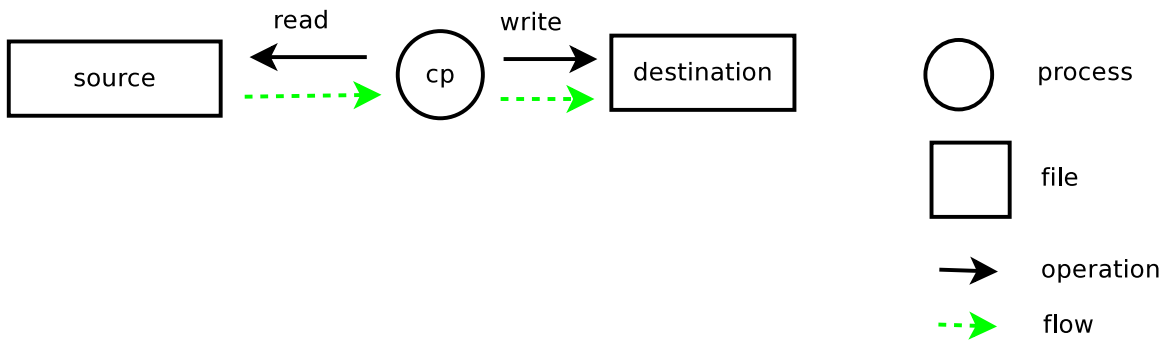

FIGURE: Example of information flow at system level

How to track pieces of information

- Taint each object based on their content feature
- Information flow $\Rightarrow$ change the tag value of modified objects

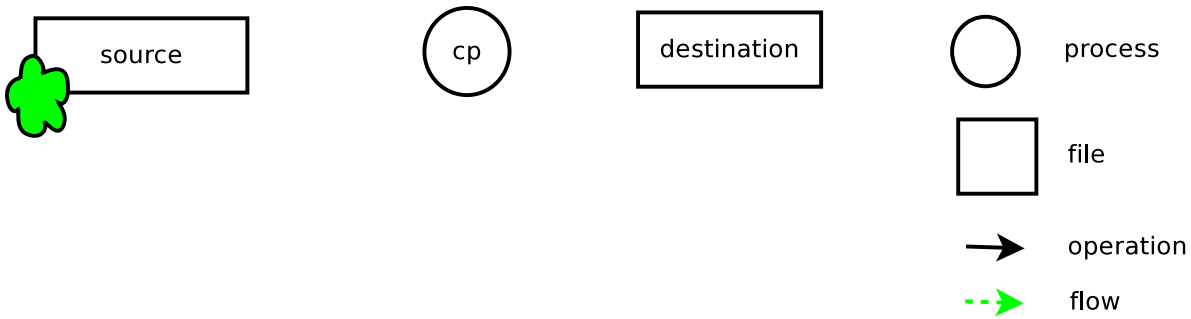

FIGURE: Example of information flow monitoring

How to track pieces of information

- Taint each object based on their content feature
- Information flow $\Rightarrow$ change the tag value of modified objects

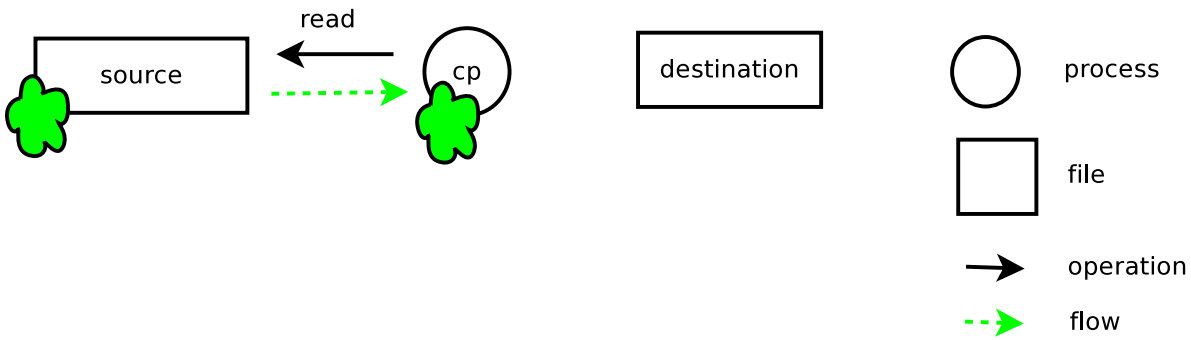

FIGURE: Example of information flow monitoring

How to track pieces of information

- Taint each object based on their content feature
- Information flow $\Rightarrow$ change the tag value of modified objects

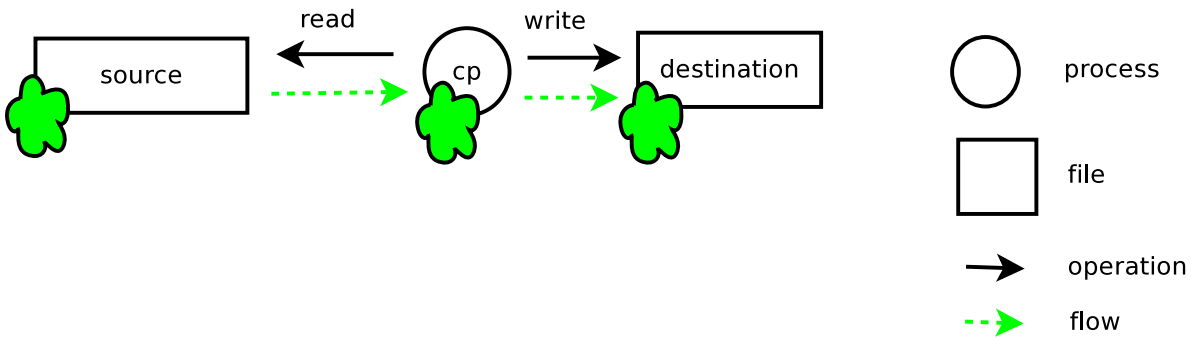

FIGURE: Example of information flow monitoring

Blare [1]

- Intrusion detection system for Linux environments
- Monitors information flow between system objects (process, file, socket etc) thanks to tainting

```
[TIMESTAMP] SRC_TYPE SRC_NAME SRC_ID > DEST_TYPE DEST_NAME DEST_ID > {i1...in}
```

General format

```
[10000] FILE SOURCE 18 > PROCESS CP :CP 147 > {1}
```

Example

FIGURE: Blare log record

1. http://blare-ids.org

System flow graph for diagnosing

Oriented graph $G = (V, E)$

- Describes how pieces of information flow between system objects
- Each $v \in V$ corresponds to a system object.
  3 attributes : a type, a name and a system identifier
- Each $e \in E$ corresponds to a unique information flow
  2 attributes : pieces of information involved and timestamps

`[10000] FILE SOURCE 18 > PROCESS CP :CP 147 > {1}`

$$G = (V, E),\ V = \{v_1 = (file, source, 18), v_2 = (process, cp, 147)\},$$
$$E = \{(v1, v2, \{1\}, \{1000\})\}$$

Supélec

Case study : DroidKungFu sample

- Detected on 05/31/2011
- Published as a SIP-client in Chinese-alternatives of Google Play
- Embeds root exploits
- Embeds an Android application meant to be installed after gaining root access
- Detection rate on *VirusTotal* : 32/46

Supélec

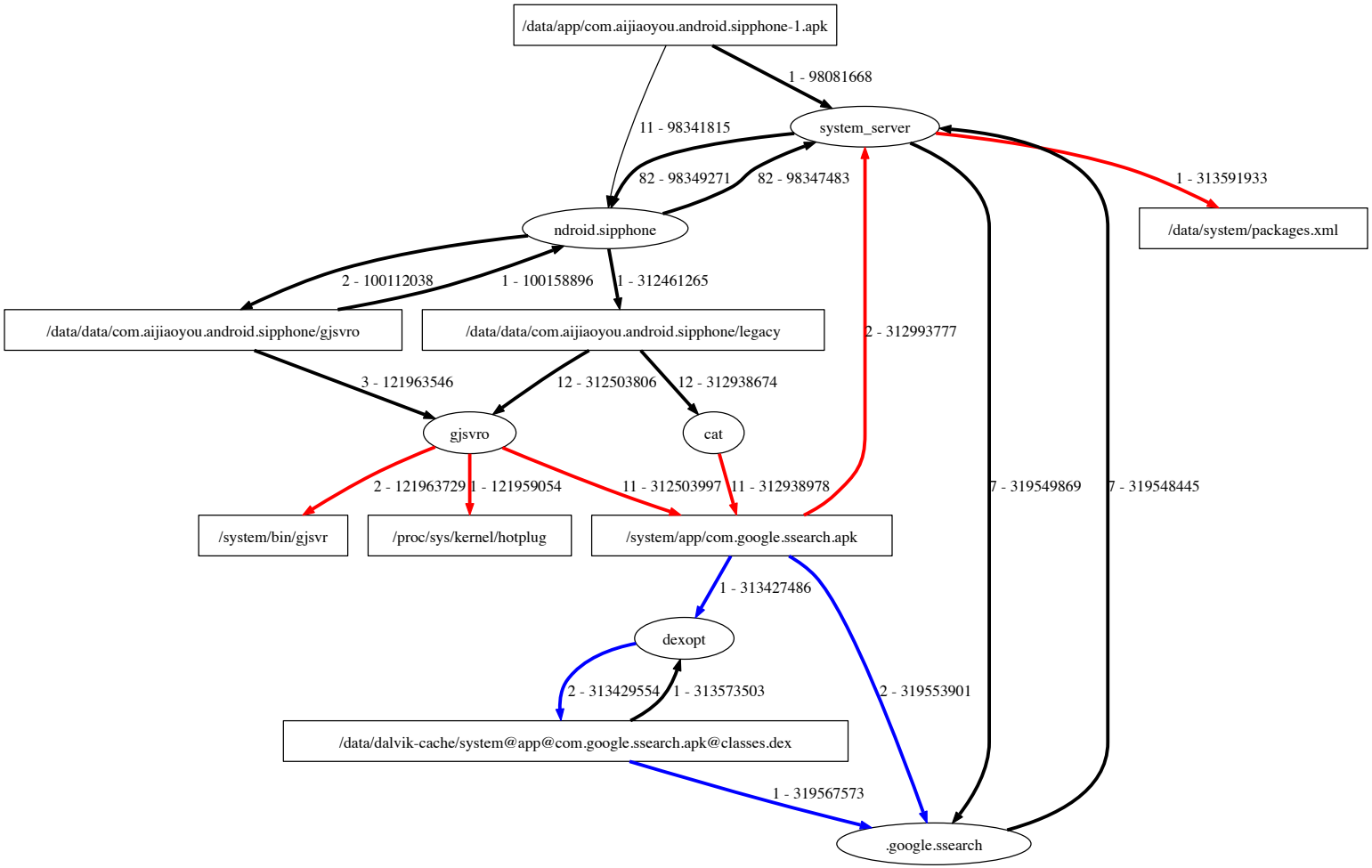

FIGURE: System flow graph of DroidKungFu

Conclusion

To sum up

- We proposed a structure named system flow graph to diagnose attacks / analyse applications
- We showed its usefulness with an analysis of a sample of DroidKungFu

What's next ?

- Use with an IDS where diagnosis is built only after policy violation
- Build a flow policy of a benign application based on its corresponding system flow graph

Supélec