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
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 of information from a third-application flow within the system
- Build a system flow graph based on observed flows to diagnose the attacks
Information flow

Definitions

**Information flow**: information transfer from one entity to another 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 ⇒ 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

```plaintext
[TIMESTAMP] SRC_TYPE SRC_NAME SRC_ID > DEST_TYPE DEST_NAME DEST_ID > \{i_1...i_n\}
```

General format

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

Example

**Figure:** Blare log record

---

1. [http://blare-ids.org](http://blare-ids.org)

R. Andriatsimandefitra (CIDRE)  Information flow, Android  WISG, January 22\(^{nd}\) 2013
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

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

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