Ovaldroid: an OVAL-based Vulnerability Assessment Framework for Android

Martín Barrère, Gaëtan Hurel, Rémi Badonnel and Olivier Festor

Madynes Research Team - INRIA Nancy Grand Est - LORIA, France.

Email: {barrere, hurel, badonnel, festor}@inria.fr

1. Introduction

- Overwhelming development of mobile technologies and services [1].
- Large-scale deployment of Android-based devices [2].
- However, security issues require special attention.
  - x Hostile environments involving vulnerabilities and malware.
  - x Lack of standard and mature means for exchanging Android related security information.
  - x Limited resources on mobile devices.

The Big Question

How do we increase the vulnerability awareness of Android-based devices?

⇒ We propose a lightweight OVAL-based framework to efficiently increase vulnerability detection capabilities in the Android platform. ✓

2. The OVAL Language

- Open Vulnerability and Assessment Language [3].
- Information security standard for assessing and reporting the machine state of computer systems.
  - Representing system configuration information for testing.
  - Analyzing the system for the presence of the specified machine state.
  - Reporting the results of the assessment.

3. The Model

- $P = \{p_1, p_2, \ldots, p_n\}$. Set of atomic properties to observe on target systems.
- $V = \{v_1, v_2, \ldots, v_m\}$. Set of vulnerabilities composed of a logical combination of properties p_i.

Vulnerability pattern matrix (n=5, m=3) example:

$$\left.\begin{array}{l} v_1 = (p_1, p_3, p_5) \\ v_2 = (p_2, p_4) \\ v_3 = (p_1, p_2, p_5) \end{array}\right\} \quad PM_{3,5} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

System state properties encoded as:

$$s = (s_1, s_2, \ldots, s_n) \quad s_i \in \{0, 1\}$$

We define the $hflatten$ operator as follows:

$$hflatten(PM) = (\sum_{j=1}^{n} a_{1j}, \sum_{j=1}^{n} a_{2j}, \ldots, \sum_{j=1}^{n} a_{mj})^T$$

⇒ **Vulnerability assessment process:**

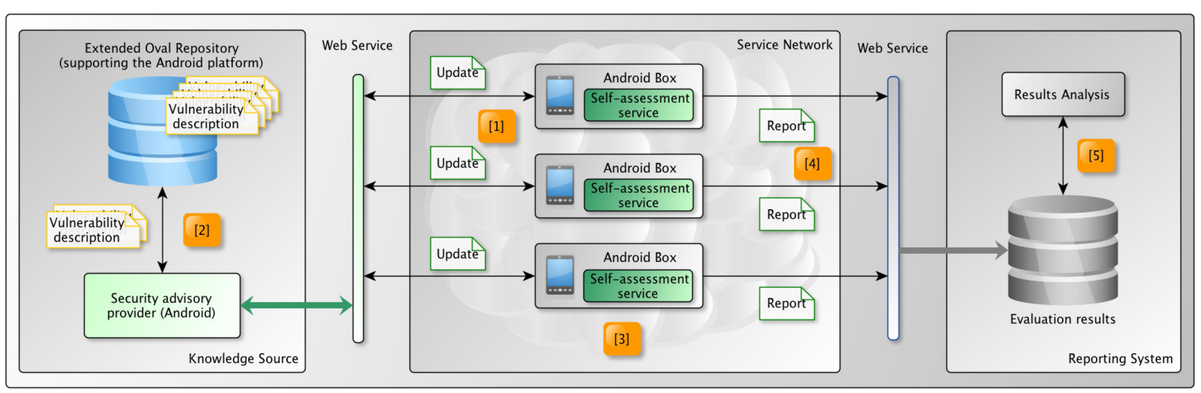
$$w = hflatten(PM) - [PM * s^T]$$

$$\Downarrow$$

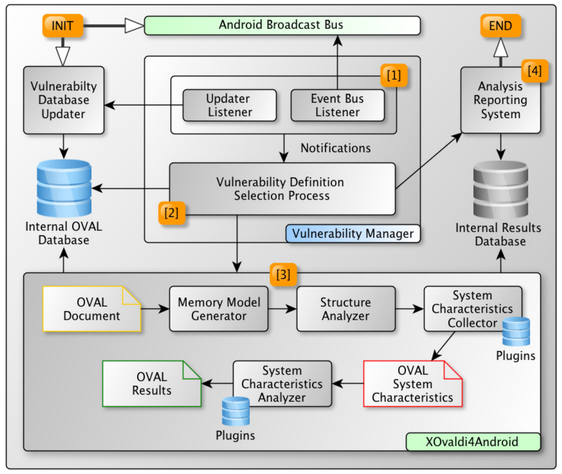
$$w = \begin{pmatrix} \sum_{j=1}^{n} a_{1j} \\ \sum_{j=1}^{n} a_{2j} \\ \vdots \\ \sum_{j=1}^{n} a_{mj} \end{pmatrix} - \left[\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \times \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} \right]$$

Vector $w = (w_1, w_2, \ldots, w_m)$ denotes the status of each vulnerability v_i in the target system. A null entry w_i indicates that vulnerability v_i is present in the system while a non null value denotes its absence.

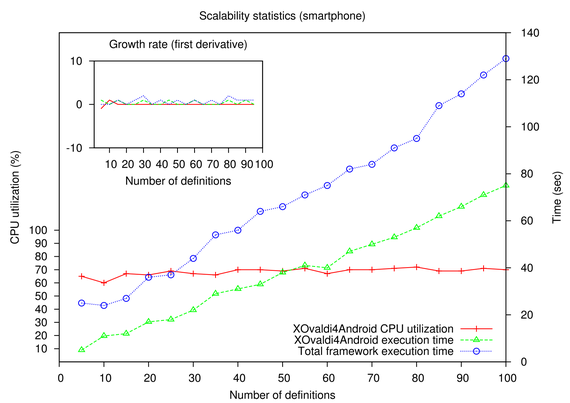
4. The Architecture

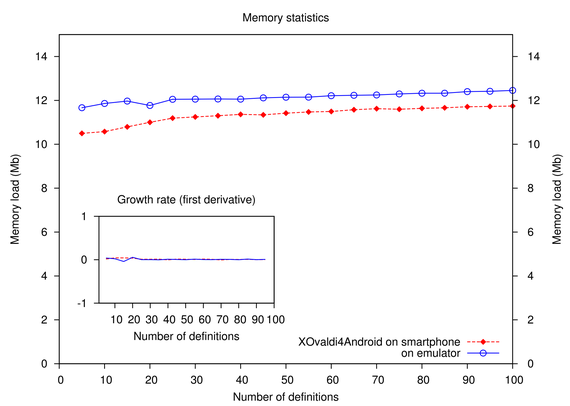


5. Self-assessment Internals



6. Scalability Statistics





The Important Fact

Real autonomy can only be achieved if we are able to ensure safe mobile configurations.

7. Conclusions and Outlook

- Mathematical model for supporting the vulnerability assessment process.
- Lightweight distributed OVAL-based framework for efficiently detecting vulnerable mobile configurations.
- Comprehensive set of experiments that show the feasibility of the proposed approach.
- Future work.
  - Protection mechanisms within the assessment framework.
  - Security advisory exchange between neighboring devices.
  - Integration of distributed vulnerabilities descriptions to detect mobile massive attack scenarios [4].
  - Closure of the vulnerability management lifecycle by performing corrective tasks thus getting closer to real autonomic solutions [5].

References

[1] M. Barrère, G. Hurel, R. Badonnel, and O. Festor. Increasing Android Security using a Lightweight OVAL-based Vulnerability Assessment Framework. In Proceedings of the 5th Symposium on Configuration Analytics and Automation (SafeConfig'12), October 2012.

[2] Android. `http://www.android.com/`. Cited May 2013.

[3] The OVAL Language. `http://oval.mitre.org/`. Cited May 2013.

[4] M. Barrère, R. Badonnel, and O. Festor. Towards the Assessment of Distributed Vulnerabilities in Autonomic Networks and Systems. In Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'12), April 2012.

[5] N. Ziring and S. D. Quinn. Specification for the Extensible Configuration Checklist Description Format (XCCDF). NIST, March 2012.

Acknowledgements

This work was partially supported by the EU FP7 Univerself Project and the FI-WARE PPP.