HAL
open science

Protection de la vie privée basée sur des ontologies dans un système Android

Johanne Vincent, Tom Dubin, Christine Porquet

To cite this version:

Johanne Vincent, Tom Dubin, Christine Porquet. Protection de la vie privée basée sur des ontologies dans un système Android. APVP 2012 (Atelier Protection de la Vie Privée, 3ème édition), Jun 2012, Ile de Groix, France. 4 p. hal-00811858

HAL Id: hal-00811858

https://hal.science/hal-00811858

Submitted on 12 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Protection de la vie privée basée sur des ontologies dans un système Android

Johann Vincent* † ‡ §, Tom Dubin * † ‡, Christine Porquet* † ‡
*Université de Caen Basse-Normandie, UMR 6072 GREYC, F-14032 Caen, France
†ENSICAEN, UMR 6072 GREYC, F-14050, Caen, France
‡CNRS, UMR 6072 GREYC, F-14032, Caen, France
§SFR, 1 place Carpeaux, Tour Séquoia, 92 915 Paris La Défense
johann.vincent@ensicaen.fr, tom.dubin@ecole.ensicaen.fr, christine.porquet@ensicaen.fr

***Résumé*—Le système Android équipe aujourd'hui plus de la moitié des smartphones vendus [1]. Les utilisateurs de ce système sont donc devenus une cible de choix pour les applications malicieuses qui s'attaquent tout particulièrement à leurs données personnelles et notamment à leur géolocalisation. Cet article présente une solution de protection de la vie privée intégrée au système Android permettant une granularité plus grande dans la gestion des droits. Cette solution, basée sur une ontologie de la vie privée, permet l'interception des requêtes sur les données et l'envoi de fausses informations aux applications tout en garantissant leur bon fonctionnement.**

***Index Terms*—Protection de la vie privée, Android, Ontologies**

I. Introduction

Dans le monde d'aujourd'hui, les téléphones intelligents, ou smartphones sont devenus un outil quotidien pour de nombreuses personnes. Grâce à des connectivités avancées, ils offrent un ensemble varié de services qui vont de l'Internet à haut débit à la télévision ou encore au services de géolocalisation. Tous ces services ont transformé les smartphones en un couteau suisse numérique. Cette centralisation, a attiré les convoitises et beaucoup d'applications malicieuses ont commencé à prendre pour cible ces appareils. Plusieurs études [2], [3] montrent que ce type d'application s'attaque plus particulièrement aux données personnelles de l'utilisateur. Au cours de précédents travaux [4], [5], nous avons proposé une solution à ce problème sous la forme d'un firewall basé sur des ontologies. Nous avons démontré qu'une ontologie, écrite en OWL [6], permet de représenter les politiques d'accès et qu'une base de règles SWRL [7] offre un moyen de décision quant à l'autorisation ou non de la requête d'une application.

L'intégration complète de l'approche restait néanmoins à réaliser. Dans ce papier, l'implémentation du firewall au sein du système d'exploitation Android est présentée. La première section consiste en un état de l'art des solutions de protection disponible sur cette plateforme et la seconde section présente notre proposition utilisant les ontologies comme approche de représentation d'une identité numérique et de raisonnement. Enfin, nous concluons se papier en donnant plusieurs perspectives à ces travaux.

II. Etat de l'art

A. Protection de la vie privée sur Android

La plupart des systèmes d'exploitation disponibles sur smartphone offrent une protection minime des données personnelles. En effet, sur Android les utilisateurs doivent accepter les permissions accordées à une application lors de l'installation de cette dernière. Ces permissions sont présentées sous la forme d'un manifeste qui décrit les accès demandés par l'application et que l'utilisateur doit accepter pour pouvoir l'utiliser. Le problème de cette approche est le choix binaire offert à l'utilisateur : il doit accepter le manifeste ou refuser l'installation. Ce modèle a rendu les attaques sur la protection de la vie privée monnaie courante [8]. Pour rester en phase avec le principe du manifeste mais protéger efficacement les données, plusieurs systèmes ont été proposés. Parmis eux, on peut citer les systèmes AppFence [9] ou encore TISSA [10]. Ces derniers, consistent en des versions modifiées du système d'exploitation android qui ajoutent une surcouche de protection permettant d'empêcher l'accès aux données ou d'anonymiser certaines d'entre elles.

B. Politiques de protection de la vie privée

Les approches mentionnées ci-dessus se basent sur des listes d'accès prédéfinies pour prendre leur décision sur une requête. Cependant, nous pensons que cette approche est trop similaire au modèle du manifeste. En effet, les implications en terme de vie privée des accès et la signification des listes ne sont pas claires. Pour régler ce problème, une approche basée sur les ontologies a été proposée dans [4], [5]. Grâce à leur forme déclarative, les ontologies sont le meilleur moyen de représenter la plétore d'identités numériques des utilisateurs qui jonglent entre plusieurs comptes, avatars, pseudonymes, mots de passe, numéros de téléphone, emails... Les langages du web sémantiques comme OWL [6] sont également particulièrement efficaces pour représenter, raisonner et appliquer des politiques de sécurité. L'approche suivie s'inspire donc des travaux sur Rei et Kaos [11] et propose une représentation efficace des politiques et de la vie privée des utilisateurs. L'ontologie s'articule autour de quatre concepts principaux que sont la *politique* qui contôle l'*action* faite par un *acteur* sur une *donnée* (voir Figure 1). Pour permettre la décision sur l'autorisation, un ensemble de règles écrites dans une variante

de SWRL [7] permet d'ajouter des contraintes à l'ontologie et la classification de cette dernière par un raisonneur permet la prise de décision. La règle de la Figure 2 par exemple, permet d'interdire à une application *TEST* d'accèder à toutes les informations de géolocalisation quelles qu'elles soient.

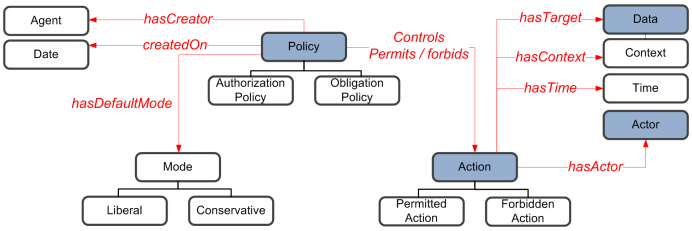

FIGURE 1. Ontologie pour la représentation de politiques

```
[rule:(?action rdf:type id:Action)
(?action id:hasActor id:TEST)
(?d rdf:type ?loc)
(?loc rdfs:subClassOf id:LocationData)
(?action id:hasTarget ?d)
-> (id:currentPolicy id:forbids ?action)]
```

FIGURE 2. Exemple de règle de protection de la vie privée

III. Approche proposée

A. Contraintes d'implémentation

Pour être efficace, notre approche doit respecter un certain nombre de contraintes. La principale d'entre elles est de pouvoir limiter en temps réel l'accès à une information personnelle de n'importe quelle application sans que cette dernière ait besoin d'être modifiée. Le firewall doit donc être implémenté au sein du système d'exploitation Android (voir Figure 3). Il doit impacter au minimum les performances du système et rester transparent vis à vis de l'utilisateur et des applications même si leur requête est interceptée.

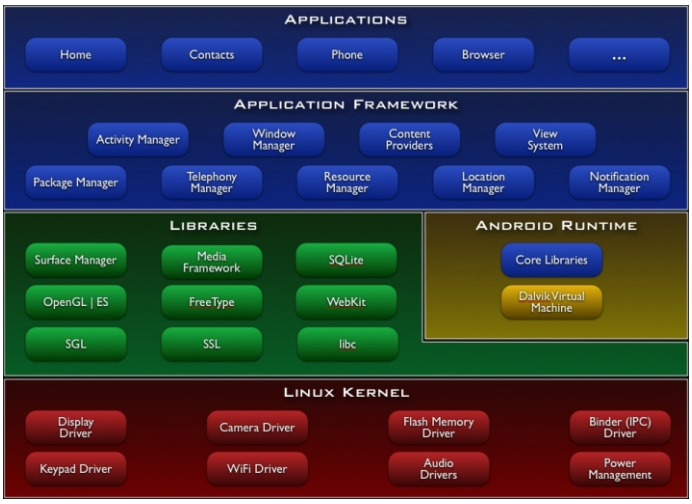

FIGURE 3. Schema de l'architecture Android [12]

Lorsque l'utilisateur lance une application, le système utilise le PackageManager pour vérifier si l'application a bien les droits d'accéder aux données et redirige lors l'application vers les "managers" appropriés (e. g. vers le LocationManager si l'application utilise des outils de géolocalisation). Deux solutions s'offrent alors :

- Il est possible de modifier le PackageManager. Ce PackageManager est appelé à chaque lancement d'une application pour vérifier plusieurs droits d'accès et entre autres, le droit d'accéder aux données du système relatives aux permissions acceptées lors de l'installation. On peut envisager d'ajouter une étape à la vérification des droits d,accès de l'application et d'y implanter notre "manager". Le problème de cette solution est que l'ont perd l'aspect temps réel du firewall ainsi que la granularité fine permise par les ontologies.
- Il est aussi possible de modifier chaque "manager" qui permet à une application d'accéder aux données du système. Le gros désavantage de cette solution est qu'il faut modifier chaque partie du framework qui permet un accès aux données. Par exemple, pour la géolocalisation, il faudrait modifier le LocationManager mais pour les contacts, il faudrait également modifier le ContactsContentProvider, pour les calendrier, un autre provider, etc...

B. Implementation

Bien que plus compliquée en terme d'implémentation, la deuxième solution est la plus adaptée au problème. Elle permet une gestion plus fine et également l'aspect temps réel de la protection. Cette approche consiste en la création d'un "manager" spécialisé appellé OntologyManager qui va être en charge de la prise de décision concernant la vie privée. Ce "manager" est implémenté comme un service système android qui sera lancé au démarrage du téléphone afin d'inclure le temps de chargement de l'ontologie au temps global de démarrage. Afin de ne pas perturber l'experience utilisateur, ce "manager" implémente également l'algorithme présenté dans [5] (voir Figure 4) qui permet en fonction du contexte de limiter les calculs réalisés en local grâce à l'utilisation d'un cache stocké dans une base de données SQLlite ainsi que d'un mode de requête à distance utilisant le protocole HTTPS. Nous avons implémenté ce manager pour la version 4.0.3 du système d'exploitation Android car ce dernier est destiné aux terminaux de dernière génération, plus puissants. Le "manager" est codé en JAVA et fait appel à la librairie Androjena [13] qui permet de manipuler des ontologies au format OWL et de raisonner sur celles-ci.

Pour permettre l'interception des demandes, nous avons expliqué précédemment que les différents "managers" doivent être modifiés pour faire appel à notre service système. Dans ce papier, nous détaillerons uniquement les modifications apportées au système Android permettant d'intercepter les requêtes de géolocalisation sachant que des modifications similaires sont à faire pour les autres "managers". Sur Android, une application qui souhaite obtenir les informations de position s'inscrit auprès du LocationManager (utilisé comme un Singleton). Ce dernier ajoute l'écoutant à une liste interne et le prévient en cas de changement de position. Pour pouvoir

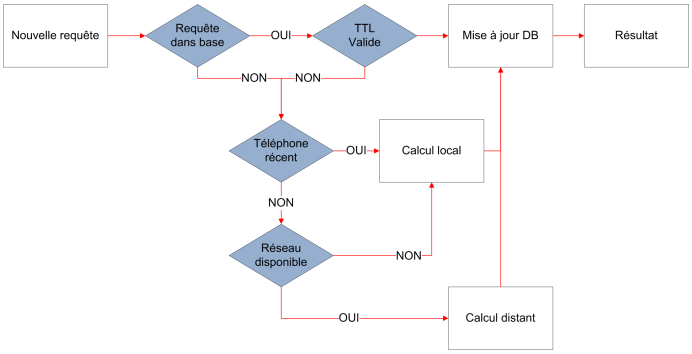

FIGURE 4. Algorithme implémenté par l'OntologyManager [5]

intercepter la demande, le LocationManager doit donc être modifié pour appeler l'OntologyManager avant de fournir l'information. L'appel au système d'ontologies pour autoriser ou non l'accès aux données est réalisé à chaque fois que le LocationManager va envoyer une nouvelle localisation à l'entité inscrite. Pour permettre aux applications utilisant les données de géolocalisation de façon critique de continuer à fonctionner en cas de refus de l'OntologyManager, une fausse position est renvoyée à l'application appliquant le principe de dissimulation de données [9]. Pour identifier une application au sein de l'OntologyManager, il a été décidé d'utiliser le nom du package afin de permettre une sécurité la plus fine possible des applications. L'architecture Android ainsi modifiée est présentée sur la Figure 5. La ROM construite avec ces modifications a permis de valider l'interception des requêtes faites par une application de test sans perturbation des autres applications installées.

IV. CONCLUSION

Dans ce papier, nous avons présenté l'intégration d'un firewall permettant de protéger la vie privée sur smartphone Android. Ce firewall utilise une ontologie décrite en OWL pour représenter les concepts de vie privée sur le smartphone et une base de règles permettant la prise de décision sur cette ontologie. Nous avons montré que pour protéger efficacement la vie privée, ce firewall devait s'insérer dans le système d'exploitation afin d'intercepter les demandes. Nous avons donc présenté les modifications qui ont été faites pour obtenir ce résultat en prennant en exemple le cas de l'accès aux données de géolocalisation.

Pour pouvoir entièrement valider l'approche sur l'utilisation d'ontologies, une étude devra être réalisée sur les performances de ce système en utilisation réelle. En effet, les mesures présentées dans [5] avait été réalisées en utilisant une application de test et les impacts sont donc différents. De même, une étude sur les logs du firewall en utilisation des principales applications disponibles sur la plateforme Android devrait nous permettre de découvrir de nouvelles menaces et ainsi affiner les règles à utiliser dans le firewall.

RÉFÉRENCES

[1] Gartner. Gartner says worldwide smartphone sales soared in fourth quarter of 2011 with 47 percent growth. *http ://www.gartner.com/it/page.jsp ?id=1924314*.

[2] A. Efrati, S. Thurm, and D. Searcey. Mobile-app makers face u.s. privacy investigation.

[3] Lookout. The app genome project. *at http ://blog.mylookout.com/2010/07/introducing-the-app-genome-project/*, July 2010.

[4] J. Vincent, C. Porquet, M. Borsali, and H. Leboulanger. Privacy protection for smartphones : an ontology-based firewall. *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication*, pages 371–380, 2011.

[5] J. Vincent, C. Porquet, and I. Oulmakhzoune. Ontology-based privacy protection for smartphone : a firewall implementation. *International Conference on Secure networking and Applications*, 2011.

[6] D.L. McGuinness, F. Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10 :2004–03, 2004.

[7] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. Swrl : A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21, 2004.

[8] W. Enck, P. Gilbert, B.G. Chun, L.P. Cox, J. Jung, P. McDaniel, and A.N. Sheth. Taintdroid : an information-flow tracking system for realtime privacy monitoring on smartphones. In *Proceedings of the 9th USENIX conference on Operating systems design and implementation*, pages 1–6. USENIX Association, 2010.

[9] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall. These aren't the droids you're looking for : retrofitting android to protect data from imperious applications. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 639–652. ACM, 2011.

[10] Y. Zhou, X. Zhang, X. Jiang, and V. Freeh. Taming information-stealing smartphone applications (on android). *Trust and Trustworthy Computing*, pages 93–107, 2011.

[11] G. Tonti, J. Bradshaw, R. Jeffers, R. Montanari, N. Suri, and A. Uszok. Semantic web languages for policy representation and reasoning : A comparison of kaos, rei, and ponder. *The SemanticWeb-ISWC 2003*, pages 419–437, 2003.

[12] Android Open Source Project. Android architecture. *http ://developer.android.com/guide/basics/what-is-android.html*.

[13] Androjena. Jena android porting. *at http ://code.google.com/p/androjena/*, 2010.

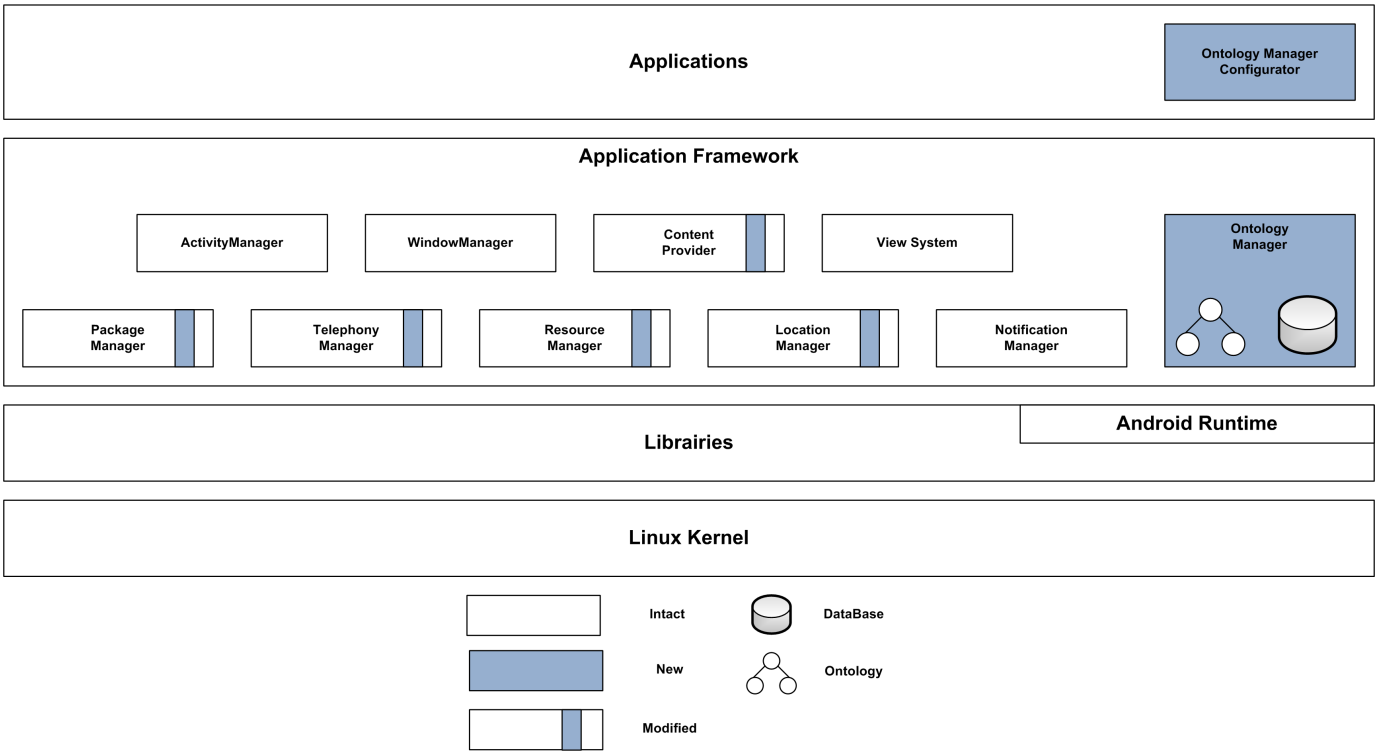

FIGURE 5. Architecture Android modifiée