Third Year Review

Joseph Hallett

Monday 13th February 2017

Thesis

Formal languages can model the natural-language, and informal policies in the mobile ecosystem, and in doing so help users, buidnesses and developers make decisions on the basis of a declarative policy, rather that na informal understanding of their preferences, rules and policies.

(Brief) Review of last year

- Implemented AppPAL interpreter
- Framework for runnng storing static analysis results for AppPAL
- Surveyed app store policies for developers and users
- Investigated whether users install apps that match their privacy preferences
- *Poster at SOUPS '15*
- *Published at ESSoS '16*

Review of this year

- Tweaked AppPAL's syntax for types
- Using AppPAL for BYOD policies
- *Doctoral Symposium Paper at iFM '16*
- *Short publication at IMPS '17 workshop*
- Full paper submisson to IFIP-SEC '17
- Response due on 7th March
- Automatic analysis of policies

Tweaked syntax for AppPAL

Standard Predicate Forms

- An AppPAL policy is a set of *assertions*
- Each assertion has a *speaker* who **says** it
- Every assertion has a *subject* who the speaker says something about
- All assertions have a *verb phrase* which describes what the speaker says about the subject

- **can-say** for delegation
- **can-act-as** for role assignment
- . . . or an arbitrary predicate (possibly with arguments)

[Speaker] ***says*** *[Subject] [Verb] . . .*

'alice' says 'angry-birds' isInstallable.

Arbitrary predicates to standardised predicates

- When writing AppPAL policies have settled into using four kinds:
- *subject is Something* (for types)
- *subject can Something* (for permission)
- *subject has Something* (for completed actions)
- *subject must Something* (for obligations)

SecPAL's safety condition

All variables in the head of an assertion must be present in the body of the assertion.

Repetetive policies

- When writing policies for *privacy preferences* work last year and *BYOD policies* this year, kept encountering the following idiom where the body of an assertion is just typing statements.

```
'user' says Friend can-say App isInstallable
if Friend isFriend,
   App isApp.
```

- If you forget one type you get an error when loading policy into AppPAL.

Typing syntax for AppPAL

Variables in the head are allowed an additional type. When parsing an assertion the type is removed and a condition that the variable *is* Type is added to the assertions body.

```
'user' says Friend:F can-say App:A isInstallable.
```

AppPAL for BYOD

- Companies publish BYOD policies to control devices employees bring to work
- Natural language policies are ambiguous, and hard to compare
- Use AppPAL to describe policies, look for common themes and concerns
- Contrast with MDM capabilities

Some of the policies are really simple

> *SANS: "Only approved third party applications can be installed on handhelds. The approved list can be obtained by contacting the IT department, or should be available on the intranet."*

NHS: "Apps for work usage must not be downloaded onto corporately issued mobile devices (even if approved on the NHS apps store) unless they have been approved through the following Trust channels: Clinical apps; at the time of writing there are no apps clinically approved by the Trust for use with patients/clients.

However, if a member of staff believes that there are clinical apps [. . .] ratification should be sought via the Care and Clinical Policies Group. [. . .] Business apps; at the time of writing there are no business (i.e., non-clinical) apps approved by the Trust for use other than those preloaded onto the device at the point of issue.

However, if a member of staff believes that there are apps [. . .] ratification of the app must be sought via the Management of Information Group (MIG). [. . .] Following approval through Care and Clinical Policies and/or MIG, final approval will be required through Integrated Governance Committee."

MDM software is primative

MDM software is limited in what it can do

- Enable or disable existing Android settings
- Install and track what employees have installed
- Hand picked app store
- App wrapping
- Mostly for adding VPNs to software

Feature	MaaS360	Blackberry BES	MobileIron	Citrix XenMobile	VMWare AirWatch	Microsoft	SOTI MobiControl	Sophos	Landdesk
Antivirus								✓	
App selection/store/management	✓	✓	✓	✓	✓	✓	✓	✓	✓
App wrapping/modification		✓	✓	✓	✓	✓		✓	✓
Authentication	✓	✓	✓	✓	✓	✓	✓		
Compliance reporting	✓	✓	✓	✓	✓	✓	✓	✓	
Device configuration	✓	✓	✓	✓	✓	✓	✓		✓
Email/Calendar/Contacts/Documents	✓	✓	✓	✓	✓	✓	✓	✓	✓
Feature Restrictions	✓	✓			✓	✓			
Licence distribution		✓							
Location based settings	✓				✓				
Network configuration	✓	✓	✓	✓	✓	✓	✓		
Password/Encryption settings	✓	✓	✓	✓	✓	✓	✓	✓	
Remote wipe	✓	✓	✓	✓	✓	✓	✓	✓	✓
Security auditing	✓	✓	✓	✓	✓	✓	✓	✓	
Tracking/Spyware	✓	✓				✓	✓		✓
Watermarking					✓				

Network Settings

Allow Wi-Fi On a Wi-Fi only device, unchecking this option will not block Wi-Fi.	Yes	Android 2.2+
Enforce Wi-Fi is always on	No	Android 2.2+
Bluetooth	User Controlled	Android 2.2+
Allow Data Network	User Controlled	Android 2.2+
Enable Background Data Synchronization Allows applications to sync, send or receive data any time.	User Controlled	Android 2.X & 3.X
Auto-Sync	User Controlled	Android 2.2+
Allow user to Mobile Data limit	Yes	SAFE 4.0+
Allow VPN Allow or disallow use of the native VPN functionality. If disabled, the user cannot establish a VPN session and the UI for using VPN through the Settings application is inaccessible.	Yes	SAFE 2.2+

- Took BYOD policies from 5 sources:
- NHS
- SANS
- HiMSS
- University of Edinburgh
- Company producing emergency sirens
- Translated them into AppPAL

- Seems to be a mismatch between what the MDM tools do and what the policies want
- Policies want employees to acknowledge rules
- Acceptable use, ethical behavior
- Delegations to different departments to set rules
- IT maintain the (un/)acceptable app list

- MDM software seems to be good for:
- Provisioning
- Remote wipe
- Enforcing encryption
- . . . but its all manual

Outcomes

- Two papers written, one presentation
- Doctoral Symposium presentation
- Short paper accepted to IMPS workshop
- Awaiting verdict on long paper in three weeks at IFIP-SEC

Automatic Analysis of Policies

- Given a policy can we identify problems automatically
- Is the policy satisfiable
  - are we missing statements required to decide a query?
- Is there redundancy in a policy
  - is there a simpler way to decide a query?
- Apply to BYOD policies to spot problems

- BYOD policies don't have multiple ways of satisfying policies
- Missing statements mostly my error
- Still think there is value in this
- but mostly as developer tooling

Outcomes

- Implementation is mostly there
- Additional tooling for exploring policies also developed
- Mostly to visualise BYOD policies
- Suspect there isn't enough for a paper
- No real application outside of AppPAL and it rehashes similar Datalog work

Probabilistic AppPAL

- AppPAL assertions with confidences attached
- Spent a few weeks thinking about this and sketching implementation
- Lots of theory, very difficult (independence)
- Suspect I don't have time before my funding runs out.

Thesis

Figure 1:

- Started writing (~50-60 pages, 2 chapters)
- Slipped by about a month on the schedule
- Wrote papers, Jim was two weeks early
- Not worried, had allowed for Jim related delays