Using Authorization Logic to Capture User Policies in Mobile Ecosystems

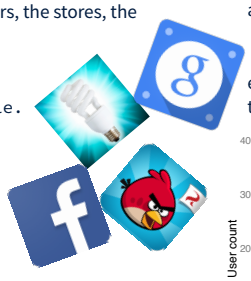
Users pick apps from app stores for many reasons. Sometimes they pick based on what the store shows them first; othertimes it's based on the reviews and information shown about the app; and sometimes it's based on what the app can do and privacy concerns.

People fall into patterns when thinking about the privacy implications of apps. By capturing these patterns explicitily as policies we can enforce them automatically using the AppPAL authorization logic. Users have opinions about the privacy implications of apps but do they follow them when picking which apps to install?

Using data app installation data from 44,000 users we find that very few users seem to be able to follow these policies consistently when installing apps on their devices.

AppPAL

AppPAL describes policies specifying when an app is installable. Statements are relative to specific principals, enabling delegation relationships to be expressed. Delegation is natural in the app store setting: it captures trust relationships among the users, the stores, the developers, and security vendors who vet apps.



Alice, a user, can say an app is installable:

```
"alice" says "com.rovio.angrybirds" isInstallable.
```

She can introduce conditions and delegation relationships:

```
"alice" says App isInstallable
  if "not-malware-policy" isMetBy(App)
  where hasPermission(App, "LOCATION") = false.
"alice" says "mcafee" can-say
  "not-malware-policy" isMetBy(App).
```

Policies for App Installs

In a user study of 725 Android users, *Lin et al.* found four patterns that characterise user privacy preferences for apps: *conservative* and *advanced* users were concerned how their data was used, but advanced users understood some parts of apps need the data to function. *Uncorncerned* and *fencesitters* were more liberal.

We wrote AppPAL policies to describe each of these behaviours as increasing sets of permissions. These simplify the policies; but AppPAL constraints can describe richer policies than permission sets.

	C	A	F	U
GET_ACCOUNTS	✘	✘	✘	✘
ACCESS_FINE_LOCATION	✘	✘	✘	
READ_CONTACT	✘	✘	✘	
READ_PHONE_STATE	✘	✘		
SEND_SMS	✘	✘		
ACCESS_COARSE_LOCATION	✘			

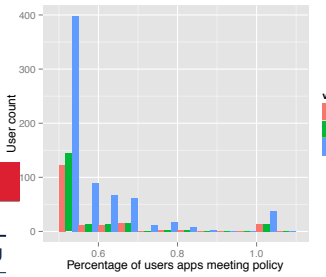
Using AppPAL we can write policies to differentiate between different kinds of malware, characterising users who allow dangerous apps or those who install merely "unsavoury" apps.

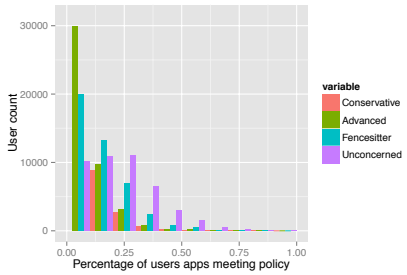
```
"user" says "mcafee" can-say "malware" isKindOf(App).
"mcafee" says "trojan" can-act-as "malware".
"mcafee" says "pup" can-act-as "malware".
```

Do Users Follow These Policies?

Most users seem to use apps irrespective of how uncomfortable they are with the permissions they request.

A small set of users do seem to enforce these policies at least some of the time however.



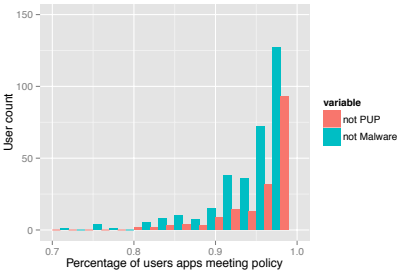


For the unconcerned policy (the most permissive) only 1,606 users (4%) had 50% compliance; and only 120 users (0.3%) had 80% compliance.

For the stricter conservative policy only 60 users were complying half the time, and just 7 more than 80% of the time.

1% of the users had a PUP or malicious app installed. Figure 2 shows that infection rates for PUPs and malware is small. A user is 3 times more likely to have a PUP installed than malware.

9 users had both a PUP and malware installed. Users who were complying more than half the time with the conservative or advanced policies complied with the malware or PUP policies fully.



Experiments

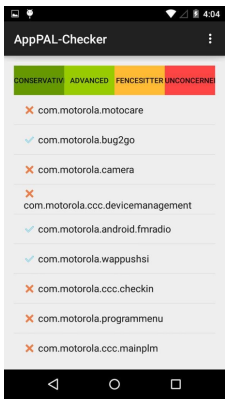
We want to test how well these policies capture actual user behaviour. We also want to know the extent malware is present on Android compared to PUPs. Do users who follow policies install fewer bad apps?

The *Carat* dataset contains anonymised records of what users installed. Each app name is replaced by its hash. Each user replaced by a number. By calculating the hash of known app names we were able to de-anonymise around 4,300 apps of the 90,000 present in the dataset. Disregarding system apps and very common apps this accounted for on average half of every user's installs. From the 55,000 users in the data we selected 44,000 users for whom we knew at least twenty installed apps.

User	com.rovio. angrybirds	?	com.facebook. katana	…
1	✘	✔	✔	
2	✔	✘	✔	
⋮				⋱

We checked which apps met which policies. We measured the percentage of each user's apps that met the policy. This let us measure the extent a user appeared to follow a policy.

Future Work



There is a disconnect between how users feel about permissions and how they act. Future work will look at using AppPAL to model the policies users actually use when selecting apps.

Exploring this disconnect is an avenue for future research. We would like to be able to take a user and their apps and produce a policy that describes their current behaviour. This will let us precisely compare user's behaviour, and compare the differences between their described and actual behavior.

We will let users explore and test policies with an Android app where they can check currently installed apps against predefined policies. We also want to enable automatically curated app stores on the basis of policies, where users and companies can create personalised app stores that are parameterised by AppPAL policies.

THE UNIVERSITY of EDINBURGH

Joseph Hallett
David Aspinall (supervisor)