An Ontology for Enforcing Security and Privacy Policies on Mobile Devices

Brian Krupp, Nigamanth Sridhar and Wenbing Zhao
Electrical and Computer Engineering Department, Cleveland State University,
2121 Euclid Ave, Cleveland, OH 44115, U.S.A.

Keywords: Mobile Devices, Security and Privacy, Ontologies.

Abstract: Mobile devices have experienced explosive growth and rapid adoption. These devices have also become troves of security and privacy data of the consumers that utilize them. What makes mobile devices unique from traditional computing platforms is the additional sensing components they contain and their ease of access which allow consumers to make these devices a part of their lives. Additionally these devices are fragmented in operating systems, sensing capabilities, and device manufacturers. In this paper we define an ontology that can be utilized as a foundation for enforcing security and privacy policies across all mobile devices, and use the ontology to define policies and to model knowledge elements for mobile devices. We also identify areas where the policies can be applied, including whether to enforce policies on the device or in the cloud.

1 INTRODUCTION

Mobile devices continue to grow in prominence in utilization across both consumers and businesses. For vast number of consumers, mobile devices have become their primary communication and entertainment devices in their day to day lives. For example Foursquare claims that there have been over five billion check-ins using their application alone (Foursquare, 2014). Likewise, Twitter claims an average of 5,700 tweets per second (Twitter, 2014).

Mobile devices are equipped with a variety of sensors that may log the activities and locations of their users. Hence, such data can be highly personal and sensitive and must be protected by proper security and privacy policies and mechanisms. The task of protecting such data is made even more challenging due to the need of synchronization of personal data across all devices using cloud based services such as iCloud, OneDrive, Google Drive, etc.

The need for developing and enforcing security and privacy policies on mobile devices can be demonstrated by recent privacy leaks and security exploitations. For example the popular social networking application Facebook was found to leak a user's phone number without a user logging into the application (Symantec, 2014). Another example regarding the popular mobile game Angry Birds not only collects user's privacy data such as their location but this data has also been targeted by agencies such as the NSA (Ball, 2014) to profile users.

In this paper we build on previous research and analysis of the most widely used mobile operating systems to propose an ontology that can be used across all mobile devices to enforce security and privacy policies for consumers. We define how to model the vast amount of knowledge that can be gained from mobile devices both static and inferred from observed activity on the device. We then recognize how this ontology can be utilized and implemented both within and outside the device.

The rest of this paper is organized as follows. In Section 2, we discuss the current state of the art in the field of defining security and privacy policies for mobile devices. In Section 3, we provide an overview of our ontology, and describe it in detail in Sections 4, 5, and 6. We then describe how the ontology could be utilized and enforce user defined security and privacy policies in Section 7. We conclude with a summary of our contributions and some pointers to future work in Section 8.

2 RELATED WORK

There has been various work in constructing ontologies around security and attack behavior with some of these targeted towards mobile. However related

Krupp B., Sridhar N. and Zhao W..
An Ontology for Enforcing Security and Privacy Policies on Mobile Devices.
DOI: 10.5220/0005081502880295
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2014), pages 288-295
ISBN: 978-989-758-049-9

work that focused on mobile has been limited. Beji et al. proposed a security ontology towards mobile devices and took a more traditional security approach by defining semantics around common security elements including *Asset*, *Vulnerability*, *Threat*, etc (Beji and El Kadhi, 2009b). They emphasize that mobile devices lack standards in this area and the ontology they proposed took a general approach without a focus on specific use cases. They then further extended their ontology to take a knowledge based approach (Beji and El Kadhi, 2009a). In both proposals the ontology was formally defined in Web Ontology Language (OWL). Tsoumas et al. took a similar approach in focusing on security and proposed an ontology with a focus more on the security management aspect (Tsoumas and Gritzalis, 2006). Their framework also has a knowledge focus which pulls data from different sources.

Wang et al. proposed an ontology where again the focus was on security but more with an emphasis on vulnerability management (Ju An WangGuo, 2010). They model similar attributes as in Beji et al with a focus on *Attack, Attack Mechanism, Attacker, Product Vendor,* etc. With a focus on being able to manage vulnerabilities, their goal was to identify patterns from external threats and vulnerabilities more formally and precise (Ju An WangGuo, 2010).

Woo et al. also focused on modeling vulnerabilities but took a different approach where they propose an ontology that models behaviors (Woo et al., 2013). They recognize the difficulty to detect a security threat with the different methods a system can be attacked, so they take the approach of looking at behaviors that would identify an attack. Their ontology models classes such as *Actions, Behaviors, Abstract Behaviors*, etc.

Focusing on defining policies, Uszok et al. proposed KAoS which is a system of policy and domain services that allow a computer system to enforce human expressible policies (Uszok et al., 2003). Panagiotopoulos et al. proposed PROACT (Panagiotopoulos et al., 2010), an ontology focusing on privacy in smaller components such as wireless sensor components.

On the contrary to the work described above, we propose an ontology that can be used to enforce security and privacy policies with a focus on mobile devices using a knowledge based approach. Also the utilization of the ontology we propose does not require modification to the operating system and focuses on both privacy and security concerns.

3 ONTOLOGY OVERVIEW

We propose an ontology that consists of the following three main categories: *Policy, Activity,* and *Knowledge*. The relationship between Activity and different type of Knowledge are illustrated in Figure 1.

In the Policy category of our ontology, we model the policies that a user would define for their mobile device, including ones described in more detail in (Krupp et al., 2013; Krupp et al., 2014).

In the Activity category, we model real time activity on the device that includes elements such as location activity, network activity, and device state. Activity is essentially a holding for any activity data that can be transformed into knowledge. Over time if it is not transformed into knowledge, the data is discarded. An example of this transformation would be gathering several locations from the device during the day. If this data doesn't contribute to new knowledge, it will be discarded.

In the Knowledge category we model existing knowledge and inferred knowledge as defined below:

- Existing knowledge models personal data on the mobile device including photos, calendar events, location, etc. The need to model this knowledge is so that the policies defined by the consumer have a consistent model to utilize when the policies are being enforced.
- Inferred knowledge is what can be *inferred* from existing knowledge. Since existing knowledge can be built both from activity and directly from the user's personal data on the device, inferred knowledge may contain previous activity data that was transformed into knowledge. Knowledge that can be inferred includes areas such as usage patterns, network activity, and movement of the user. The two primary drivers of including this in our ontology is both performance and space. In performance, since inferred knowledge summarizes a collection of existing knowledge and activity elements, this saves the enforcing application from having to analyze all elements that made up the inferred knowledge each time it needs to utilize it. Also in regards to space, inferred knowledge allows us to remove existing knowledge elements that may no longer be needed as the inferred knowledge from the elements that made it would no longer be needed. This would be an obvious constraint on an implementation utilizing this ontology if it held a historical record of all activity and existing knowledge without ever disposing the data. This additional subcategory in our ontology prevents the ontology from losing the value of what that historical data provided.

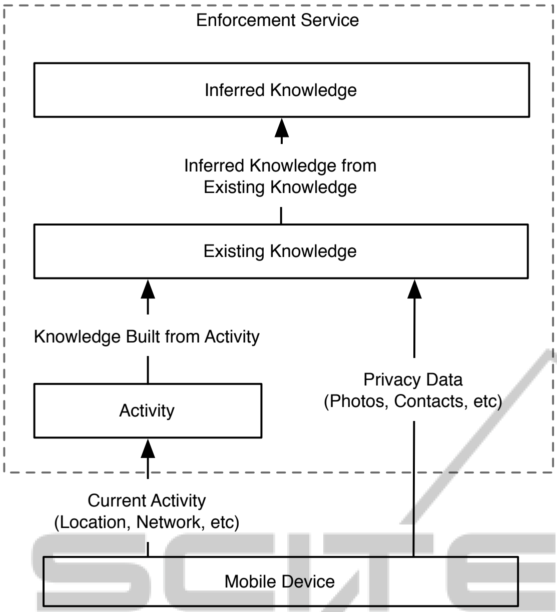
Enforcement Service
Inferred Knowledge
Inferred Knowledge from Existing Knowledge
Existing Knowledge
Knowledge Built from Activity
Privacy Data (Photos, Contacts, etc)
Activity
Current Activity (Location, Network, etc)
Mobile Device

Figure 1: Ontology relationship depicting how data that is modeled exists from one state to another.

For this ontology we identified the most common knowledge and activity elements across the most widely used mobile operating systems. We recognize that there will be outliers that are not covered in this ontology based on OS, device, and sensor capability. However our primary goal is to provide an ontology that can cover a broad spectrum of devices as opposed to one set or the union of all attributes of mobile devices. All of the ontology objects discussed above will be described in greater detail in the following sections. For this paper we define the properties that ontology elements contain, their relationships with other objects in the ontology, and examples of application of the ontology. The ontology we propose here has been formally defined in OWL using Protege.

4 POLICY

The *Policy* portion of our ontology is mainly derived from our previous research building a framework for enhancing security and privacy on mobile devices (Krupp et al., 2013; Krupp et al., 2014). The policies defined in the ontology are divided into three main categories: *Security*, *Privacy*, and *General*. Another type of policy defined in our ontology is the *ChainablePolicy* which links both a security policy and a privacy policy to provide more fine grained policies.

4.1 Security

Our main goal in this category is to define policies around data transferred or persisted, authentication data, and ensuring that they are handled using the proper security mechanisms such as transport layer security or encryption. The policies defined here are targeted for non-jailbroken devices but can be used for jailbroken devices as well. Within the security category we define *DataPersistence*, *NetworkSecurity*, and *UserCredential* security policies.

With *DataPersistence* our primary objective is to ensure that any data persisted on the device would be properly secured using the minimum encryption level. Additionally with the availability of syncing data to cloud based storage providers, we add the ability for a user to specify policies on permitting data to be synced. This policy could be used within corporate networks to ensure that sensitive data is encrypted on mobile devices in case they are lost or stolen, and that sensible data would not leave the local network.

The *NetworkSecurity* policy specifies where data can leave the device as opposed to *DataPersistence* that defines how data can be persisted on the device. This policy contains properties that define what domains the application can communicate to and any transport layer security requirements on the transmission of data. This can be be used to restrict any privacy leakage of an application that sends data on behalf of the application to third party servers (for example, for the purpose of tracking user online behaviors). Additionally for any sensitive information that is in transit, a user can ensure that they are using transport layer security.

Lastly with the *UserCredential* policy our primary objective is to ensure that specific policies around a user credential being transmitted over a network could contain its own restrictions. With this policy, we allow a global setting that would allow the credential to be used by all applications.

4.2 Privacy

Both iOS and Android provide general privacy controls that allow an application access to user's photos, contacts, location, and other personal data. These controls provide an all or nothing access to a specific class of a user's personal data. Currently consumers cannot define finer granular controls such as allowing an application to only gather location data in specific locations, restricting the access to certain photos from an application, or not allowing access to a subset of contact records. With the *Privacy* category, we aim to provide these controls.

We define two general categories for privacy policies which include policies around data that sensors create and policies around data that users create. Within the policies that users can create, we define policies for *Calendar, Contacts,* and *Multimedia*. For all policies we define a generic policy that specifies read/write permissions to these data stores. Additionally *Calendar* and *Multimedia* contain temporal restrictions and *Multimedia* also allows for spatial restrictions. For example, a user could define that an application does not have access to photos that were taken during certain dates or at certain locations. Furthermore, the ontology contains policies that specify whether or not a particular attribute such as timestamp or location of an entity can be accessed by an application. This is important because even if a user does not allow an application access to location data, an application can easily determine a user's location patterns based on the location attribute of the photos taken by the user. With the *Contacts* policy there are several attributes such as name, address, phone number, email, etc. that we define individual settings to protect access to each of these elements.

For all of the policies defined, a user can selectively specify which calendar events, photos and other multimedia, and contacts that cannot be accessed by the application. This allows a user to ensure that a sensitive piece of privacy information is not accessible by a mobile application.

By sensor data, we refer to the data generated by various sensors equipped in a mobile device. While devices can contain accelerometer and gyroscopes, we find that location data is most critical and define a robust set of privacy settings around location. Besides allowing general read access to the location, we define properties to allow the user to specify spatial and temporal constraints on gathering the location. Spatial constraints include specified regions that the user does not want their location to be gathered. This policy could be used to disable location gathering in highly sensitive locations that a user would not want to share. This policy could also enable a user to block out location gathering at certain times during the day to prevent an application from predicting the pattern of the user's movements. Additionally we allow the user to define whether or not regional monitoring (geofencing) is allowed and when gathering the location data if it should be anonymized, generalized, or provided as bogus data. Bogus data may be provided when the application requires a location to be provided to function, regardless of what that location data is. Generalized would give a consistent location within a given region for applications that do not need the specific location to provide their functionality such as a weather forecasting application.

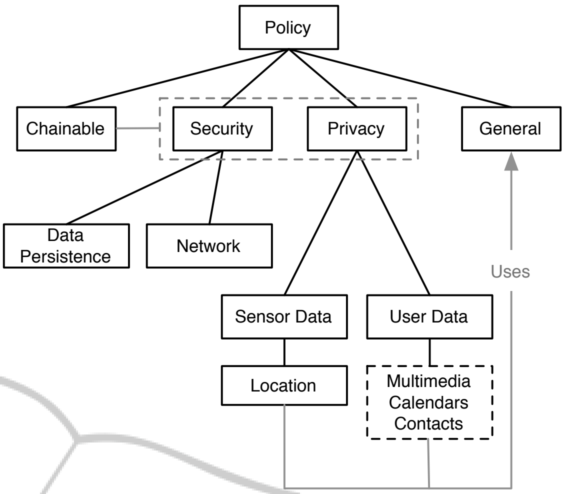

Figure 2: Ontology relationship depicting how policy is organized in the ontology.

4.3 General

The *General* policy encompass temporal and spatial policies that would be used as described in the above policies.

4.4 Chainable

The primary goal of a *Chainable* policy is to allow a user to attach several security policies to a privacy policy. This allows a user to specify that an application can gather its location but gathered location data cannot be transmitted over a network. The policy also enables a user to allow an application to access a user's photos but not write them to the local disk unencrypted. A *Chainable* policy gives the user a deeper level of control of their privacy data by allowing security and privacy policies to be combined together, as shown in Figure 2.

5 ACTIVITY

With the *Activity* category our objective is to model common activity that can be gathered from how the user interacts with the device, and to transform such activity into knowledge. Activity usually represents the current state of the device. We specify the following Activity policies: *Network Status*, *Location*, *Battery*, *Network*, *Motion*, *Multimedia*, and *Bluetooth*. Most of these policies are self explanatory in what activity they model, however some need additional clarification below.

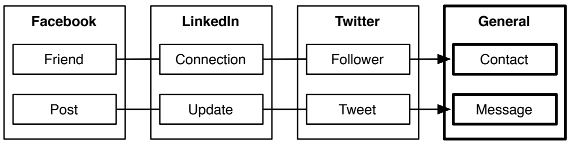

Figure 3: Shows generalization of integrated application data.

The *Multimedia Activity* policy does not contain the same attributes as *Multimedia Knowledge*. It is more concerned with if microphone/camera is enabled, or is being recorded or replayed. *Network Status Activity* models what kind of network connection the user has, and if they are connected using WiFi to what SSID the device is connected. With *Motion Activity*, we specify how motion data can be used to help in determining location and movement from one location to another.

6 KNOWLEDGE

The *Knowledge* category encompass both *Existing Knowledge* and *Inferred Knowledge*. *Existing Knowledge* is gathered from personal data on the device as well as activities on the device, such as current location, network activities, etc. From *Existing Knowledge*, we can infer deeper knowledge, which we refer to as *Inferred Knowledge*, such as the pattern of the user's movement with respect to the time of day, or the kind of personal data the user typically shares with others. In the following sections we further describe this part of the ontology.

6.1 Existing Knowledge

Existing Knowledge is used to model data that exists on the device that is to be persisted as part of the enforcement of a user's given policy. *Existing Knowledge* can be further divided into *Contacts, Credentials, Multimedia, Integrated Applications, Events,* and *Sensing*.

Contacts, Multimedia, and *Credentials* model user supplied data, and they would need to be updated as needed from what the user has on the device so that when enforcing a policy, there is a model representing this data. *Integrated Applications* is more concerned about applications that have ties into the operating system such as *Facebook*, *Twitter*, and *LinkedIn*. The data that these applications provide can be generalized into a model that includes contacts and messages as depicted in Figure 3.

Events include *Calendars* and *Task Lists*. Both are very similar in nature but have subtle differences in that certain elements such as a date and time one *must* contain and the other *can* contain. For example, both a calendar event and a task list contain a title and some additional notes. A calendar always has a date and time associated with it while a task list can contain those attributes as well if there is a reminder event to be fired. Additionally both a calendar and a task list item can have a location associated with it where the calendar event could be the location of the event and a task list could be a reminder that is triggered when the user enters a geofenced location.

6.2 Inferred Knowledge

Inferred Knowledge constitutes valuable elements for the enforcement of security and privacy policies specified for a mobile device. *Inferred Knowledge* includes *Spatial Data*, *General Usage*, and *Network Activity*. *Spatial Data* further consists of *Location at Time of Day* and *Movement from Location to Location*. The former is built from gathered location activity over time and can be used to predict the user's location given the time of day. The latter is also built from location data and time, and can be used to predict the user's movement behavior. Both allow the application that enforces the policy (referred to as "enforcing application" in later text) to determine if an application is tracking the user's travel patterns, and prevent such personal data to be leaked. Additionally both allow the enforcing application to predict where a user may be at a given time and ensure that location data is not leaked if the enforcing application does not have access to the user's current location. This prediction would become more accurate as more location activity data is provided.

General Usage captures the knowledge regarding usage patterns. *Time of Day Device is Used Most* and *Location Device is Used Most* can be used to help find opportune times and locations when a device is less utilized to carry out more resource intensive operations such as synchronizing data from the device to an enforcement application. Additionally the enforcement application would know from the most utilized locations and date times when it would need to allocate additional resources.

General Usage also includes *Application Used at Location*, *Application Utilized During Day and Time*, and *Application Currently Utilized. Application Currently Utilized* is determined from the network activity on the domain, attempted personal data access, and data persistence. This allows the application enforcing policies to understand what mobile application is currently active and enforce the appropriate policies. If the enforcing application is uncertain what mobile application is currently being utilized, it can use in-

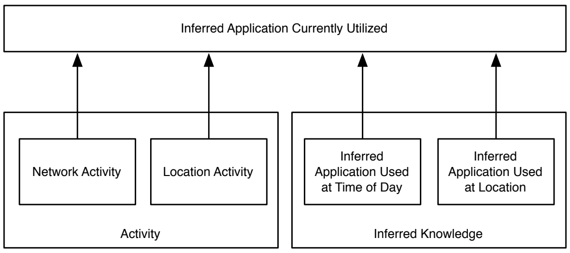
Inferred Application Currently Utilized
Network Activity
Location Activity
Inferred Application Used at Time of Day
Inferred Application Used at Location
Activity
Inferred Knowledge

Figure 4: Various inputs into building inferred knowledge from activity and previous knowledge.

ferred knowledge from *Application Used at Location*, *Application Utilized During Day and Time*, to determine what application is active. In each scenario, if there is a high degree of confidence from the determined application, the detected application would feedback into the inferred knowledge around application utilization (Figure 4).

Network Activity includes *Domains Visited at Location/Day and Time*, *High Network Activity on Network Level*, *Privacy Data to Send*, and *Large Privacy Data Upload*. By inferring what domains are visited at a location and time, we can determine what application is currently being utilized on the mobile device if we are unable to determine it from the current activity on the device. With inferring the domains, we can then determine what data may be sent and what policies to enforce. By inferring how much network activity is expected given a type of network connection (Wi-Fi, Cellular), the enforcing application can detect abnormalities in the amount of data that is being transferred and designate appropriate resources to enforce the user's policies. For example if a user usually only uploads several or more photos on a Wi-Fi connection, if we infer that there is a high level of network activity on a cellular connection we can investigate further to determine if the abnormality is violating any security or privacy policies.

By inferring what personal data is sent for a given application, the enforcing application can expect what additional personal data may be sent when that application is used. This again allows the enforcing application to anticipate activities, which strengthens its enforcement of the policy. *Large Privacy Data Upload* can be inferred by looking for personal data that may have been sent out over a longer period of time by an application as opposed to being sent all at once. The inferred knowledge can be used to detect stealth attacks that leak privacy data.

7 ONTOLOGY ENFORCEMENT

In the previous sections we defined the ontology that can be used to define security and privacy policies across mobile devices. In this section we address how the ontology can be enforced and applied as well as address some potential issues in the implementation of ontology enforcement. In this section we outline a service-based ontology enforcement approach. Unlike previous research on detecting leakage of privacy information from a device which has traditionally required modification to the operating system, we aim to remove this requirement to lower the barrier of effective enforcement of security and privacy policies. We believe that having the enforcement as a service outside of the device allows the user's mobile experience to remain unaffected by not requiring consumption of the the device's power as well as not produce delays in the user experience by performing computationally expensive operations on the device to enforce the user's defined policies.

7.1 Ontology Enforcement Service

An example implementation of the ontology enforcement service is illustrated in Figure 5. The service consists of two main components. The first component is a web based proxy service that is responsible for maintaining the model of the ontology and enforcing the defined policies by examining the intercepted data originating from the mobile device. The second component is a client application that is responsible for reading system-level data from the device that the proxy service would be interested in, and uploading the data to the proxy service. We describe these two components in more detail in the following sections. Note that all networked communications from the applications in the device are routed to the proxy service.

7.1.1 Client Application

The client application on the device is responsible for ensuring that system-level data on the device is synchronized with the web based proxy device, as shown in Figure 5. For example, when a user adds new photos on the device, the client application needs to ensure that this photo exists on the service as quickly as possible. The client application is also responsible for occasionally polling activity data on the device and sending it to the service. The activity data is needed for both knowledge building and for the proxy service to enforce spatial and temporal parameters in the predefined policy. Furthermore, the client application is responsible for authenticating the remote service to

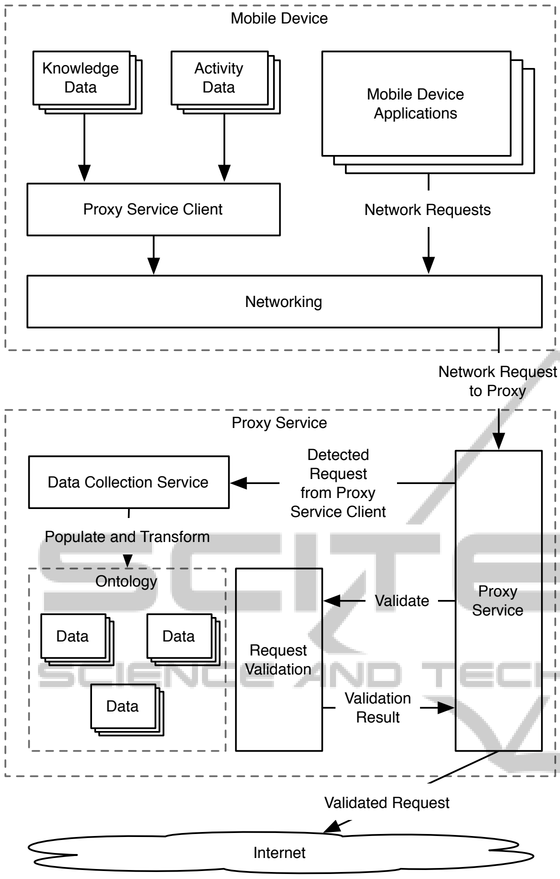

Figure 5: Example implementation for enforcing ontology.

ensure that data is not leaked to a service imposing as the dedicated service that the client is utilizing. It is inevitable that there will be a large amount of data synced to the proxy service from the client to facilitate the modeling of a large amount of existing knowledge in the ontology.

7.1.2 Proxy Service

For regular consumers, the proxy service can be offered as a public cloud service. However for organizations that provision mobile devices to their employees, it may be more attractive to deploy the proxy service on-premise where global policies could be defined and the service could be easily scaled.

In either deployment scenario, the proxy service is responsible for examining networked traffic from the mobile device and determining if the data that it is receiving are requests from the client application on the device or requests from other applications on the device, as illustrated in Figure 5. When updates are received from the client application, they are passed on to a data collection service that is responsible for building the ontology model based on attributes of the data that it receives. The data collection service may need additional data from the client to make a decision, in which case, such information is indicated in the response to the client.

If the request is detected to be from any other application, the proxy service delegates the validation of the request to a request validation service. The request validation service examines the data in the request and it determines if there are any corresponding knowledge entities in the request using the ontology. Based on this determination, it then locates the policies that correspond to the identified knowledge entities in the request and validates whether or not the request should be made. This is by far the most critical piece in the service and where our future research will target as this service needs to be as efficient as possible to ensure that the user does not experience a noticeable delay.

7.2 Discussion

As we recognized earlier, the request validation component of the proxy service may be computationally expensive. This will be the primary focus of our future research so that we can minimize the computation needed to allow for timely enforcement without a perceived user delay. We also recognize that the policies we defined in this ontology may not be ideal for all consumers to create and manage. However, consumers seeking the additional fine grained control can add this service and build these policies. Alternatively, organizations that would have the proxy service on premise could define and manage these policies for use across their entire workforce. In any scenario, we recognize that managing the policies needs to be as simple and intuitive as possible for user adoption.

The client application must keep the proxy service in sync with data that exist on the device. This requirement raises two primary concerns. One concern is that there must be a level of trust to the proxy service as it would hold a large amount of data from the device. This may seem impractical. However, cloud based services that synchronize data across devices are pervasive today, such as OneDrive, iCloud, DropBox, and Google Drive. This does stress the importance of establishing trust with the proxy service and performing authentication both on the client and the service. The other concern is the ability to synchronize the data efficiently and at periods that do not affect the user experience. We envision that the client application would have to be ”smart” and take advantage of opportunistic periods where utilization is perceived low and the device is either plugged in or has sufficient battery power. The most opportune periods for performing sync can also be determined from the inferred knowledge that is built on previous usage activity.

Within the proxy service there would also need to exist a background service that performs necessary transformations of activity data into existing knowledge and existing knowledge into inferred knowledge. Similar to the client application, this transformation service should take advantage of periods that are determined to be most opportunistic to perform these transformations so that the user's experience is unaffected. From these transformations, validation of intercepted network requests should become more efficient during more active periods.

8 CONCLUSION AND FUTURE WORK

In this paper we proposed an ontology by examining previous research and analyzing the most widely used mobile operating systems for common attributes. The ontology we propose can be used to enforce security and privacy policies on a mobile device using a knowledge based approach. The key difference in the ontology we propose here compared with previous research is that it focuses on enforcement of policies as opposed to detection of security vulnerabilities as was the primary focus of the related work we identified. Additionally the ontology we propose along with how it can be enforced does not require modification to the operating system. Our ontology also focuses on privacy concerns as well as security concerns.

For elements in the ontology such as activity and existing knowledge, we recognize that personal data and sensing components would have to be made available to the enforcing application. The determination of possible policy violations can be performed locally at the mobile device, or via a cloud service. We are working on implementing such a cloud enforcement service based on the proposed ontology. We take this approach because it may consume less resources for policy enforcement compared with the local enforcement approach. Furthermore, with most modern mobile operating systems implementing sandboxing controls, any enforcing application that lived locally on the mobile device would need to violate the sandboxing mechanism to inspect data being leaked from the device to enforce the user's defined privacy and security policies. Therefore, a cloud based enforcement service is more feasible, which will be our focus in future work.

REFERENCES

Ball, J. (2014). Angry birds and 'leaky' phone apps targeted by nsa and gchq for user data. http://www.theguardian.com/world/2014/jan/27/nsa-gchq-smartphone-app-angry-birds-personal-data.

Beji, S. and El Kadhi, N. (2009a). A knowledge based process proposal for mobile security. In *Developments in eSystems Engineering (DESE), 2009 Second International Conference on*, pages 166–172.

Beji, S. and El Kadhi, N. (2009b). Security ontology proposal for mobile applications. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09. Tenth International Conference on*, pages 580–587.

Foursquare (2014). About foursquare. https://foursquare.com/about.

Ju An WangGuo, Michael M.Camargo, J. (2010). An ontological approach to computer system security. *Information Security Journal: A Global Perspective*, 19(2):61 – 73.

Krupp, B., Sridhar, N., and Zhao, W. (2013). A framework for enhancing security and privacy on unmodified mobile mobile operating systems. In *The First International Workshop on Mobile Cloud and Social Computing*.

Krupp, B., Zhao, W., and Sridhar, N. (2014). Tell me the truth! what is your intent with my mobile data? Technical Report TR-CSU-ECE-1411, Electrical and Computer Engineering, Cleveland State University.

Panagiotopoulos, I., Seremeti, L., Kameas, A., and Zorkadis, V. (2010). Proact: An ontology-based model of privacy policies in ambient intelligence environments. In *Informatics (PCI), 2010 14th Panhellenic Conference on*, pages 124–129.

Symantec (2014). Norton mobile insight discovers facebook privacy leak. http://www.symantec.com/connect/blogs/norton-mobile-insight-discovers-facebook-privacy-leak.

Tsoumas, B. and Gritzalis, D. (2006). Towards an ontology-based security management. In *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, volume 1, pages 985–992.

Twitter (2014). New tweets per second record, and how! https://blog.twitter.com/2013/new-tweets-per-second-record-and-how.

Uszok, A., Bradshaw, J., Jeffers, R., Suri, N., Hayes, P., Breedy, M., Bunch, L., Johnson, M., Kulkarni, S., and Lott, J. (2003). Kaos policy and domain services: toward a description-logic approach to policy representation, deconfliction, and enforcement. In *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on*, pages 93–96.

Woo, S., On, J., and Lee, M. (2013). Behavior ontology: A framework to detect attack patterns for security. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 738–743.