Common Concerns in BYOD Policies

Joseph Hallett
University of Edinburgh

David Aspinall
University of Edinburgh

Abstract—**Companies publish BYOD policies for employees to use their device at work. These policies are written using natural language, and vary in length and style. Using BYOD policies from five different sources we explore common concerns. Existing tools for enforcing policies has focussed on restricting app and device functionality. Our work looks at 5 BYOD policies and presents the common concerns and structures found in the policies themselves. This suggests where future MDM tools should focus their efforts.**

I. Introduction

Many employees bring their personal mobile devices to work. To control the access these devices have, 70% of companies publish *bring your own device* (BYOD) policies [16]. These BYOD policies are written documents for employees to read and follow. They describe steps to take to secure devices appropriate to the workplace. The policies say how employees should access data, and who should authorise decisions.

Companies have a variety of means to implement their policies. Some companies may trust employees to follow the rules on their own. Alternatively *Mobile Device Management* (MDM) software can implement part of the policies: packages such as IBM's MaaS360 and Blackberry's BES [9, 17] can configure devices to restrict functionality and manage apps. Research has looked at developing other tools such as UC-Droid [12] or BYODroid [2] which was used to implement parts of a NATO BYOD policy [1].

Commercial tools are limited in what polices they can enforce. Some tools can only enable simple on-off configuration settings, and ban explicitly black-listed apps. More sophisticated systems can use app-rewriting to recompile apps to tunnel traffic through a VPN, or geofencing to apply policies in predefined areas. These tools are not infallible. One survey found that 50% of companies with MDM software still had non-compliant devices in their networks [13]. Whilst app wrapping can protect some apps, in general it is ineffective [7].

MDM software and research has, so far, focused on restricting apps and device functionality. But what are the concerns and restrictions described in the policies themselves? By analysing 5 BYOD policies, we present the common concerns and structures found in the policies themselves. The policies display various styles for writing policies, and differing concerns. In comparison to MDM software, the policies do not focus exclusively on restrictions but rather require employees acknowledge company rules and the policies describe trust relationships amongst different departments. This suggests little consensus on the best way to write such policies and that more research is needed.

II. BYOD Policies

We analysed 5 policies from different sources to find common concerns. These five were chosen as they come from a variety of sources and industries. Two are advisory, published specialist institutions to help other companies implement their own policies. Three are policies used in a hospital, a company selling emergency sirens, and a university.

- The SANS policy [14] is a hypothetical BYOD policy that a company could use as a starting point to base its own policy on. It is prescriptive and long, focussed on technical restrictions such as disabling device features.
- The HiMSS policy [8] also is a hypothetical policy, for US hospital trusts planing to implement a BYOD policy. It is short, but and written from the perspective of the employee agreeing to the policy. This is different to the other policies where instead the policy is written as rules for employees to follow.
- The NHS policy [10] is a BYOD policy used in a British hospital trust. It is long, and describes a complex policy in a large organisation with a complex hierarchy.
- The final two policies are simpler, the fourth is taken from a company selling emergency sirens for cars [4], the fifth is by the University of Edinburgh. Both are short, relatively uncomplicated, but typical examples of policies found *in the wild*.

Not all the policies are written in the same style. Most are written from the perspective of a company or IT department telling an employee what to do. The HiMSS policy, however, is written as a contract where the user tells the company how they will behave. Most policies separate individual policy rules into small individual rules each which must be followed. The Edinburgh policy groups them together into two or three large rules, with different increasing sets of sub-rules for low and high risk employees to follow.

III. Translation to SecPAL

To help compare the policies, they were first translated them into a formal language [6]. We use a dialect of SecPAL [3] that we used for describing app privacy preferences [5]. SecPAL is designed to be readable, and requires an explicit authority to *speak* individual statements. This allows SecPAL-based languages to capture delegation relationships and the differences in style between policies. Understanding the contents of policies and making comparisons is easier when using SecPAL as it helps to remove the ambiguities of natural language [6].

Each of the policies are split into a series of rules. The SecPAL used to translate the rules has a structure similar to

Datalog. An authority (the speaker) will decide that a fact is true, if it can be convinced of a series of conditional facts are also true.

$$\langle\texttt{speaker}\rangle\ \texttt{says}\ \overbrace{\langle\texttt{X}\rangle\ \langle\texttt{predicate}\rangle}^{\text{fact}}\ \texttt{if}\ \overbrace{\langle\texttt{Y}\rangle\ \langle\texttt{predicate}\rangle}^{\texttt{condition}}\cdots.$$

A simple example is the following example from the Sirens-company policy. The policy states that devices may access various company resources. For each resource we create a SecPAL assertion that states that a device can access it.

Sirens: *Employees may use their mobile device to access the following company-owned resources:*
● Email ● Calendars ● Contacts ● Documents ● Etc.

```
'department' says Device:D canAccess('email').
'department' says Device:D canAccess('calendars').
'department' says Device:D canAccess('contacts').
'department' says Device:D canAccess('documents').
```

A more complex example can be taken from the NHS policy. Employees are not allowed to call non-domestic, or premium rate numbers on company-owned phones, however an exception can be made if approved by the appropriate manager. To implement this we have the default rule that international calls are banned, a second rule stating that it is allowed if an exemption is made, finally a third rule delegating the exemption making process to the employee's manager.

NHS: *All mobile devices will be configured for national access only. Premium/international calls will be barred. International call barring and roaming arrangements can be lifted for specific periods, to be stipulated on request, on approval of the relevant manager/budget holder.*

```
'nhs-trust' says Device canCall(TelephoneNumber:X)
  if Device isOwnedBy('nhs-trust'),
     X isNationalNumber, X isStandardRateNumber.

'nhs-trust' says Device canCall(TelephoneNumber:X)
  if Device isOwnedBy(Staff),
     Staff hasCallExemption.

'nhs-trust' says Manager can-say
  Staff hasCallExemption
  if Manager isManagerOf(Staff).
```

The formalisation of the policies[1] and tooling for our variant of SecPAL[2] are available online.

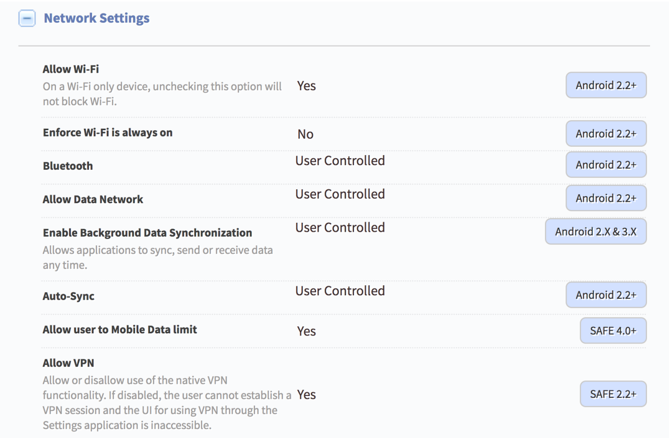
IV. Rule Structures

The predicates used in the formalisation of the rules fall into 4 categories. *Can* predicates describe what their subjects can do; for instance whether a device can connect to a server. *Must* predicates describe obligations, such as reporting a lost device. *Has* predicates ensure an action has been completed in the past, such as approving an app. Finally, *is* predicates describe a typing property about their subjects.

The occurrence of each type of predicate is shown in Table I. The use of each is also split by whether the predicate is a *decision* made by the policy, or a *condition* for making that decision. *Can* and *must* decisions feature in all policies excepting *can* decisions in the Edinburgh policy, in part due to the structure of the policy as discussed in section II. This is expected, these are access control decisions and reactions to events; both topics that existing MDM tools have focused on implementing. *Has* and *is* predicates are the majority of the conditions, but there are also decisions using them too.

	Decision				Condition			
Policy	Can	Must	Has	Is	Can	Must	Has	Is
SANS	35% (35)	29% (29)	9 % (9)	27% (27)	2% (2)	2% (2)	8% (8)	87% (87)
HiMSS	21% (21)	41% (41)	31% (31)	7 % (7)	0	0	13% (13)	87% (87)
NHS	19% (19)	26% (26)	33% (33)	23% (23)	2% (2)	0	19% (19)	83% (83)
Sirens	27% (27)	45% (45)	11% (11)	16% (16)	2% (2)	7% (7)	2% (2)	89% (89)
Edinburgh	0	18% (18)	82% (82)	0	7% (7)	7% (7)	50% (50)	37% (37)

TABLE I
Occurrences of predicate-types in each policy.



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+

Fig. 1. Example policy from the MaaS360 MDM software.

Existing MDM tools present policies as a series of tick-boxes for what a device *can* and *must* do (Figure 1). An administrator selects which policies each device must follow, a predominantly manual process. In our examination of the policies, as well as finding rules that describe what a device *can* do, we find rules that group devices by what they *have* or *are*. Selecting which restrictions to apply to a device is defined by policies; but existing MDM tools do not allow policies to be selected on the basis of policies. MDM tools perhaps need greater flexibility to fully implement all aspects of a BYOD policy.

V. Common Concerns and Checks

Analysing each of the policies common concerns become apparent. A summary of predicates, with the same meaning, used in multiple policies by our translation is given in Table II.

Acknowledgements, where individuals are asked to acknowledge other policies, and predicates linking devices to owners are used in all policies. Most policies described rules for when device features should be enabled and disabled. Configuring device features is a common feature to many MDM packages, but tracking what a user agrees to is not seen in leading MDM packages like MaaS360 or BES [15]. Only 2 out of 5

[1] https://github.com/apppal/apppal-byod-policy-translations
[2] https://github.com/apppal/libapppal

Predicate	SANS	HiMSS	NHS	Sirens	Edinburgh
mustAcknowledged	✓	✓	✓	✓	✓
hasAcknowledged	✓	✓	✓	✓	✓
isOwnedBy	✓	✓	✓	✓	✓
isDevice	✓	✓	✓	✓	✓
mustDisable	✓		✓	✓	✓
isLost	✓	✓	✓	✓	
isEmployee	✓		✓	✓	✓
isApp	✓	✓	✓	✓	
isActivated	✓	✓	✓		✓
mustEnable	✓	✓		✓	
mustWipe		✓	✓	✓	
isEncrypted	✓		✓		✓
hasMet	✓		✓		✓
canMonitor	✓		✓	✓	
mustInform	✓		✓		
isTelephoneNumber	✓		✓		
isString			✓	✓	
isSecurityLevel	✓	✓			
isInstallable	✓		✓		
isFeature	✓			✓	
isData	✓			✓	
isApprovedFor		✓	✓		
isApproved	✓	✓			
hasFeature			✓		✓
hasDevice		✓	✓		
hasDepartment		✓			✓
canUse	✓		✓		
canStore	✓	✓			
canInstall	✓		✓		
canConnectToServer	✓		✓		
canConnectToNetwork	✓			✓	
canConnectToAP	✓	✓			
canCall	✓		✓		
canBackupTo		✓			✓

TABLE II
OCCURRENCES OF PREDICATES COMMON TO MULTIPLE POLICIES.

	SANS	HiMSS	NHS	Sirens	Edinburgh
Rules in policy	33	15	56	20	25
Rules using acknowledgement	6% (6)	67% (67)	20% (20)	24% (24)	5% (5)
Rules using delegation	70% (70)	33% (33)	59% (59)	52% (52)	10% (10)
Rules describing a restriction	54% (54)	20% (20)	14% (14)	20% (20)	5% (5)

TABLE III
SUMMARY OF IDIOMS IN EACH OF THE POLICIES.

policies had rules limiting what networks, servers, or access points a device could access; and only the two most complex policies had rules limiting what apps could be installed. This is surprising as a common feature of MDM tools is controlling how devices and apps access networks. Users have privacy preferences about apps [11], but not all companies try to control what apps employees install. Providing curated app stores and blacklisting apps is a feature common to many MDM programs. Not all policies express rules about which apps to install, however.

All the policies we looked at required employees to be aware of and acknowledge the existence of other policies. The use of acknowledgements is noteworthy because policies acknowledged may not be ones that are enforcible automatically. These other rules include ethical or legal guidelines and disclaimers about data-loss. Writing software to check a user is aware they may lose the data, and is behaving ethically may not be possible.

An aspect of acknowledgements is that they require a delegation of trust from the company to their employees. The employees have to be responsible for stating what they do or do not acknowledge. Delegation is key to other policy decisions in the policies. The IT department is delegated to audit apps. Users are responsible for reporting their device missing.

We summarise the number (and percentage) of rules in the policies using acknowledgements and delegation relationships in Table III. For comparison, we also give the at the number of rules which describe some form of restriction. Delegation relationships form a significant proportion of all the policies. Acknowledgements are used extensively in the HiMSS policy, but rarely in the SANS policy. Overall acknowledgements form as much a part of the policies as do device restrictions. MDM software has focussed on implementing device restrictions and configurations; but it would seem that other aspects are equally important.

VI. AUTHORITIES AND DELEGATION

Each of the policies use delegation to describe rules. A delegation requires at least two parties: someone to hand off the decision, and someone to hand the decision to. They can be individuals, but often they are a role that several individuals may fulfil When translating the policies into SecPAL we created an authority to deliver the policy. In most cases we took the company (who had authored the document) as the authority. In the HiMSS policy, however, rules are phrased as a user stating what they will do.

Most policies used three authorities to make the bulk of the decisions. A primary authority expresses the bulk of the policy and delegation relationships. They act as the *voice* of the policy. The primary authority delegates to the technical authority for some decisions. They may maintain inventories of devices, approve apps for devices, and describe what users may connect to. Their job varies between policies, but in general they are delegated to in order to provide more detailed policy rules. The user is responsible for stating who they are, what devices they have, and what the status of their device is (is it lost or no longer required, for instance).

Some policies have more authorities, than others (Table IV). The NHS policy has various managers that approve decisions for their staff. There are different groups that make decisions for the clinical and business halves of the business. If a clinical user wishes to use an app with a patient they must seek approval from two policy groups, as well as their line manager. Others make less use of different authorities. In the Edinburgh policy, the records-management office states how a low or high risk

SANS	Authorities	10
	Primary Authority	company
	Technical Authority	it-department
	User Authority	user
HiMSS	Authorities	3
	Primary Authority	user
	Technical Authority	xyz-health-system
	User Authority	department
NHS	Authorities	11
	Primary Authority	nhs-trust
	Technical Authority	it-department
	User Authority	staff
Sirens	Authorities	4
	Primary Authority	department
	Technical Authority	it-department
	User Authority	employee
Edinburgh	Authorities	2
	Primary Authority	records-management
	Technical Authority	
	User Authority	employee

TABLE IV
SUMMARY OF DIFFERENT AUTHORITIES IN POLICIES.

must be configured. There is no delegation to others to further specify aspects of the policy. Delegation of responsibilities is an important part of BYOD policies. MDM software seems largely to ignore it, however. These tools instead allow IT staff to set fixed policies and push them to devices. No further requesting of information is typically needed or required.

VII. CONCLUSIONS

We have looked at 5 different BYOD policies and presented a summary of their contents, their styles of presentation and the relationships within them. To analyse the policies we translated them into SecPAL. The translation from natural to formal language is somewhat subjective. We have tried to mitigate this by attempting to preserve the style of the original and by careful use of predicates between different policies. Working with a company to implement the SecPAL policy inside their business would help to ensure the formalisation is correct.

Comparing the policies we found a diverse range of concerns. Some of these concerns (the use of acknowledgements in particular) are not addressed by current BYOD tools. The tools focus on device configuration, which are only part of the concerns of the BYOD policies. We also found the policies were presented in a variety of styles. Some, like SANS, are prescriptive and describe a company telling employees what to do. Others, like HiMSS, take the form of a user contract, where employees state that they acknowledge the companies rules and agree to follow them. The lack of commonality suggest there may be no perfect solution for implementing BYOD policies. Future tools must be flexible enough to express a range of policies, and model the trust relationships these policies contain.

REFERENCES

[1] A. Armando et al. “Developing a NATO BYOD security policy”. In: *International Conference on Military Communications and Information Systems*. May 2016.

[2] A. Armando et al. “Enabling BYOD through secure meta-market”. In: Aug. 2014.

[3] M. Y. Becker, C. Fournet, and A. D. Gordon. “SecPAL: Design and semantics of a decentralized authorization language”. In: *Journal of Computer Security* (Jan. 2010).

[4] Code3PSE.org. *Sample BYOD Policy*. URL: http://www.code3pse.com/public/media/22845.pdf (visited on 10/14/2016).

[5] J. Hallett and D. Aspinall. “AppPAL for Android”. In: *Engineering Secure Software and Systems*. Apr. 2016.

[6] J. Hallett and D. Aspinall. “Specifying BYOD Policies with Authorization Logic”. In: *PhD Symposium at iFM’16 on Formal Methods*. Reykjavik University: Reykjavik University, June 2016.

[7] H. Hao, V. Singh, and W. Du. “On the effectiveness of API-level access control using bytecode rewriting in Android”. In: *ASIA CCS* (2013).

[8] Healthcare Information and Management Systems Society. “Mobile Security Toolkit: Sample Mobile Device User Agreement”. In: *Healthcare Information and Management Systems Society* (2012).

[9] *IBM MaaS360 - Enterprise Mobility Management (EMM)*. URL: http://www-03.ibm.com/security/mobile/maas360.html (visited on 10/12/2016).

[10] G. Kennington et al. *Mobiles Devices Policy*. Tech. rep. Torbay, Southern Devon Health, and Care NHS Trust, Mar. 2014.

[11] J. Lin et al. “Modeling Users Mobile App Privacy Preferences: Restoring Usability in a Sea of Permission Settings”. In: *Symposium On Usable Privacy and Security* (2014).

[12] F. Martinelli, P. Mori, and A. Saracino. “Enhancing Android Permission Through Usage Control: A BYOD Use-case”. In: *Symposium on Applied Computing*. 2016.

[13] MobileIron Security Labs. *Q4 Mobile Security and Risk Review*. Tech. rep. MobileIron Security Labs, Dec. 2015.

[14] Nicholas R. C. Guerin. *Security Policy for the use of handheld devices in corporate environments*. Tech. rep. SANS, May 2008.

[15] Rob Smith et al. *Magic Quadrant for Enterprise Mobility Management Suites*. Tech. rep. G00279887. Gartrer, June 2016. URL: https://www.gartner.com/doc/reprints?id=1-390IMNG&ct=160608&st=sb (visited on 11/23/2016).

[16] H. Schulze. *BYOD & Mobile Security 2016 Spotlight Report*. Tech. rep. LinkedIn Information Security, 2016.

[17] *Secure Android Solution BlackBerry and Android for Work*. URL: http://us.blackberry.com/enterprise/android-for-work.html (visited on 12/13/2016).