EAI Endorsed Transactions
on Cloud Systems
Research Article EAI.EU

How Location-Aware Access Control Affects User Privacy and Security in Cloud Computing Systems

Wen Zeng[1]*, Reem Bashir[2], Trevor Wood[3], Francois Siewe[1], Helge Janicke[4] and Isabel Wagner[1]

[1]School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH U.K.
Email: wen.zeng.wz@gmail.com, FSiewe@dmu.ac.uk, isabel.wagner@dmu.ac.uk

[2]HORIBA MIRA Ltd, Watling Street, Nuneaton Warwickshire CV10 0TU U.K.
Email: reem.bashir.12@gmail.com

[3]Network Midlands Ltd, 446 London Road, Leicester LE2 2PP U.K.
Email: trevor@the-woods.org.uk

[4]Cyber Security Cooperative Research Centre, Perth, Australia.
Email: heljanic@gmail.com

Abstract

The use of cloud computing (CC) is rapidly increasing due to the demand for internet services and communications. The large number of services and data stored in the cloud creates security risks due to the dynamic movement of data, connected devices and users between various cloud environments. In this study, we will develop an innovative prototype for location-aware access control and data privacy for CC systems. We will apply location-aware access control policies to role-based access control of Cloud Foundry, and then analyze the impact on user privacy after implementing these policies. This innovation can be used to address the security risks introduced by inter-cloud use and communication, and will have significant impact in making citizen's personal data more secure.

Received on 03 May 2020; accepted on 07 June 2020; published on 10 June 2020

Keywords: cloud computing, user privacy, Cyber security, information hiding, threats

Copyright © 2020 Wen Zeng *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi:10.4108/eai.13-7-2018.165236

1. Introduction

The importance of cloud computing (CC) is increasing due to the high demand of internet services. Hosting data of organizations on the cloud is cost-effective. A challenge for the adoption of cloud hosted data for any organization is that the resource allocations and organization differ between the various providers of CC services. In addition, most employees and users can access CC systems from various locations. This movement of the users might leak sensitive information to the public, due to insecurities of the accessing network infrastructure, vulnerabilities in the devices used for the access as well as different laws and regulations governing the access of personal information across various jurisdictions. Therefore, it is important to integrate location-aware access control policy within the CC systems, especially when considerations to implement controls for GDPR compliance.

Location-aware access control can be used to implement the principle of least privilege, by allowing access to specific resources only in specific locations [32]. Another benefit of location-aware access control is that it supports separation of duty based on the employee's location [32]. Governments can use location-aware access control to deploy policies, for example, the government can put sanctions for a specific location and prevent the users there from accessing some services.

The advent of ubiquitous cloud computing has raised further concerns, those of privacy related to location and remote access to data over insecure networks or in insecure locations. Much has been written on keeping stored location data secure, most of it from a

*Corresponding author. Email: wen.zeng.wz@gmail.com

legal perspective and concerned with citizens' rights as to what data is stored and their knowledge of what is stored. However, legal positions change (e.g., introduction of GDPR in the UK in May 2018 [21]) and none of this addresses how to keep these data private – only the legality of collecting and sharing it. Therefore, it is necessary to analyze how location-aware access control security policies affect the user privacy.

The contribution of this paper is an innovative prototype for location-aware access control security policies and data privacy for CC systems. This paper is organized as follows: We present related work in Section 2, followed by a discussion of our threat model in Section 3. In Section 4, we discuss the location-aware access control policies for Cloud Foundry and implement these security policies. In Section 5, case studies will be used to evaluate the implementation of security policies. Section 6 will examine the privacy issues that arise from implementing location-aware access control security policy. In Section 7, we will use case studies to describe how privacy protections can be integrated to protect user privacy in location-aware access control. Section 8 will conclude the paper.

2. Related Work

In this section, we will discuss background and related work on access control and data privacy in CC.

2.1. Access Control in Cloud Computing

According to [29], there are three main access control policies models, namely discretionary access control (DAC), mandatory access control (MAC) and role-based access control (RBAC). In DAC, the resource owner is determining the access to his/her resource. In MAC, a central authority is responsible for setting the rules. RBAC differs from the DAC and grant access based on the role of the requester. In this paper, we will focus on RBAC.

The RBAC is the common form of access control to support the polices within classified systems, such as financial organizations where the data of users are classified. In such organization the security administrator will assign the users with roles and assign roles with access permissions [17]. The RBAC polices consists of users (i.e., subjects), objects, operations and roles. The user is a person in the organization. The role is an abstraction of individual users, with the view that a user can act in a defined role. Operations are therefore only assigned to a role, access is then determined by checking whether the role has the required operations and that the user is currently acting in that role. The objects represent the data that will be accessed [17].

The following papers extended the access control for CC to meet the workflow security requirements of the federated cloud computing. [39] presented a solution to store the data of the organizations securely on public cloud by presenting cloud storage architecture based on RBAC roles. [37] introduced a formal model to capture the dynamic workflow in the federated cloud where the entities present in the cloud system can be assigned different security levels belonging to a given security lattice, and each cloud is assigned a security level to set the confidentiality level of the cloud. [34] introduced – opacity – as a security property to analyze the workflow security after implementing security policies in the CC systems. However, none of the above papers considered the location of the users and data privacy when applying access control policy in the cloud.

There are some existing work on role-based access control and attributed-based access control. [35, 36] investigated that access control policies and technologies would affect the productivity of the organizations. [11] implemented location attribute into RBAC by assigning privileges to resources based on the attribute values of the resources, and roles to users based on user attributes. [20] enforced and tested location-aware attribute-based access control for online social networks on a personal computer as a virtual server, the geo-location was used to get the user location. [31] proposed a new programming paradigm called context-aware secure action system where the functional, security and context-awareness requirements of pervasive computing systems can be specified and reasoned about in a uniform manner, based on the key concepts of agent, action, context and security policy. The security policy is specified in the attribute-based access control model and uses the attributes of context agents to control access permissions dynamically as the context of the system changes.

In addition, [4] presented a model for representing and evaluating location-aware access control (LBAC) conditions such as time-dependency and uncertainty of location. Also, the proposed architecture that integrates the represented LBAC evaluation with traditional identity-based access control, which supports a broad variety of location-aware policies. [12] introduced the concept of location-aware access control and discussed the requirements for the location-aware model. [27] suggested using two methods to get the location information, the first method is by using the GPS. Although the GPS data can change within the same build, the GPS covers a vast area with accuracy within a few meters. The second method is by using infra-red sensors, where the location is accurate more than the GPS. However, this approach cannot be used within the cloud because not all the users can have the sensors. In our study, the Fixed IP address and the GPS data will be used to get the user location information.

2.2. Privacy and Security in Cloud Computing

[16] assert that in the collection and correlation of large amounts of data and with the high level of interconnectivity, combining data from multiple sources may improve service quality but it also increases the risk of privacy violations. The paper addressed location privacy and other issues associated with ubiquitous connectivity and cloud computing, but did not address the issue of restricting access to data to specific locations. For example, a company may not want staff to access sensitive documents while away from their office, or to only be able to access client documents while at their office and the office of the client.

As early as 1996, [13] argued that existing user authentication methods based on something the user knows (e.g., username/password or PIN), something the user has (e.g., access token or crypto-card) or something the user is (e.g., biometrics) are not fool proof. They assert that geodetic (geo-location) information would add an additional layer of security by supplementing or complementing other methods of authentication, e.g., only allowing a user to log in to a system from a specific location. They go on to explore further practical uses of geo-location-based authentication. [6] introduced this concept to the construction industry where construction site staff must have on-demand access to site specific data, e.g., plans, drawings, schedules, and budgets. More recently, [18] examined the security aspects of geo-location-based privacy and identity and access management. However, there is still a long way to go to develop applications that use geo-location based security.

Much has been written concerning security issues with CC services. Most of these focus on traditional security threats, e.g., network-based attacks, Virtual Machine (VM) based attacks, storage-based attacks and application-based attacks [23]. [28] address several additional challenges: resource location, multi-tenancy issues, authentication and trust of acquired information, and cloud standards. Here, resource location is concerned with where the information is stored and raises the question of legislative jurisdiction. A fuller list of what is now being called Mobile Cloud Computing (MCC), security and privacy challenges is presented by [24]: data security, virtualization security, partitioning and offloading security, mobile cloud application security, mobile device security, data privacy, location privacy, identity privacy.

Location Privacy concerns the privacy issues surrounding the location of the user. As a user moves from one location to another, a cloud service may need to track where that user is so that it can provide location specific information (Google Maps is probably the best-known example of this type of location usage) or provide location-based security (as this project will be examining in relation to Cloud Foundry). Should this information become available to unauthorized actors, it could be used to help profile the user, for example finding places that the user frequents regularly. If data from multiple users is exposed, this can be used to create cross-connections between users. Are the same two people meeting up regularly, perhaps repeatedly using the same locations or at the same time?

Identity privacy concerns knowing who is using the cloud service. This information is needed by the cloud service to provide the correct and/or customized service to the user. Should an unauthorized actor gain access to this information, the actor could impersonate the user or use the information to apply further refinements to the profile of the user. [22] discussed the issues of location and identity privacy along with proposing a low communication cost k-anonymity algorithm which may solve this issue.

Resource location concerns where the resources are geographically located. Cloud services can, and do, span the world making it difficult or impossible to know where information is being stored. This could present legislative issues, for example in determining which nation's laws covers disputes. In May 2018 the General Data Protection Regulation (GDPR) came into effect in Europe [21]. Among other things, this law restricts geographically where Personally Identifiable Information (PII) is stored. The law imposes restrictions on the transfer and storage of PII outside of the European Union. Cloud services, being global, may store PII anywhere without the person identified by the information giving consent or even knowing.

3. Threat Model

We propose that location-aware access control can mitigate some security threats in CC, in particular threats coming from external attackers. For example, if an attacker compromises a user's login credentials or externally accessible data flows (where data travels from the user over an internet connection to the cloud app), the attacker could pretend to be an authorized user and identify who the user is.

The addition of location-aware access control will mitigate this threat by restricting authorized access to specified geographic locations, so even if the authorized user's login credentials are compromised, an attacker could only log in if the attacker was at an authorized location or able to spoof the attacker's location accordingly. This would require the additional compromise of the authorized locations and the ability to spoof those locations. However, the introduction of location-aware access control may have privacy implications for authorized users. By using location-aware access control, an authorized user will have to

provide his location and an attacker may be able to access this location information, allowing him to build up a profile of the authorized user's location. Successful attacks over a period of time may allow an attacker to build a profile of the authorized user's movements and habits, for example he may visit the same location at the same time every day. Importantly, the attacker in this case may be *internal*, such as the user's employer or another employee.

4. Implementation of Location-aware Access Control in Cloud Foundry

In this section, we will explain the design and implementation of a location-aware access control in Cloud Foundry.

4.1. Cloud Foundry

Cloud Foundry is an open source cloud application platform governed by the Cloud Foundry Foundation [9]. Figure 1 shows the security architecture of Cloud Foundry.

Cloud Foundry use *demilitarized zone (DMZ)* and *virtual LAN (vLAN)* to protect the system security [9]. The components of the Cloud Foundry run within different vLANs on virtual machines, where the public network only get access to the Cloud Foundry through *Load Balancers* [9]. The load balancer communicates only with the Cloud Foundry *Go Routers*, *Outbound NAT* virtual machine (VM) and *Jump Box*. The load balancer minimizes the security vulnerabilities, by limiting the contact point of the public access to the Cloud Foundry system. Using https *BOSH* Operators to deploy software over hundreds of VMs. The BOSH consist of *BOSH Director*, which controls VM creation and deployment, as well as other software and service life cycle events. To increase the security, the communications between VMs only launched over the *Message Bus (NATS)*. NATS is an open source cloud native infrastructure messaging system, which cannot be access from outside the Cloud Foundry.

The public access to the *Cloud Controller* and authentication *UAA* happens over the HTTPS protocol, meanwhile the interaction of the Cloud Foundry components happens over one of the three protocols names, a publish-subscribe *message bus NATS*, *HTTP* and *SSL/TLS* [9]. To identify and manage the users, the *UAA* is an *OAuth2 authorization server*, which issues access tokens for the applications that request platform resource. The OAuth2 is a protocol that allows third-party applications to grant limited access to an HTTP service, either on behalf of a resource owner or by allowing the third-party application to obtain access on its behalf [15]. The UAA owns the user accounts and authentication source, which support standard protocol SAML, LDAP and OpenID [15]. To authenticate every request with the *Service Broker API*, the Cloud Controller rejects any registration without a user name and password. The *Service Broker* is the component of the service that implements the Service Broker API, by advertising a catalog of service offerings and service plans to the marketplace.

The public access to the Cloud Controller in the Cloud Foundry provides REST API endpoints to access the system and maintains a database with tables for Orgs, spaces, services, user roles. Also, the cloud controller manages the deployment of the application when the user pushes the application on the Cloud Foundry [10].

Another security component is the segments isolation that isolates the deployment of the apps' resources to avoid redundant management components and network complexity. Using isolation segments helps to set security policies for different apps, Orgs and spaces. The Org is a "development account that an individual or multiple collaborators can own and use" [8], and the spaces are the shared apps locations within the Orgs. To view and access an Org or the spaces the user has to be the member of the Org and spaces, the Cloud Foundry users RBAC to grant permissions to an Org or spaces based on the user role [9].

4.2. Location-aware Access Control in Cloud Foundry

The Cloud Foundry uses RBAC policy to control the access to resources based on the user role [8]. The user can have different roles within the Org in different spaces, in other words, the user can have more than one role. The user can have read or write scope; reading scope is to view resources, and writing scope is to create, update and delete resources [8]. The type of users in the Cloud Foundry is based on the role, for example, Org auditors, Org billing managers, Org user, developers and space auditors [8].

The location-aware RBAC of the cloud foundry will be based on the existing RBAC rules of the Cloud Foundry. The existing RBAC rules of the Cloud Foundry will be integrated based on the location of the user. The location is divided into private and public, the private has full access (read-write), while the public has only read access. The integration of location-aware access control is on two stages: first getting the user location, then applying the existing RBAC rules for the Orgs and spaces, based on the user location (private or public).

The user location can be used in the Cloud Foundry to grantee the type of access the user can have within the specific location. If the access is not authorized in the RBAC rules then it will not be authorized in the any location as well.

For example, the user can get full access to the Orgs and spaces in the Cloud Foundry from the private

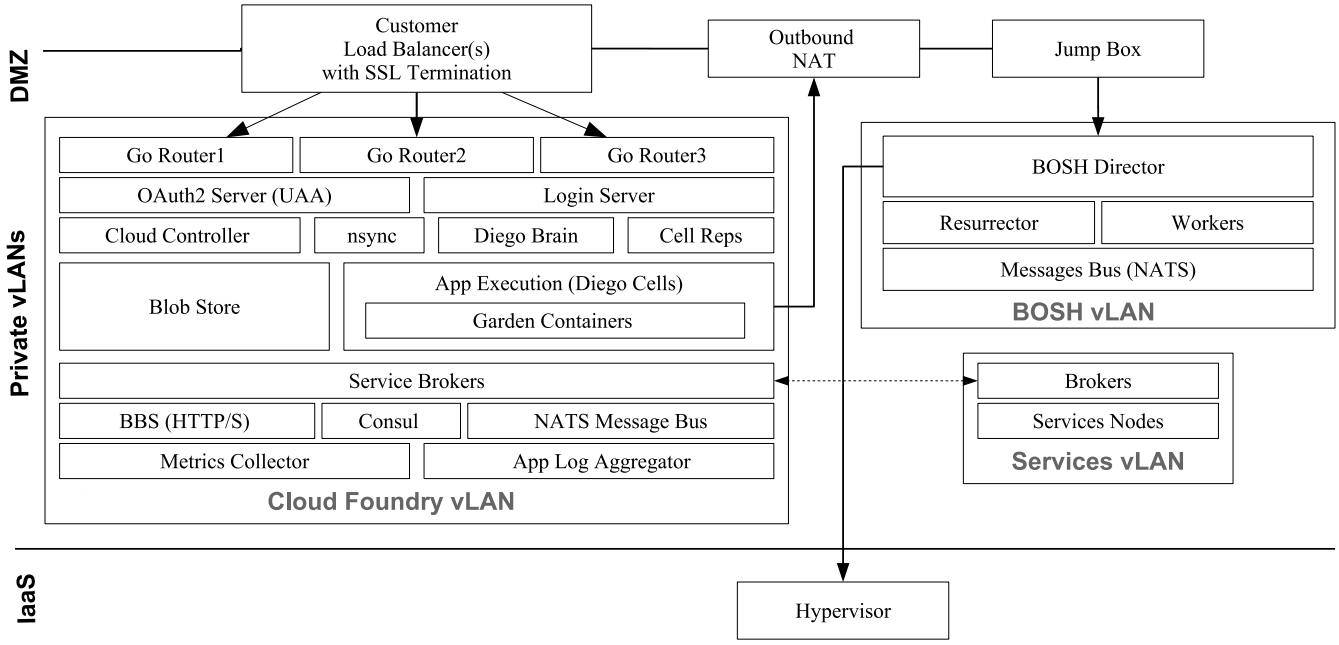


Figure 1. Security architecture of Cloud Foundry

(e.g., office). However, the user can only get read-only access from public places (e.g., train). Applying the location-aware access control can help in achieving accountability, separation of duties, least privilege and usability. In our study, we will use GPS and the static IP addressee to determine the user location, an approximate of three-digit GPS longitude and latitude will be used.

The role access control policy of the Cloud Foundry code is implemented in the Cloud Foundry Controller[1], where the policies implemented using Ruby programming language. The location-aware RBAC policy for active Orgs is listed in Tables 1 and 2, where Table 1 shows the policy for private locations, while Table 2 shows the policy for public locations.

The private locations are saved on the SQL database within the Cloud Controller, the tables are managed by the Admin. The private locations can be IP address or GPS coordinates, each type has a different table on the database. After the user successfully login to the Cloud Foundry, the Cloud Controller check the user location and role. First, the user's IP address is checked if the IP address is not saved on the IP table the user's GPS coordinates will be collected using Geolocation API and checked within the GPS tables. If the location is not found on both tables the location will consider public, to add the location to the private locations on the database the Admin have to do it.

[1] https://github.com/cloudfoundry/cloud_controller_ng

5. Case Studies

To implement location-aware access control, additional information over and above the user's normal access credentials are required. This information is used to describe the location of the device that is being used to access the sensitive data, for example an IP address or the GPS coordinates, i.e., the location metadata.

In this section, we consider three case studies and describe how they would use our implementation of location-aware access control.

5.1. Confidential Documents

In a business organization, documents for a client proposal are being stored in a CloudFoundry org. The user has full read/write access to these documents while in the user's office, read-only access at the client's site, and is not able to access the documents from anywhere else. To implement this policy, the user must be the org manager or a member of the org. The office is saved as a private location and the client's site as public. When the user logs in, our system detects the user's location and makes the access decision based on the user's role and location.

5.2. Health Records

In a healthcare system, health records for a patient are being stored in a CloudFoundry org. Doctors should have full read/write access from locations within the hospital, the patient's GP should have read-only access

Table 1. Location-aware RBAC for Active Orgs when the location is **private**. "√" indicates that access is allowed; "★" indicates the access is not by default; "★★" indicates Admin role does not need to be added as member of Orgs or spaces to view resources; "◇" indicates Org Managers can rename and edit their Orgs, but cannot delete them.

Activity	Admin	Org Manager	Org User	Space Manager	Space Developer
Scope of operation	Org	Org	Org	Space	Space
Assign user roles	√				
View users and roles	√	√	√	√	√
Create, assign Org quota plans	√				
View Org quota plans	√	√	√	√	√
Create Orgs	√	★	★	★	
View all Orgs	√★★				
Edit, rename, delete Orgs	√	√◇			
View spaces	√	√		√	√
Edit spaces	√	√		√	
View the status, number of instances, service bindings, and resource use of applications	√	√		√	√
Add private domains					
Deploy, run, manage apps	√				√
Rename applications	√				√
List application, service usage	√				√

Table 2. Location-aware RBAC for Active Orgs when the location is **public**. "√" indicates that access is allowed.

Activity	Admin	Org Manager	Org User	Space Manager	Space Developer
Scope of operation	Org	Org	Org	Space	Space
Assign user roles					
View users and roles	√	√	√	√	√
Create and assign Org quota plans					
View Org quota plans	√	√	√	√	√
Create Orgs					
View all Orgs	√				
Edit, rename, delete Orgs					
View spaces	√	√		√	√
Edit spaces					
Delete spaces					
Rename spaces					
View the status, number of instances, service bindings, and resource use of applications	√	√		√	√
Add private domains					
Deploy, run, manage apps					
Rename applications					
List application, service usage	√				√

from his surgery, and the patient should have full access to certain information that he can change (address, phone number, etc.) from home. To implement this policy, the doctors are org managers, the patient's GP is a member of the org, while the patient is space manager within the org. For the doctors, locations within the

hospital are saved as private and for the patient, their home is saved as a private location.

5.3. Online Banking

A user has an online banking app, where the data is saved in a CloudFoundry org. The user has full access to his account while at home, with the ability to set up standing orders, direct debits and to make payments. The user has limited access while connected from somewhere else, with the ability to view account statements. To implement this policy, the user's home is saved as a private location, while all other locations are public. The user will be a member of the org.

6. Privacy Analysis of Location-Aware Access Control

This section will examine the privacy issues that arise from implementing location-aware access control security policy.

We will describe the additional information that will be required over and above a user's normal access credentials, and then examine the threats and privacy issues relating to this information using the LINDDUN (Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance) threat modeling methodology, and then we will present generic privacy mitigation strategies and tailor them to our case studies.

We use a generic app hosted on Cloud Foundry to examine threats and privacy issues. Figure 2 shows the data flow diagram for a user's access to such an app. The *User* (Entity *E1*) connects via a *portal* (Process *P1*) to the cloud app which is running in what Cloud Foundry calls a *Diego Cell* (Process *P2*). The app communicates with the *Cloud Foundry Cloud Controller* (Process *P3*). The Cloud Controller gathers the information that the app needs to run from the *CCNG Blob Store* and the *CCDB* (Data Stores *DS1* and *DS2*). The Cloud Controller also communicates via the *Service Broker* (Process *P4*) with the *Service Backend* (Data Store *DS3*). The Service Backend is where the sensitive data associated with the app is stored. This is the data that, ultimately, the user wishes to access and modify. It is also where the data to verify the user is stored (username, password and location metadata). The Cloud Controller CCNG, CCNG Blob Store and CCDB are within the same trust boundary as the CCNG Blob Store and CCDB only contain data required by the Cloud Controller CCNG to run the app.

6.1. Analysis of Privacy Threats

Table 3 shows the threats associated with each element of our data flow diagram. Numbered threats are discussed in detail in the next sub-sections. We do not consider threats marked with X because of the following assumptions. We assume that the two data stores DS1 and DS2 contain only data pertaining to the app that the user is running and no sensitive or personally identifiable information (PII). Equally, the data flows DF4 ($DS1 \rightarrow P3$) and DF6 ($DS2 \rightarrow P3$) only carry data pertaining to the app that the user is running and no sensitive data or PII. The privacy threats to processes are addressed by addressing the threats to data flows and data stores connected to the processes. Finally, we assume that non-repudiation is not an issue. Because we focus on technical threats, we do not consider non-compliance and unawareness threats.

Threats to Data Stores. The data store DS3 Service backend is susceptible to linkability, identifiability and disclosure of information attacks.

A disclosure of information attack against a data store requires the attacker to access the data store, for example by hacking the server that the data store is attached to. He can then inspect the data to locate that which he wishes to disclose.

Linkability and identifiability attacks against a data store require the attacker to enact a disclosure of information attack against the data store. The attacker can then access the user's personally identifiable information (login credentials, location metadata), identify the user and/or build a profile of the user's access to the Cloud Foundry hosted app based on the time and location the app was accessed, and link data entries that are accessible by the user. This could lead to identifying the user via the data he is able to access and/or linking multiple users together as they are able to access the same data.

Threats to Data Flows. The data flows DF1, DF2, DF3, DF5 and DF7 are susceptible to linkability, identifiability and disclosure of information attacks.

A disclosure of information attack against a data flow requires the attacker to intercept the data flow and then examine and understand the data within the data flow.

A linkability attack against a data flow requires the attacker to enact a disclosure of information attack against the data flow on multiple occasions. The attacker could then access the user's personally identifiable information (login credentials, location metadata) and identify the user and/or build a profile of the user's access to the Cloud Foundry hosted app based on the time and location the app was accessed.

An identifiability attack against a data flow requires the attacker to enact a disclosure of information attack against the data flow. Through this information disclosure, the attacker could then access the user's personally identifiable information (login credentials, location metadata) and identify the user and/or build a profile of the user's access to the Cloud Foundry hosted app based on the time and location the app was

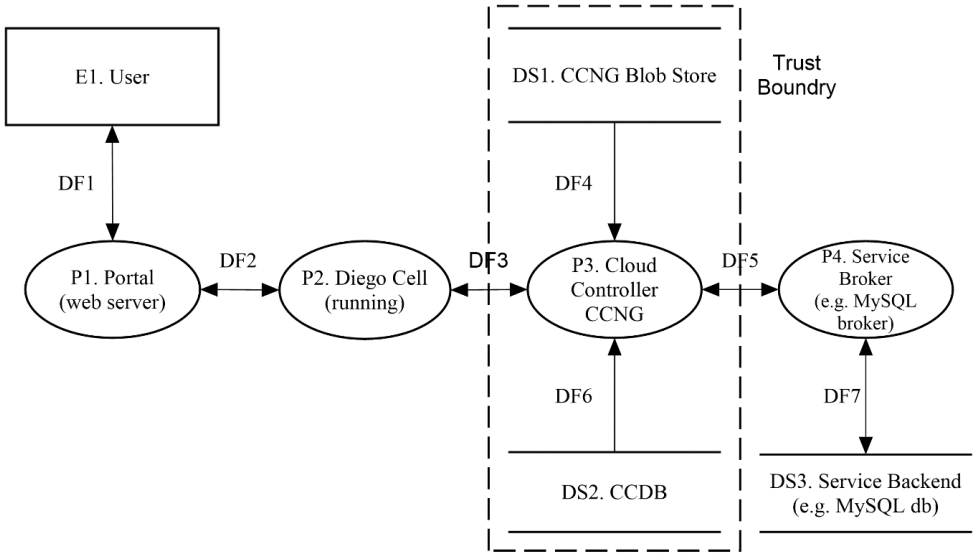


Figure 2. Data Flow Diagram for User

Table 3. LINDDUN privacy mapping for Cloud Foundry hosted app.

		L	I	Nr	D	Di	U	Nc
Data store	DS1. CCNG Blob Store	X	X	X	X	X		X
	DS2. CCDB	X	X	X	X	X		X
	DS3. Service Backend	1	2	X	X	3		X
Data flow	DF1.	4	5	X	X	6		X
	DF2.	7	8	X	X	9		X
	DF3.	10	11	X	X	12		X
	DF4.	X	X	X	X	X		X
	DF5.	13	14	X	X	15		X
	DF6.	X	X	X	X	X		X
	DF7.	16	17	X	X	18		X
Process	P1. Portal	X	X	X	X	X		X
	P2. Diego Cell	X	X	X	X	X		X
	P3. Cloud Controller	X	X	X	X	X		X
	P4. Service Broker	X	X	X	X	X		X
Entity	E1. User	19	20				X	X

accessed. Additionally, the attacker is can perform a spoofing attack against the user.

Threats to Entities. The entity E1 User is susceptible to linkability, identifiability and unawareness attacks.

Identifiability and linkability attacks against an entity require the attacker to enact a disclosure of information attack against a process, data flow or data store. The attacker could then access the user's personally identifiable information (login credentials, location metadata) and identify the user and/or build a profile of the user's access to the Cloud Foundry hosted app based on the time and location the app was accessed. Additionally, the attacker is can perform a spoofing attack against the user.

This analysis reveals multiple occasions where a user's personally identifiable information is at risk of disclosure. In nearly all cases, exposure of the user's login credentials will mean that the associated location metadata is also exposed. Exposure of the location metadata can lead to additional risks to the user, namely the user's location can be determined, either in real time (when data flows or processes are compromised) or historically (when data stores are compromised). Exposure of this information would allow an attacker to build a more detailed profile of the user by including the location (and time, if exposure is in real time) that the user accesses the Cloud Foundry hosted app.

6.2. Privacy Protections for Location-Aware Access Control

The previous section identified ways in which a user's personally identifiable information (PII) could be compromised by using the LINDDUN privacy threat modeling methodology. This section is concerned with the protection of the user's personally identifiable information. It will propose ways in which each threat could be mitigated by applying technological solutions or implementing suitable policies and procedures.

Mitigation Strategies for Data Stores. The assets held by a data store that need protecting are the user's login credentials (user name, password, PIN), the user's internal identifier and the location metadata pertaining to locations from where the user has attempted access (IP address, GPS coordinates). These assets can be protected from identifiability and linkability threats by minimizing exposure of the information, for example by removing, hiding, or generalizing, and by ensuring confidentiality (see Table 4).

Techniques to ensure confidentiality include encrypting data at rest and secure password storage. Removing unnecessary data, such as a long history of user locations, reduces the amount of data that can be exposed. Stored location data can also be obfuscated, for example by hashing IP addresses, and generalized, for example by reducing the accuracy of GPS location data.

Mitigation Strategies for Data Flows. The assets in a data flow that need protection are the user's personally identifiable information (login credentials and location metadata). These assets can be protected against linkability, identifiability, and detectability by removing, hiding, or generalizing transactional data (the data being communicated) and contextual data (the data necessary for communication) on the data flows.

Contextual data such as IP addresses can be removed by using a Virtual Private Network or Onion Routing, e.g. Tor [14], to hide the link between the elements connected by the data flow.

To hide transactional data from external attackers, data flows should use encrypted connections, such as HTTPS (secure HTTP) and SSL/TLS (Secure Sockets Layer/Transport Layer Security).

To hide data from internal attackers, multi-party computation [33, 38] could be used so that user and server jointly compare the user's location against the set of permitted locations, while the server does not learn the user's location, and the user does not learn the set of permitted locations, but both learn the result of the computation. Alternatively, the location metadata could be removed from the data flow by performing the location verification in a trusted way on the user's device, for example by using a zero-knowledge proof [19, 26].

To generalize contextual data, data can be generalized by using anonymous communications. [30] review anonymous communication protocols in a number of different scenarios, including anonymous web browsing and hidden web services, both of which would be useful in this situation. Additionally, user and server can insert dummy traffic [25] to make it harder for external attackers to infer that communication is really taking place.

To generalize transactional data, i.e. to protect the users PII against a curious server, the user could reduce the accuracy of location data reported to the server or inject noise into the location data [3].

Mitigation Strategies for Entities. The entity assets that need protecting are the user's login credentials and location metadata, e.g., geographic location or IP address of the user's device. The assets can be protected against linkability and identifiability threats by protecting the user's ID, e.g. through the use of pseudonyms and technologies that preserve privacy during the authentication process.

For example, private authentication [1, 2] can be used to protect the authentication process against external attackers so that the external attacker does not learn the user's identity. Anonymous credentials [5, 7] can protect the user's identity from internal attackers by allowing anonymous but authenticated usage of the system.

7. Case Studies

In this section, we revisit the case studies from Section 5 and describe how privacy protections can be integrated to protect user privacy in location-aware access control.

In each example the user's location can be determined through the user's IP address or geo-location. Business locations (the user's office, client office, hospital and surgery) would normally have a fixed IP address. The IP address would be queried and compared against a list of valid IP addresses. If the IP address matches any of these, access would be granted.

A user's home or home office is likely to have a dynamic IP address assigned to the Internet connection. In this situation the user's geo-location, for example given as GPS coordinates, would be used to determine his location. The set of valid GPS coordinates could be defined as either within a defined distance of a fixed point or inside a geo-fence defined by a group of GPS coordinates. The user's GPS coordinates would be queried and compared to the valid locations. Again, if it matches any, then the requisite level of access would be granted.

We distinguish three cases where privacy protections are needed: the user's location metadata during authentication, the user's identity during authentication, and the user's data in the service backend.

Table 4. Mitigation strategies for linkability and identifiability of data store 3 (service backend)

Assets	User login credentials (user name/password), internal user ID, location metadata
Confidentiality	Secure password storage (e.g., salted hash)
Remove	Limit amount of stored location data history
Hide	Obfuscate location metadata (e.g., hash IP addresses)
Generalize	Reduce accuracy of GPS location data

Privacy protections for the user's location metadata need to be designed such that the access control policy cannot be violated by an attacker. If location is determined based on a fixed IP address, protection options are limited because IP addresses are transmitted as part of the communication protocols. Importantly, in this case, users cannot use self-defense mechanisms such as Tor to hide their IP address from the server because their access level is based on their real IP address. If location is determined based on geo-location, obfuscation-based methods such as geo-indistinguishability can result in random failures of user access and should thus be avoided in our case studies. Instead, we can use multi-party computation or a zero-knowledge proof on the user's device to avoid transmitting unprotected location data and to avoid that the server learns the user's location. In these cases, the user is free to use Tor, e.g. to ensure that the adversary cannot learn which CloudFoundry service user is accessing.

Privacy protections for the user's identity and login credentials need to protect the data in transit, while still allowing successful user authentication. Anonymous credentials are not useful in our case because our implementation of location-based access control needs to know the user's identity. Instead, private authentication mechanisms may be used. In addition, we need to use standard mechanisms to ensure encryption of data in transit, such as HTTPS.

Privacy protections for user data in the backend should include secure password storage. In addition, the server should limit size of its log files so that it stores only a small amount of the user's location history.

8. Conclusion

In this study, we developed an innovative prototype for location-aware access control and data privacy for CC systems. We applied location-aware access control policies to role-based access control of Cloud Foundry, and then analyzed the impact on user privacy after implementing these security policies. Location-aware access control can improve security in CC, but care needs to be taken to protect the privacy of its users. This paper established that: i. the user's location metadata during authentication; ii. the user's identity during authentication; iii. the user's data in the service backend are key cases where privacy protections need to be considered when implementing location-aware access control for an organization that uses CC. This study can be used to address the security risks introduced by inter-cloud use and communication. In addition, this study can help information security providers to make security investment decisions.

Acknowledgement

This research was supported by De Montfort University HEIF 2017-18.

References

[1] Martın Abadi and Cédric Fournet. Private authentication. *Theoretical Computer Science*, 322:427–476, April 2004.

[2] William Aiello, Steven M. Bellovin, Matt Blaze, Ran Canetti, John Ioannidis, Angelos D. Keromytis, and Omer Reingold. Just fast keying: Key agreement in a hostile internet. *ACM Transactions on Information and System Security (TISSEC)*, 7:242–273, April 2004.

[3] Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential Privacy for Location-based Systems. In *20th ACM Conference on Computer and Communications Security (CCS)*, CCS '13, pages 901–914, Berlin, Germany, 2013. ACM.

[4] Claudio A Ardagna, Marco Cremonini, Ernesto Damiani, Sabrina De Capitani di Vimercati, and Pierangela Samarati. Supporting location-based conditions in access control policies. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 212–222, 2006.

[5] Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. pages 1087–1098. ACM, May 2013.

[6] Amir H. Behzadan, Zeeshan Aziz, Chimay J. Anumba, and Vineet R. Kamat. Ubiquitous location tracking for context-specific information delivery on. *Automation in Construction*, 17:737–748, August 2008.

[7] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. *ACM Transactions on Information and System Security (TISSEC) - Special Issue on Computer and Communications Security*, 15, May 2012.

[8] cloudfoundry. Orgs, Spaces, Roles, and Permissions, May 2018.

[9] cloudfoundry.org. Understanding Cloud Foundry Security | Cloud Foundry Docs, November 2017.

[10] cloudfoundry.org. Cloud Controller | Cloud Foundry Docs, August 2018.

[11] Isabel F Cruz, Rigel Gjomemo, Benjamin Lin, and Mirko Orsini. A location aware role and attribute based access control system. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, page 84. ACM, 2008.

[12] Michael Decker. Requirements for a location-based access control model. In *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, pages 346–349, 2008.

[13] Dorothy E. Denning and Peter F. MacDoran. Location-based authentication: Grounding cyberspace for better security. *Computer Fraud & Security*, pages 12–16, February 1996.

[14] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-generation Onion Router. In *13th USENIX Security Symposium*. USENIX Association, 2004.

[15] Docs.cloudfoundry.org. UAA API Reference, 2018.

[16] David Eckhoff and Isabel Wagner. Privacy in the smart city —- applications, technologies, challenges, and solutions. *IEEE Communications Surveys & Tutorials*, 20:489–516, 2018.

[17] D Ferrialo, A Cugini, and R Kuhn. Role based access control: Features and motivation. In *Computer Security Application Conference*, 1995.

[18] Vandana T. Goikar, Supriya K. Jagdale, Priya B. Parade, and Sumedha D. Pawar. Improve security of data access in cloud computing using location. *International Journal of Computer Science and Mobile Computing2015*, 4:331–340, February 2015.

[19] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, 18(1):186–208, February 1989.

[20] Andy Chunliang Hsu and Indrakshi Ray. Specification and enforcement of location-aware attribute-based access control for online social networks. In *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*, pages 25–34, 2016.

[21] Information Commissioner's Office. Guide to the general data protection regulation (gdpr), 2018.

[22] Hiba Jadallah and Zaher Al Aghbari. Aman: Spatial cloaking for privacy-aware location-based queries in the cloud. In *Proceedings of the International Conference on Internet of things and Cloud Computing*. ACM Digital Library, 2016.

[23] Ainhaj Ahmad Khan. A survey of security issues for cloud computing. *Journal of Network and Computer Applications*, 71:11–29, 2016.

[24] Muhammad Baqer Mollah, Abul Kalam Azad, and Athanasios Vasilakos. Security and privacy challenges in mobile cloud computing: Survey and way ahead. *Journal of Network and Computer Applications*, 84:38–54, April 2017.

[25] Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Do Dummies Pay Off? Limits of Dummy Traffic Protection in Anonymous Communications. In Emiliano De Cristofaro and Steven J. Murdoch, editors, *Privacy Enhancing Technologies*, number 8555 in Lecture Notes in Computer Science, pages 204–223. Springer International Publishing, January 2014.

[26] Jean-Jacques Quisquater, Myriam Quisquater, Muriel Quisquater, Michaël Quisquater, Louis Guillou, Marie Annick Guillou, Gaïd Guillou, Anna Guillou, Gwenolé Guillou, and Soazig Guillou. *How to Explain Zero-Knowledge Protocols to Your Children*. Springer, April 1989.

[27] Indrakshi Ray and Mahendra Kumar. Towards a location-based mandatory access control model. *Computers & Security*, 25(1):36–44, 2006.

[28] Chunming Rong, Son T. Nguyen, and Martin Gilje Jaatun. Beyond lightning: A survey on security challenges in cloud computing. *Computers & Electrical Engineering*, 39:47–54, January 2013.

[29] Pierangela Samarati and Sabrina Capitani de Vimercati. Access control: Policies, models, and mechanisms. In *International School on Foundations of Security Analysis and Design*, pages 137–196, 2000.

[30] Fatemeh Shirazi, Milivoj Simeonovski, Muhammad Rizwan Asghar, Michael Backes, and Claudia Diaz. A survey on routing in anonymous communication protocols. *ACM Computing Surveys*, 51, July 2018.

[31] François Siewe. Towards the modelling of secure pervasive computing systems: A paradigm of context-aware secure action system. *Journal of Parallel and Distributed Computing*, 87:121–144, January 2016.

[32] André Van Cleeff, Wolter Pieters, and Roel Wieringa. Benefits of location-based access control: A literature study. In *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pages 739–746, 2010.

[33] Andrew C. Yao. Protocols for secure computations. pages 160–164. IRRR, April 1982.

[34] W. Zeng and M. Koutny. Quantitative analysis of opacity in cloud computing systems. *IEEE Transactions on Cloud Computing*, pages 1–1, 2019.

[35] Wen Zeng. A methodology for cost-benefit analysis of information security technologies. *Concurrency and Computation: Practice and Experience*, 31(7), 2019.

[36] Wen Zeng and Maciej Koutny. Modelling and analysis of corporate efficiency and productivity loss associated with enterprise information security technologies. *Journal of Information Secururity and Application*, 49, 2019.

[37] Wen Zeng, Maciej Koutny, Paul Watson, and Vasileios Germanos. Formal verification of secure information flow in cloud computing. *Journal of Information Security and Applications*, 27-28:103–116, April 2016.

[38] Chuan Zhao, Shengnan Zhao, Minghao Zhao, Zhenxiang Chen, Chong-Zhi Gao, Hongwei Li, and Yu-an Tan. Secure multi-party computation: Theory, practice and applications. *Information Sciences*, 476:357–372, May 2019.

[39] Lan Zhou, Vijay Varadharajan, and Michael Hitchens. Achieving secure role-based access control on encrypted data in cloud storage. *IEEE transactions on information forensics and security*, 8(12):1947–1960, 2013.