An Identity Management Survey on Cloud Computing

Ardi BENUSI

Faculty of Natural Sciences, University of Tirana, Albania

Abstract

Identity Management has become a very important topic in the cloud computing environments, where cloud providers need to control usernames, passwords and other information used to identify, authenticate and authorize users for many different hosted applications. All the vulnerabilities seen on non-cloud solutions are now seen in the cloud, but other flavors and issues are introduced. One would be the ability to manage identities of users when sending data to the cloud. Second would be the identity management of users receiving data from the cloud. And third would be identity management when data is moved from cloud to cloud. In this paper some of the known identity management systems are investigated together with some known cloud providers and the effort to standardize their proprietary identity solutions using some of the relevant technologies like Security Assertion Markup Language (SAML), Simple Cloud Solution Management (SICM) etc.

Keywords: cloud computing, identity management (IDM), personally identifiable information (PPI), authentication, authorization, single sign on (SSO), federated identity (FID), Software as a Service (SaaS), identity and access management (IAM).

1 Introduction

With the cloud invention, information stored locally on a computer can be stored in the cloud, including spreadsheets, presentations, audio, photos, videos, financial statements etc. A cloud user provides sensitive personal information
(e.g., name, phone number, credit card number, driver's license number, date of birth, etc.) while requesting services from the cloud. Managing the amount of information disclosed to any one party is something that has become increasingly important in terms of personal security. The PII left can be used to identify, contact, or locate a particular user, may be exploited and abused [12].

Handing sensitive data to cloud service provider (SP) is a serious concern. Cloud computing can increase the risks of security breaches. Knowing who has user's personal data, how they are being accessed, and the ability to maintain control over them prevents privacy breaches of PII, and can minimize the risk of identity theft and fraud.

IDM is the key to cloud privacy and security but IDM in cloud is more complex than in traditional web-based systems since the users hold multiple accounts with different SPs or with a single SP [1]. In this paper different IDMs are analyzed together with various implementations from known cloud providers like Google, Salesforce, Amazon, Microsoft etc.

2 Digital identities solutions in the Cloud

An identity is a set of unique characteristics of an entity: an individual, a subject, or an object. An identity used for identification purposes is called an identifier [14].

Just as in the real world, different digital identities are used in different contexts. It's common, for instance, to associate different information with each identity. An identity that is used in Amazon might allow access to credit card numbers, while one used with MySpace.com does not. The rules for getting each identity are also different. Getting a digital identity at Amazon is easy: just make up a username and password. Getting a digital identity at work is probably somewhat more difficult, since, at a minimum, it requires the approval of the administrators who run company's network. Entities may have multiple digital identities. An Identity Management System (IDM) supports the management of these multiple digital identities. It also decides how to best disclose PII to obtain a particular service [5]. Despite their diversity, digital identities all have one important thing in common: when transmitted on the network, every digital identity is represented by some kind of security token. A security token is just a set of bytes that expresses information about a digital identity. A simple security token might include only a claim containing a username, while a more complex one might include claims containing a user's first name, last name, home address, and more. Security tokens for some digital identities might also include claims that contain sensitive information such as credit card numbers. Some information is provided in order to prove that these claims really do belong to the user who's presenting them. One simple way to do this is to send a password along with the claims. A more powerful approach is to digitally sign all or part of the claims by using a private key, and then provide the corresponding public key, perhaps wrapped in a certificate. The security tokens that represent digital identities must provide some
kind of proof that allows a receiver of the token to verify that this token really does represent the person or organization with that identity [9].

The interaction of users/entities with IDMs and SPs is shown in figure 1:

1) Identity provider (IdP). It issues digital identities.
2) Service provider (SP). It provides services to user/entities that have required identities.
3) User/Entity. Have to claim who they are prior of accessing services in the cloud.
4) Identity management. A third trusted party used to manage digital identities.

Figure 1 Identity Management System

Based on this general idea of a third party to trust when validating digital identities, many IDM solutions have come to play and some are discussed next.

A. Windows CardSpace

When a CardSpace-enabled application or website wishes to authenticate a user, it requests a particular set of claims from the user. The user selects an InfoCard to use among the ones visually presented to him, and the CardSpace software contacts an IdP to obtain a digitally signed XML security token that contains the requested information, which is communicated to the requesting application.

A user might rely on an application that supports CardSpace, such as a Web browser, to access any of several relying parties. It might also be able to choose from a group of identity providers as the source of the digital identity presenting those relying parties. Whatever choice is made by the user, the basic exchange among these parties has three steps:

First, the application gets the security token requirements of the relying party (RP) that the user wishes to access.

This information is contained in the RP's policy, and it includes things such as what security token formats the relying party will accept, and exactly what claims those tokens must contain.

Second, once it has the details of the security token this RP requires that the application passes this information to CardSpace, asking it to request a token from an appropriate IdP.

And third, once this security token has been received, CardSpace gives it to the application, which passes it on to the RP.
RP can then use this token to authenticate the user or for some other purpose [6, 13].

Figure 2 Interaction between user, IdPs and RP in CardSpace IDM

A drawback in CardSpace is that the user still needs to rely on an IdP providing that certificate to the RP, thus the user needs to trust the IdP. In case of a compromised IdP the digital identity is compromised too. Another drawback is the judgment of the user in trusting the RP certificate and sometimes, in the CardSpace framework RPs with no certificates at all are used.

B. PRIME – Privacy and Identity Management for Europe

PRIME’s approach uses “private credentials” which enable proving one’s authorization (e.g., to be over 18 years old) without revealing information that may identify the individual. These private credentials are derived from certificates issued on different pseudonyms of the same person. Multiple private credentials can be created from a single certificate that are neither linkable to each other nor to the issuance interaction in which the master certificate was obtained. Private credentials provide accountability while protecting the anonymity of the user as long as there is no misuse – in this case the user’s anonymity can be revoked. The user-side component uses protocols for getting IdP endorsements for claims to RPs. Anonymous credentials are provided using an identity mixer protocol (based on the selective disclosure protocol) that allows users to selectively reveal any of their attributes in credentials obtained from IdP, without revealing any of their information. The credentials are then digitally signed using a public key infrastructure. Users in PRIME are used to intuitively decide what to tell whom according to the specific situation. As an example data required in professional life are different from what is needed in private life, and in a book store other data are relevant than in the sports club. Nobody gets to know the complete identity of a person, instead only specific partial identities can be perceived. The same drawback as in CardSpace is the trust in a third party (TTP) IdP [9].
C. Open ID

With Open ID multiple digital identities are controlled with a single username and password called OpenID. Receiving a user name and password from a TTP OpenID, the user interacts with an RP providing only OpenID credentials. Upon being discovered by the RP, the OpenID provider authenticates by prompting a password and asks the user whether the RP should be trusted to receive the necessary identity details for the service. After accepting, the user is redirected to the RP along with the credentials, which need to be confirmed by the RP to provide service. Verifying OpenID account, authentication is considered successful, and the user is considered logged in to the RP under the identity specified by the given OpenID [7]. As in many IDMs, phishing in OpenID is of great concern. Even if passwords may be not transmitted with the security token, an attacker may trick the user into accessing the phisher's site and that site might accept any security token the user provided asking for information such as a credit card number. The phisher wouldn't learn the user's password from the faked site, but he/she might learn other useful things. [11].

D. Entity Centric Approaches

The traditional application-centric IDM model requires that each application keeps trace of entities that use it like for example, a tree of users together with a set of policies for user provisioning/de-provisioning. In cloud computing, entities may have multiple accounts associated with different SPs. Also, entities may use multiple services offered by the same SP. A cloud user has to provide his/her PII, which identifies him/her while requesting services from the cloud. Sharing PIIs of the same entity across services can lead to mapping of PIIs to the entity. The main issue is how to secure PII from being used by unauthorized parties in order to prevent serious crimes against privacy, such as identity theft. A story told in the PRIME movie, illustrates crucial privacy issues on the Internet and shows two simple rules explaining how people can protect themselves in the online world:
1. Separating contexts so that observers cannot accumulate sensitive data.
2. Being cautious when personal data are requested and keeping track of information disclosure [8].

An approach proposed in [1] eliminates the need for a TTP IdP and the need for a trusted host that holds the PPI. In this approach adaptation of Fiat and Shamir Zero Knowledge Proofing protocol, together with Active Bundle (AB) are used to anonymously authenticate entities to SPs.

3 Managing digital identities on cloud SP

SAML (Security Assertion Markup Language)

SAML is an open standard protocol used to exchange authentication and authorization data between two different security domains. SAML is a secure
based XML communication mechanism that shares identities between multiple organizations and applications. It has the ability to eliminate most passwords in the cloud and enable SSO, a desired feature since it does not require passwords or logins to each application. Instead of password, application that use SAML, accepts secure tokens which only reveal what is needed to gain access to applications. Since no password exists there is nothing for customers, partners or employees to forget loose or have their passwords stolen. A users signs into his/her company network with corporate credentials and when clicks a link to access applications or secure content at the SP, the IdP generates a secure token to be sent to SP. The token grants accesses to applications and content, but does not pass any information that can be used by anyone else to access them [10].

**SSO Service for Google Apps**

IDM in Google is an entity-centric model that provides partner companies with full control over the authorization and authentication of hosted user accounts that can access web-based Google applications. The SAML model is used making Google partners act as IdPs controlling usernames, passwords and other information used to identify, authenticate and authorize users for web applications hosted on Google. Here is how the SAML model works on Google [4]:

1. The user attempts to reach a hosted Google application or service.
2. Google generates a SAML authentication request. The SAML request is encoded and embedded into the URL for the partner's SSO service.
3. Google sends a redirect to the user's browser. The redirect URL includes the encoded SAML authentication request that should be submitted to the partner's SSO service.
4. The partner decodes the SAML request and extracts the URL for both Google's ACS (Assertion Consumer Service) and the user's destination URL. The partner then authenticates the user. Partners could authenticate users by either asking for valid login credentials or by checking for valid session cookies.
5. The partner generates a SAML response that contains the authenticated user's username. This response is digitally signed with the partner's public and private DSA/RSA keys.
6. The partner encodes the SAML response and returns that information to the user's browser. The partner provides a mechanism so that the browser can forward that information to Google's ACS. For example, the partner might embed the SAML response and destination URL in a form and provide a button that the user can click to submit the form to Google.
7. Google's ACS verifies the SAML response using the partner's public key. If the response is successfully verified, ACS redirects the user to the destination URL.
8. The user has been redirected to the destination URL and is logged in to Google Apps.
Force.com User Management and Sign-on

Two primary mechanisms control user access to resources on the Force.com platform: profiles and sharing rules. A profile is a template containing a collection of pre-defined settings that determine what a user can see and do within the platform. Profiles set the basic rules for whether a given user can see and use each application as well as each tab within the application. By default, web access to Force.com is granted by requiring users to provide a username and password that match values stored within Salesforce database. Users are directed to a single form-based sign-in page to enter their credentials. Once users sign in, they can access any Force.com property that is authorized by their profile, including their own applications, Salesforce CRM, Portals, etc. without re-authenticating. When users interact with software that is not running directly on the Force.com platform, the software creates a session on behalf of the user, and uses that session to communicate with the platform. In order to do this, the software usually prompts the user for their login credentials and stores those credentials. The platform also requires that any external application connecting to the platform sends a security token on behalf of the user. In order to control the risk of password compromise, the Force.com platform provides two alternatives for user sign-on [2].

1. Delegated authentication where an organization sets up a client web service called a delegated authority (DA) that replaces the mechanism for all sign-in. Users must first authenticate to their enterprise and the enterprise must then create a session to Salesforce by sending in the URL the username and a token to Salesforce for validation by the DA. Once this has occurred, the user may travel between cloud and the company without re-authentication.

2. Federated Authentication. The Force.com platform is able to natively validate SAML assertions and create a session for the users when appropriate. Compared to delegated authentication, which requires the organization to host a service which makes proprietary API calls, SAML can securely send PPI without proprietary coding. In Federated Authentication the company must implement or trust a SAML IdP before using cloud service.

Federation to Amazon Web Services (AWS)

AWS added support for SAML. This feature enables federated SSO which lets users sign in to AWS console or make programmatic calls to AWS API-s by using assertion from a SAML compliance IdP like Active Directory Federation Services (ADFS). By using AWS IAM role, instead of creating AWS accounts for all corporate users, AWS roles may be created and mapped to roles in Active Directory. The AWS IAM role may have various policies that control access to AWS resources like for example EC2 or S3 and these policies may be mapped to an IdP that the company trusts. The authentication process is then invisible to user since SSO is implemented together with IAM role [3].
4 Conclusions

Cloud identity management may be viewed as having three different aspects. One would be IDM to the cloud – being able to send something from the enterprise to the cloud. The second would be IDM from the cloud – being able to send something that exists somewhere else back to the enterprise. And the third would be identity management within the cloud to cloud. Regarding IDM, cloud does add its own set of issues. Almost every cloud SP has its own proprietary identification management system. The identification component has been an afterthought and as such carries a lot of vulnerabilities. There is no clear answer that if IDM is rights in one context, it will be right in another. The term identity and access management has come to play to describe the challenge and many SaaS providers have begun reinventing IAM functions, for instance applying APIs for federation and authentication to achieve something resembling single sign on. The other trend is the emerging market of IAMaaS or IAM as a service, where SPs offer core IAM function to the cloud or from the cloud. On the authentication side, Security Assertion Markup Language (SAML) has been the major winner in terms of federation because it provides for single sign on capability. OpenID Connect (based on the OAuth 2.0 protocol) is also emerging and could be useful. OpenID Connect is a suite of lightweight specifications that provide a framework for identity interactions via APIs. Simple Cloud Identity Management (SCIM) is a protocol that will build upon authentication, authorization, and privacy models with the goal of reducing the cost and complexity of user management operations.

References


Received: May 1, 2014