Athens Institute for Education and Research
ATINER

ATINER's Conference Paper Series
COM2016-1978

Identity Management and Access Control Techniques and Mechanisms for Managing Identity Management and Access Control

Sotiris Skevoulis
Professor
Pace University
USA

Constantine Coutras
Professor
Montclair State University
USA

Parag Patel
Bank of New York Mellon
USA

An Introduction to ATINER's Conference Paper Series

ATINER started to publish this conference papers series in 2012. It includes only the papers submitted for publication after they were presented at one of the conferences organized by our Institute every year. This paper has been peer reviewed by at least two academic members of ATINER.

Dr. Gregory T. Papanikos
President
Athens Institute for Education and Research

This paper should be cited as follows:

Skevoulis, S., Coutras, C. and Patel, P. (2016). **"Identity Management and Access Control Techniques and Mechanisms for Managing Identity Management and Access Control",** Athens: ATINER'S Conference Paper Series, No: **COM2016-1978.**

Athens Institute for Education and Research
8 Valaoritou Street, Kolonaki, 10671 Athens, Greece
Tel: + 30 210 3634210 Fax: + 30 210 3634209 Email: info@atiner.gr URL: www.atiner.gr
URL Conference Papers Series: www.atiner.gr/papers.htm

ISSN: 2241-2891
14/09/2016

Identity Management and Access Control Techniques and Mechanisms for Managing Identity Management and Access Control

Sotiris Skevoulis

Constantine Coutras

Parag Patel

Abstract

In today's world businesses transactions are increasingly performed online. Transactions include business to consumer transactions as well as business to business transactions also known as B2B in popular vernacular. Pervasive computing nowadays is ubiquitous and this paper will examine options that exist in this arena as well. It explores the concepts of federated entities and describes models of controlling the access to valuable resources. Each model is presented and compared with similar models and approaches. A major issue in controlling access is authentication. The paper takes a detailed look at how secure are the contemporary authentication techniques and mechanisms. It also describes what could be the future in authentication mechanism including biometrics. We are definitely in need of an evolution in this arena. As people start to do more and more online, they are becoming more and more vulnerable. Failing to protect people's identities and valuable data could have a knock on effect in the ecommerce space, which is a multi- billion dollar industry and thus could have huge economic implications.

Keywords: Access Control, Identity Management.

Introduction

One may ask what Identity Management is and why is it important? Identity management is a term that describes the effort to manage the identities of entities that access applications, services or resources. Poor identity management can and does lead to information security breaches that result in significant financial losses and arguably more important reputational damage. Identity is the key to accessing business services, resources and applications. In addition there are regulatory standards that must be complied with such as Sarbanes-Oxley. With the proliferation of on line services, there has come a proliferation of identities. Business to business interactions require different user identities to sign on to different systems in order to participate in transactions. Managing different user accounts and passwords among business partnerships becomes very complex especially when new identities need to be added, old ones removed or existing identities need to be managed for password changes etc. Business partners would need to agree on how to provision new user accounts in each system, with each system having its own security requirements.

Businesses have developed custom implementations and workflows for dealing with the problem of user authorization and authentication. Hence, users are required to create and maintain multiple identities to the many systems that they interact with. What is needed is a single identity that would work across multiple systems. This would enable flexibility and collaboration between interdependent systems and applications. This kind of security integration would only be achievable if there existed some a standard that would be used by all participants.

This is where the role identity management systems come into play. This paper will examine the current industry standards and solutions and find out how they work. It will look at whether the existing solutions are sufficient and what, if any, shortcomings they may have. It will examine the most common frameworks that are being used and ask the question, what other options exist? Finally, it will look at emerging identity management tools and take a look at what may be on the horizon.

Federated Identities

What are Federated Identities

Let us take a look at the concept of network identities and federated identities, as these concepts are key to understand the currently used principles of identity management [3, 6]. A network identity refers to a software solution that encompasses a set of network centric processes to manage the life cycle of entities as well as the relationship between these entities and business applications and resources. The key point is that these practices, in addition to performing authentication and authorization manage the life cycle of these entities and implementation of business processes that support the life cycle.

A federated identity refers to the use of network identities across applications across different domains, businesses and applications. In essence, a federated identity extends the use of network identities across domains. An example of a federated identity is single sign on. The concept of single sign on is that it enables a user to authenticate once, and then access multiple remote applications without having to re-authenticate. Federation lets you share digital IDs with trusted partners. It's an authentication-sharing mechanism designed to allow users to employ the same user name, password or other ID to gain access to more than one network. It's what is known as a "single sign-on." A single sign-on standard lets people who verify their identity on one network or website carry over that authenticated status when moving to another. The model works only among cooperating organizations-known as trusted partners-that essentially vouch for each other's users.

The federated model relies on the security assertion markup language specification, better known as SAML. This open specification defines an XML framework for exchanging security assertions among security authorities. SAML was developed by the Liberty Alliance, an organization formed to establish guidelines and best practices for federated ID management. The Sun Microsystems-backed group developed SAML to achieve interoperability across different vendor platforms that provide authentication and authorization services.

Identity management then, is the process of managing both network and federated identities and encompasses the following:

- User Provisioning; this is the process of creating and administering identities that can access various business applications and resources.
- Roles and Groups; this is the process of mapping user entities to specific roles and groups that have permissions to access specific resources. This greatly eases management of access rights to specific users.
- Account Service Provisioning; this is the process of how account services are provisioned in different systems.
- Delegated Administration; this allows an administrator to create and update a hierarchy of user identities and roles that grant access to applications and resources. The hierarchy allows an organization to delegate administration of application specific functions to different roles that are from sub organizations within a network.
- Audit Trails and Reporting; this allows for the tracking of historical changes as well as the monitoring of any suspicious activity.
- Single Sign On; This allows for the sharing of user identities across domains. This provides a lower cost of interoperability and enhances the user experience. If a system invalidates an identity or session, a global logout can automatically invalidate or sign out from the rest of the sessions.

Security threats and identity fraud have become common and will only continue to grow as more and more business is conducted electronically. Identity management is becoming more important to application security. Having a robust identity management system can decrease administrative costs because of the ability to auto provision, enhance user productivity via streamline authentication processes and deliver a strong and consistent security model that consists of a central standards based authentication point and a centrally managed credential management system.

Security Assertion Markup Language also known as SAML in the vernacular is an XML based frame work for exchange security assertion information about subjects. A subject is an entity who has identity related information specific to a security domain. SAML provides a mechanism to deliver standards based infrastructure for enabling single sign on without any dependency or use of a specific security architecture implantation. SAML provides a framework however it does not provide the underlying security authentication mechanism.

Because SAML enforces a standards based mechanism for achieving single sign on and its design is abstracted out from the underlying application and platform, it is able to be used among heterogeneous applications and platforms. Before the SAML standard existed customized and proprietary systems were developed to enforce centralized security. Apart from the fact that this caused interoperability issues among different vendors and companies it also was not cost effective. For example, one type of proprietary approach was to encrypt the user credentials in the HTTP-POST header and pass it to different applications via a secure transport mechanism such as SSL. The target applications would then decrypt the credentials and use them to authenticate the identity.

Access Control Models

So what exactly is Access Control? [4] At its simplest, it is the process of protecting resources from being accessed by unauthorized entities. So the first step in defining an access control system for any platform is to identify the resources that are being protected, the entities that will be requesting access to the resources and the activities that can be executed on the resources that have to be controlled. These three things, the resources, the entities accessing the resources and the actions that these entities can access exist on different platforms and applications. Access control models can be grouped into three main categories: Discretionary Access Control (DAC) [8], Mandatory Access Control (MAC) [9], and Role Based Access Control (RBAC)[1].

Discretionary Access Control

Discretionary Access Control (DAC) is a policy based on the identity of the requester and what permissions the requestor has or does not have.

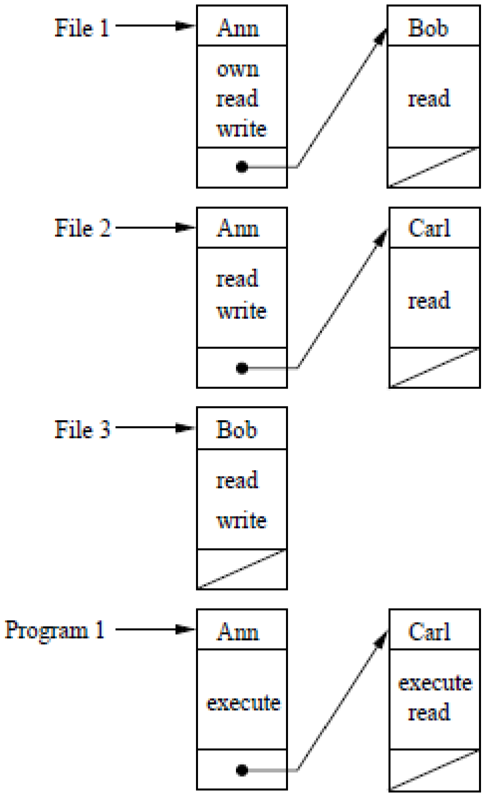
Discretionary policies are called discretionary because users in the system can grant and revoke permissions to resources in a system based on their individual discretion rather than following an administrative policy.

Discretionary policies are typically implemented using some sort of a matrix, usually called an access matrix. At its most basic, an Access Matrix is a grid that lists entities on the left (the rows) and resources on the along the top (the columns). The appropriate places in the grid are then populated with some sort of an indicator to indicate that the entity has access to this resource. However in the real world this sort of an implementation is too inefficient and not optimized to efficient processing. One of the problems is that most of the places in the grid are left unoccupied and therefore wastes memory and resources. There are three ways to implement this matrix in an efficient way:

- An Access Control List
- Authorization Table
- Capability

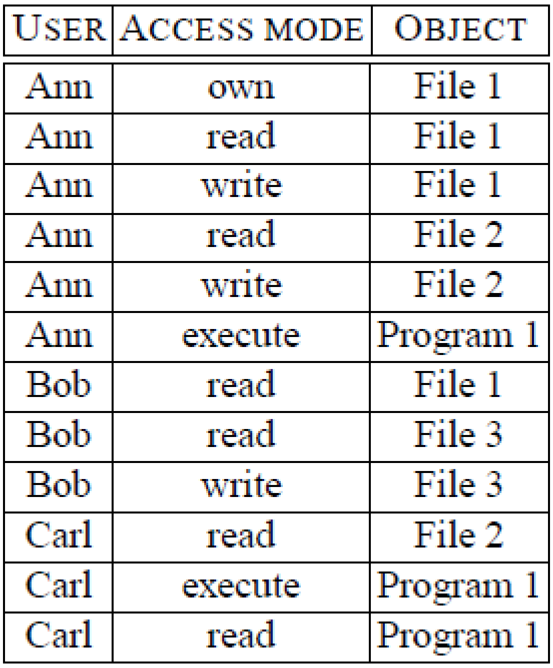
An access control list or ACL is a mechanism whereby the matrix is stored by column. Each resource is linked to a list that defines the actions that each entity can perform on the resource. The diagram below (Figure 1) is a visual representation of the access control model. Note that each resource or object is mapped to a list that defines the access permissions for it.

Figure 1. *Visual Representation of Access Control Model*



In an authorization table model, populated entries in the matrix are placed in a table with three columns which correspond to the resources, entities and objects. Each row in the table maps to an authorization. This approach is generally used by databases, where authorizations are stored as catalogs within the database. Table 1 below illustrates what an authorization table might look like:

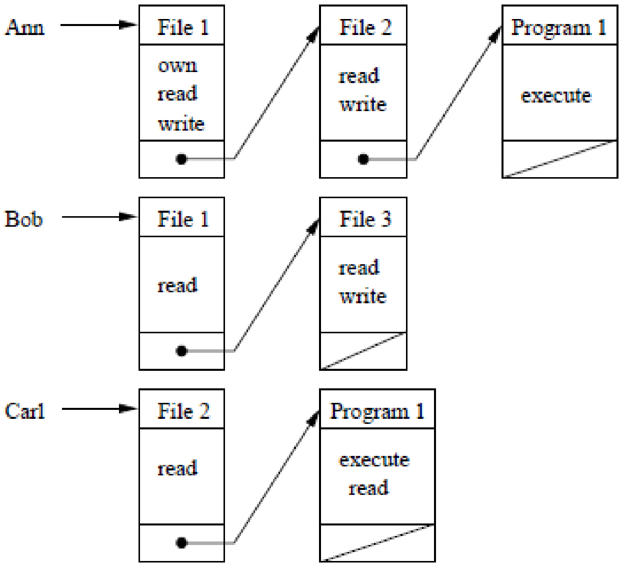
Table 1. *An Example of an Authorization Table*



USER	ACCESS MODE	OBJECT
Ann	own	File 1
Ann	read	File 1
Ann	write	File 1
Ann	read	File 2
Ann	write	File 2
Ann	execute	Program 1
Bob	read	File 1
Bob	read	File 3
Bob	write	File 3
Carl	read	File 2
Carl	execute	Program 1
Carl	read	Program 1

Capability is a mechanism whereby the matrix is stored by row. Each entity has an associated list that defines what their capabilities are with respect to each resource. This is the origin of the name, capability list. The diagram is a visual depiction of a capability based system. Notice that each user has an associated list that defines what they have access to Figure 2.

Figure 2. *Capability Based System*



Access Control Lists and Capability Lists are extremely efficient. An access control list can immediately check authorizations on a resource. However retrieving the authorizations of an entity requires the ACL to be examined for all the resources. Conversely, a Capability is able to immediately retrieve the authorizations of an entity. However retrieving the authorizations for a resource requires the examination of the entire capability list. Unfortunately, capabilities are vulnerable to forgery, meaning that they can be copied and reused. Most modern operating systems make use of ACLs.

One of the main disadvantages of the discretionary policy is that it vulnerable to Trojan Horse attacks. A Trojan Horse is a program that contain hidden functions that exploit the authorizations of the host process. Viruses are generally transmitted as Trojan horses.

The integrity of the discretionary model is able to be violated because once an authorized user has legitimately connected to a system, users originate processes that execute on their behalf and submit requests to the system. Discretionary models ignore this difference and evaluate all requests as if they were submitted by the host process. This can happen without the awareness of the data owner and provides a false sense of security as each and every request passing through an access control list. To illustrate the workings of a Trojan Horse attack, consider the following scenario:

A Manager has a confidential document and an employee wants access to that document. The employee creates a secret file and gives his manager access to write to the file. The manager has no knowledge about the existence of this file. In addition, the employee modifies an application that is used by the manager. The operations he modifies are to: read from the confidential file and to write to the secret file.

Once the manager logs onto the application and starts using it, unbeknownst to the manager the application uses his authorizations to read from the confidential file and write to the secret file. The contents of the confidential file have now been copied to the secret file, to which the employee has access and so the information has been compromised.

Mandatory Policies

Mandatory policies work by enforcing access control on the basis of a centralized authority and policy. A common form of mandatory policy is the multi-level security which forms its basis on the classifications of subjects and objects in a system. Objects are resources that store information. Subjects are entities that request access to the objects or resources. Mandatory policy distinguishes between users and subjects. It defines users as human beings who can access the system while subjects are programs or processes operating on behalf of them. This allows the policy to avoid the leakage of information caused by execution of processes.

An access class is assigned to each subject and object. The access class is one element of a set of partially ordered classes. The partial order is defined by the dominance. Most commonly an access class is defined as consisting of two components: a security level and a set of categories. The security level is an element of a hierarchically ordered set, such as Top Secret (TS), Secret (S), Confidential (C), and Unclassified (U), where TS > S > C > U. [8, 9] The set of categories reflect functions or areas of competence. Access classes along with their set of categories form a lattice. For example:

A subject is usually a process or thread and an object is a resource that the subject wants to access. This resource could be a file, a directory, a port or some kind of IO device. Both subjects and objects have a set of security attributes. When a subject attempts to access an object, the kernel of the operating system will examine these security attributes and determine if the request is valid, in other words, if the access can occur. An operation on any subject (resource) by any object (process or human) will be vetted against the security policy (rules for authorization) to determine if the operation is allowed.

One of the key features of the mandatory access policy is that the security policy is centrally administered and controlled. Users do not have the ability to override these policy settings. This is in direct contract to the discretionary model which allows data owners to control access to their resources. The advantage of the centrally located security policy is that it allows security administrators to implement enterprise wide security policies. The *Trusted Computer System Evaluation Criteria* (aka TSEC), also referred to as the “Orange Book” talking about MAC states “"it is a means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity". Mandatory Access Control and its variations are widely used in many of today’s consumer based electronic and computer systems. For example there is an implementation for Linux called SELinux and one for Windows called Mandatory Integrity Control, which has been part of Windows since Windows Vista. Mandatory policies are also incorporated into the MAC OSX and the MAC iOS.

Role Based Access Control

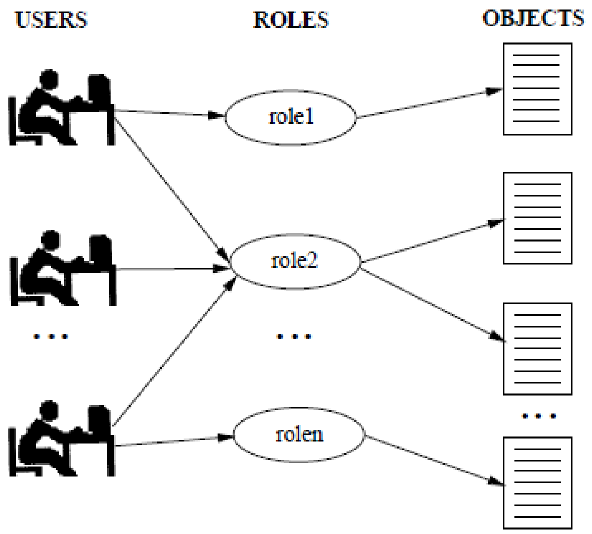
Within any organization, there exist a number of roles. Each of these roles will be responsible for executing different functions within the organizations ecosystem. Each role may need access specific resources and may need the ability to execute action steps on these resources. Users are assigned to roles and inherit the permissions of those roles. Since individual user Ids are not given any specific permissions, only roles are given permissions, it makes the management and maintenance of permissions much easier.

Based on the above description it can be deduced that Role Based Access Control (RBAC) is an access policy determined by the system, not the data owner. RBAC is used in commercial applications and also in military systems, where multi-level security requirements may also exist. Three primary rules are defined for RBAC:

1. Role Assignment: A subject can access a resource or execute an action only if it is a member of a suitable or appropriate role that has permissions.
2. Role Authorization: Determines that the role of the subject has appropriate authorizations. Taken in conjunction with Role Assignment, this rule ensures that users can take on only roles for which they are authorized.
3. Transaction authorization: A transaction can only be executed by a subject only if the transaction is authorized for the subject's active role. In conjunction with the Role Assignment Rule and the Role Authorization rule this rule makes sure that users can execute only transactions for which they are authorized.

The Diagram 2 below presents a visual representation of a Role Based Policy implementation.

Diagram 2. *Visual Representation of Role Based Policy*



In addition, other constraints could be applied and roles can be combined in a hierarchy where higher-level permissions supersede sub-roles permissions.

Comparing RBAC, DAC and MAC

RBAC is different from DAC because DAC allows data owners to control access to their resources, whereas in contrast, RBAC access is outside of the user's control. Even though RBAC is non-discretionary, it can be distinguished from MAC primarily in the way permissions are handled. MAC controls read and write permissions based on a user's clearance level and additional labels. RBAC controls collections of permissions that may include complex operations such as an e-commerce transaction, or may be as simple as read or write. A role in RBAC can be viewed as a set of permissions.

How Secure are Today’s Mechanisms for Authentication?

So far, we have looked at what identity management means in today’s world; why it is important and what are the risks of not managing identities properly. We have also examined the concept of federated identities; i.e. the concept of sharing identities across different security domains and how this helps in implementing making for a safer, lower cost system that also provides a better user experience. We also examined the SAML framework [7, 5] and how it is architected to provide a frame work for implementing a federated identity framework. We then went on to examine various access control mechanisms, how they work and the advantages and disadvantages of some of these systems. The one thing that everything we have examined to this point is that at the lowest level, they all depend on password based systems to authenticate the subject. Remember, the process of authentication means the process of a subject proving that he or she is who they say they are. This is typically done by the subject entering a user Id (either a system issued identification or a self-generated identification) and a password. This combination of user Id and password forms the basis of the subject proving they are who they say they are. Upon successful authentication, authorizations are carried out and applied to the subject. However, with the significant increases in computing power, the ability of computers to crack passwords is increasing year upon year. This coupled with the fact that many people, still choose passwords that are extremely simple, is making the traditional authentication method of user id/password more and more vulnerable.

There have been some wide scale changes in the way that organizations require their employees to authenticate themselves when logging into a company network remotely. The RSA token is almost ubiquitous in most large scale organizations. Employees are provided with an electronic token that displays a constantly changing one time password. In addition the employee is given a unique four digit pin that is tied to his or her particular one time password generator. When the employee attempts to authenticate into the network, he or she must enter his or her secret pin as well as the temporary one time password that is displayed on the RSA token. In addition they must provide their organizational user Id (which is not representative of their name but is typically a randomly generated string that is impossible to correlate to their name). The combination of these pieces of information is what is authenticated. This kind of authentication is called multifactor authentication. This is because there are multiple factors required for a successful authentication. And the core principal is that the authentication process must consist of something you know and something you have. In this example, the user knows their user Id and their 4 digit secret pin. The something they have is the RSA token Id that is tied to their pin. Without that tangible thing, they would not be able to read the correct on time password and so would be unable to authenticate. Interestingly, Google has started providing a similar feature to users of Gmail. It works in the following way. A user of Gmail can setup their account so that a onetime password is required whenever the account is accessed from a computer or device that is not explicitly acknowledged as a trusted device. Once the user enters a validate set of credentials on the sign in page, Google will text a onetime password to the account holders cell phone. The onetime password is valid for thirty seconds and must be entered into the challenge page presented to the user in order for them to be able to access their account. This ingenious method replicates the functionality of the RSA secure token and makes it available to the masses, without have to setup the costly infrastructure of the RSA token authentication service. So this is definitely a huge step forward from relying on a password only based system.

Looking Ahead: The Use of Biometrics as an Access Control Mechanism

Biometrics has been around for some time but it has not been widely adopted largely due to the fact that is extremely costly. The actual data points that are available for potential biometric use are mind boggling. The biometrics institute lists the following as potential sources of biometric data usage [2]:

- DNA Matching. This would be a chemical biometric that would be able to identify an individual on the basis of a segment of DNA.
- Ear. A visual biometric based on the shape of the ear.
- Iris recognition. A visual biometric that uses the features found in the iris to identify an individual.
- Retina recognition. A visual biometric that uses the patterns in the veins in the black of the eye to identify an individual.
- Facial recognition. A visual biometric that uses the facial features or patterns to identify an individual. Most facial recognition systems use either eigenfaces or local feature analysis.
- Finger print recognition. A visual biometric that the ridges and valleys (minutiae) found on the surface tips of a human finger to identify an individual.
- Finger Geometry Recognition. A visual biometric that uses the 3D geometry of the finger to identify an individual.
- Gait recognition. A behavioral biometric. Use the way an individual walks to identify them
- Hand Geometry recognition. A visual biometric. Use of the geometric features of the hand to identify an individual.
- Signature Recognition. This is a combined visual and behavioral biometric.
- Typing recognition. This is a behavioral biometric. The use of the unique characteristics of a person typing for establishing identity.
- Vein Recognition. Vein recognition is a type of biometrics that can be used to identify individuals based on the vein patterns in the human finger or palm
- Voice Recognition. This consists of the following:
    - Speaker Verification and Authentication. This is auditory biometric. Speaker verification is usually provided as a gatekeeper in order to provide access to a sensitive system such as an interactive voice response application for a banking application. These systems operate with the user’s knowledge and require their cooperation.
    - Speaker Identification. This is the task of determining an unknown users identity.

As can be seen from this list, the potential sources for biometric authentications are vast. However since the cost required to implement these mechanisms is so high at the current time, the probability that these will become common place anytime soon is slim. They will exist only to protect the most sensitive of data, data that is gleamed top secret. There is one biometric however that is starting to be used as a form of authentication. Voice recognition is being adopted by mid to large companies as a means for employee authentication. Albeit still in its infancy and not being used to authenticate access to any critical data, it is starting to be used as a peripheral alternative for employees to do simple tasks such as resetting network passwords, that they would otherwise have to call up the help desk for. The process is as follows. The employees of the company are given a one sentence phrase. Using the organizations telephone system, they will call a specific number, enter their employee identification number using the telephone dial pad, and when prompted, will speak the phrase. The phrase will be recorded and associated with their employee Id. Later on, when the employee needs to reset a password, they will call a telephonic application which will prompt them to enter their employee id and then prompt them to say the phrase that they had recorded earlier. Biometric software will then match the employee's voice print against the voice print on record. There are one hundred data points that the software can match on. If the software determines that the voice print is a match, it will allow the caller to proceed and reset their password.

As the cost of implementing biometric solutions drops and as more data becomes available about its reliability, I predict that its use will become increasingly widespread. Returning to the idea of multifactor authentication, something you know and something you have as means for authentication, it is extremely viable that biometrics could serve as the something you have.

In fact we are starting to see biometric solutions being adopted by many smart phones that run the android operating system. Facial recognition as a biometric is an available option on the latest android phones. In its initial release the phone required the user to look at it, in order for it to unlock. However, this mechanism was soon cracked by hackers who used still images (photographs) to fool the authentication mechanism into unlocking the phone. In a second revision, Google modified the mechanism so that the user had to blink while looking at the phone. Again the hackers managed to break the system. In its newest release, Google are now asking users to concoct a distorted face and use that as the authentication biometric. A few examples they gave were to stick the tongue out, or to twist the eyebrows. According to Google, these “modified” biometric images would be much more difficult to break.

Conclusions

We have certainly come a long way in the last decade. All aspects of our lives from business to social have unrelentingly marched onward toward the technological drum beat. Yet, in many ways we are more vulnerable than ever. The overwhelming majority of online systems rely purely on an outdated user Id/ password mechanism for authentication. Coupled with this is the fact that the vast majority of users of these systems still used passwords that are “easy for them to remember” and easy for others to guess. In my opinion we are overdue for an evolution in the arena of access control mechanisms. Identity management has become more sophisticated over the last decade with the advent of federated Ids and mechanisms to exchange them such as SAML, however, these are mainly restricted to the confines of private organizations. What is the single sign on solution for a typical consumer who has multiple on line identities on Amazon, Google, Apple and a whole host of other on line merchants. The short answer is that at present there is no solution. The consumer is left to fend for themselves and struggle to manage and maintain his or her multiple online identities. We are definitely in need of an evolution in this arena. The risks of not doing do are tremendous. As people start to live their lives online, they are becoming more and more vulnerable and equally important, this could have a knock on effect in the ecommerce space, which is a multi- billion dollar industry and thus could have huge economic implications.

References

[1]Christopher Steel, Ramesh Nagappan and Ray Lai "Core security Patterns: Best Practices and Strategies for J2EE, Web Services and Identity Management", pp. 360-372, October 2005.

[2]http://www.biometricsinstitute.org/pages/types-of-biometrics.html.

[3]http://spdp.di.unimi.it/papers/sam-fosad.pdf.

[4]http://en.wikipedia.org/wiki/Access_control.

[5]https://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf.

[6] http://developergeeks.com/article/51/how-to-securely-expose-webservices.

[7]https://www.pingidentity.com/resource-center/SAML-Tutorials-and-Resources.cfm

[8]Access Control: Policies, Models, and MechanismsPierangela Samarati, Sabrina Capitani de Vimercati Foundations of Security Analysis and Design Volume 2171 of the series Lecture Notes in Computer Science pp 137-196, October 2001.

[9]E. Bertino, Database Security, Approaches and Challenges; IEEE Transactions on Dependable and Secure Computing, Jan. 2005.