Identity Management

marc.sel@pwc.be

Context for Identity Management

To say that there are many different views on how identity should be established and managed is probably an understatement. Identity is important, and history shows it has many different facets. On the everyday level, your identity is the proof you need for receiving welfare or housing benefits, but on a more sinister level it can lead to genocide.

In many situations, identities and authorisations are not trivial. Consider transactions where monetary value is involved, where medical records are updated, or where sensitive items such as drugs for hospitals or arms for a police force are traded. The more granular you want controls to be, the more complex their management becomes. Consider the exponentially increasing number of features to manage when you want to control access to a building, to an individual room, to a transaction, to a field in a database etc. Needless to say that in many cases the situation is actually less under control than desirable.

On the other hand your reading of this article does not seem to imply much identity management. Unless the article would be published e.g. electronically, in a subscriber-only magazine, and the editor would like to be sure that only paying subscribers can read it.

Historical background

Identity and access management has a colourful history. In the 1970's, significant work was done in the area of OSI (Open Systems Interconnection), culminating in standards such as ISO 7498 (ISO1) and ISO 7498-2 (ISO2). These foundations were further elaborated in OSI/ODP (Open Systems Interconnection / Open Distributed Processing) security, for example in EC sponsored projects such as COST-11 Ter (COST1).

In the commercial world, as most organisations relied on systems from different vendors, security implementations hardly met the expectations. Architectures such as SNA or DECnet were predominant, and did not integrate well with OSI. Niche-solutions have been available which allowed e.g. Single Sign-On across IBM mainframe applications. However, these solutions rarely, if ever, inter-worked with non-SNA systems. Many organisations relied on a mainframe, where authentication was based on a userid/password that was transported in cleartext from a 3270 terminal to a transaction monitor. Authorisation was based on products such as IBM's RACF (RACF1).

Nevertheless, there were many circumstances where interoperability was required, which led, amongst others, to the ISO's standardisation of email (X.400) and directories (X.500). With the arrival of public key cryptography, the X.500 directory and the corresponding X.509 authentication mechanisms evolved into Public Key Infrastructure systems that provide strong mechanisms for authentication. In the year 2000, ISO published the X.509 v.4 standard, which included both standards for PKI (Public Key Infrastructure – providing authentication mechanisms) and PMI's (Privilege Management Infrastructures – providing attributes for access control).

Microsoft invested significantly in the Windows distributed security model. Windows2000's security coupled with the Microsoft Meta-directory Server and Active Directory led to a powerful instrument. This will probably be extended further with the launch of Windows2003, that can be expected to incorporate more .NET and Microsoft 'Passport' features. However, this model is mainly oriented towards a Windows-only universe.

The arrival of web technology greatly facilitated the construction of extranets, which led to a significant increase in demand for security solutions, including authentication and authorisation aspects. Due to scalability requirements, symmetrical solutions such as e.g. Kerberos had a natural disadvantage, and PKI-based solutions came more to the forefront. We saw the arrival of products that used a userid/password or a client-side certificate to authenticate a user. The authenticated user receives a cookie (or a ticket, or an attribute certificate), which stores security-related information on the client side. The integrity and/or confidentiality of such a cookie can be cryptographically protected. The cookie can be used to provide authentication and access control information to multiple applications, ranging from portals to back-office systems. Instead of a cookie, an attribute certificate can also be used.

It is important to realise that such security technology can be deployed both at application and infrastructure layer. The technology was first used to 'keep the bad guys out'. However, with the proliferation of web technology, the technology made it's way into all layers, and now it "allows the good guys in". In 1996 in Belgium, KBC Bank were the first to launch browser-banking. They based the security of their Internet retail banking service on an in-house PKI, storing public keys in a database. Here digital signatures are used at application layer to guarantee integrity and non-repudiation, complemented by encryption to guarantee confidentiality of transport.

During the 1990's, to 'let the good guys in', systems became increasingly sophisticated in supporting:

- a wide range of authentication schemes (from anonymous to certificates, smart cards and biometrics);
- the management of large groups of users (both internal and external one) typically through a directory (LDAP) and some form of delegation management;
- integrating with existing authentication mechanisms e.g. on IBM mainframes, Windows or Unix platforms;
- integrating with a range of web-enabled applications through agents or reverse-proxy technology, for passing-on both authentication and authorisation credentials;
- propagation of authorisation decisions from the management system to all the affected target systems (typically across many different technologies such as mainframe, Unix, NT, W2K, etc).

Today cryptographic protection can be found in a variety of places, ranging from application to infrastructure. For examples, refer to [EXAMP1]. As a consequence, we find many security controls embedded in most leading-edge systems, both at application and at infrastructure layer.

The problem space of Identity Management

Identity Management (IdM) is concerned with the lifecycle of individuals and their authorisations in an ICT[1] universe. A simple 3 dimensional representation allows us to visualise its key aspects.

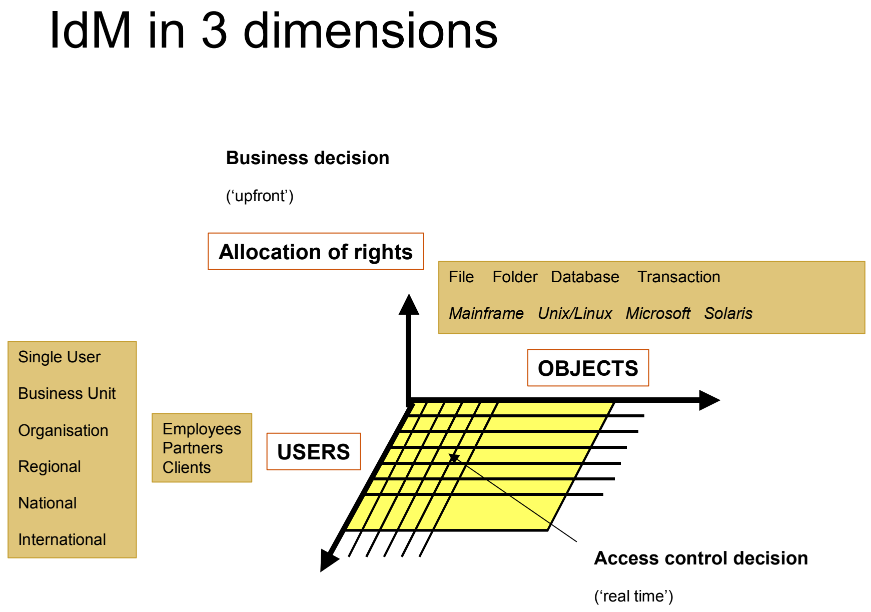


Figure 1 Three dimensions in Identity Management

We define users along the first axis. User communities range from small (e.g. employees with an SME[2]) to large (e.g. the population of a country, accessing the national e-government portal). The objects that will be accessed by the users are defined along the second axis. They range from files and folders to transactions, printers, etc. It can be noted that when a users attempts to access an object, this requires a real-time decision from the access control enforcement function. Finally, the allocation of privileges (or authorisations), is defined along the third axis. The objects' owners typically perform this. It can be noted that decisions with regard to granting or revoking privileges are typically taken 'upfront', as opposed to the real-time decisions when a user is actually requesting the access.

It can be argued that the third dimension (the allocation of privileges) can also be described as part of the first dimension (the user community), since decisions with regard to privileges will essentially be taken or driven by users. As such, the traditional two-dimensional access matrix[3] should be sufficient to represent the problem space. However, we think this representation does not do enough right to the notion of ownership in a DAC[4] context.

[1] ICT: Information and Communication Technology
[2] SME: Small to Medium size Enterprise
[3] as underlying e.g. the Bell-LaPadula model [BLP]
[4] DAC: discretionary access control

The four components of identity management

Identity management has recently been coined as a term to describe four components of computer security: authentication, authorisation, user provisioning and the deployment of LDAP-based directories. We can easily relate these four components to the three dimensions introduced earlier.

Authentication takes place along all three dimensions. It is commonplace for users to be authenticated. However, mutual authentication might require that objects be authenticated to users as well. And of course those parties granting accesses need to be authenticated as well.

In most cases, a user when accessing an object requires authorisation. This applies to the case where a user performs a business transaction, as well as to the case when an owner grants permission to a user.

Directories store users and objects in a tree format, which conveniently models organisational grouping. Directories store all users involved, typically with a number of attributes. Often policy and object-related information is stored too. With the users and the objects, many types of attributes can be stored. While historically each system had its own directory (e.g. IBM mainframes rely on their 'Global Catalog', Windows relied on the registry and the Active Directory, etc), there is a strong convergence towards the use of LDAP[5] directories today.

User provisioning is the process that enrols people, and makes sure they receive the appropriate authorisations to perform their tasks. This typically includes a verification of their claimed identity (e.g. on the basis of some credential such as a national ID card or comparable evidence), and the creation of accounts on all appropriate systems. These accounts then need to be granted the appropriate authorisations. When a user changes role within the organisation, this needs to be reflected within his authorisations. And finally, should a user leave the organisation, all authorisations should be withdrawn.

Let us now review each of these four components slightly more in detail.

[5] LDAP: Lightweight Directory Access Protocol

Authentication

Two types of authentication are traditionally distinguished[6]:

- Entity authentication, or the corroboration of a claimant's identity through actual communication with the verifier by the execution of a protocol; the claimant's identity is typically established for a limited duration; and
- Message authentication (sometimes also referred to as source authentication), where the authenticity of the message's source is demonstrated.

Identity Management is primarily concerned with entity authentication, and we witness a re-use of 'traditional' solutions - something you *know, have, are, do voluntary or involuntary, ...* This is a rich field with an ISO standard (ISO 9798) and many existing solutions. These solutions can be structured into the fields of passwords, challenge-response systems, zero-knowledge systems and 'customized' systems (including e.g. biometrics).

Password-based systems are commonly used due to their limited cost & high convenience, but have traditionally been viewed as weak. However, recent research (refer to the session by Paul van Oorschot) seems to offer new possibilities. Challenge-response solutions relying on the use of tokens (e.g. RSA's SecurID card, VASCO's Digipas) have been around for a long time but do not offer a panacea (a 'one-size-fits-all' solution). Smart cards have been around for a long time but require readers and software such as drivers. As such they have not been widely embraced. However, various implementations within private companies lead to successful projects (e.g. the PricewaterhouseCoopers smartcard in Belgium, granting all employees and staff access to buildings, computers, and including an electronic wallet). On the other hand, GSM and the SIM card have been very successful, with more than 700 million users worldwide. Firms such as Cryptomathic and Utimaco have products that offer interesting authentication functionality on the basis of SIM cards. With the arrival of UMTS and the USIM[7], its success will probably only increase, and new uses in an authentication context will certainly be found.

PKI (public key infrastructure) has in the past often been selected for its huge potential, but was equally often found to be complex in implementation. This led amongst other to the EU-sponsored pkiChallenge. Nevertheless, most national electronic ID-card initiatives are essentially based on PKI and smart cards (for example in Finland, Sweden, Estonia, Belgium, ..). There is a tendency to use different certificates for different purposes. For example, the Belgian EID card will make use of one certificate for authentication, and another one for signing.

[6] For a comprehensive treatment of authentication protocols, refer to [HAC]

[7] USIM: UMTS SIM

Authorisation

To discuss authorisations (also referred to as 'access control') we need to define some terminology. The simplest model is probably this: a user (or 'subject'), represented by a software process, requests some form of access to an object through an application. His access request is evaluated by the application, which will typically call upon a security reference monitor for a decision. Systems that implement access control through a focus on what users can do are often referred to as 'capability'-oriented. Systems that focus on what objects allow are often referred to as 'ACL[8]'-oriented. Obviously, both users ('subjects') and objects may need to be authenticated.

The oldest security models were created to meet the requirements of the US Orange Book and focused on confidentiality and information flow aspects. The origin of most models can be traced back to the original Bell-LaPadual model [BLP]. However there are at least two major problems with this model:

- The BLP model assumes that access rights are fixed, and its creators did not really contemplate the case of changing authorisations. For a real-world system, this can hardly be called a realistic assumption;
- The BLP model focuses on confidentiality and information flow, while in a commercial world, integrity is often considered of more importance than confidentiality.

In 1987, Clark and Wilson put forward their instrumental paper that discussed the need for integrity [ClarkWilson]. Gradually, Role Based Access Control (RBAC) became the predominant model. RBAC, with its focus on integrity, is present for example in Windows, SAP, Tivoli, etc. Unfortunately, there is no agreement amongst vendors with regard to an exact definition of RBAC. However, a commonly accepted model was put forward by NIST[9]:

[8] ACL: Access Control List

[9] NIST: the US National Institute for Standards and Technology

Role Based Access Control

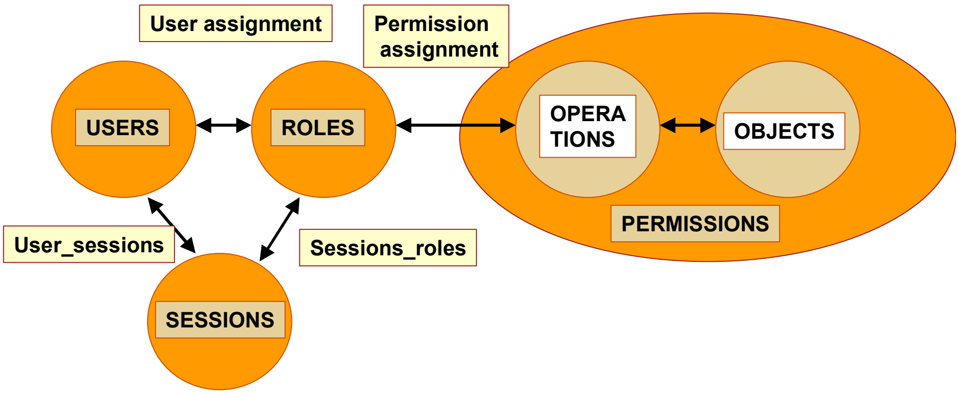


Figure 2 RBAC model

An important notion in the context of authorisations is 'Segregation of Duty'. The first extensive and formal evaluation of it was by Brewer and Nash [BrewerNash]. Specified in a more informal way, SOD includes the following roles in a typical organisation:

- The security officer/administrator decides on global options & mechanisms;
- The data owner controls the authorisations related to data resources – this may require further segregations, where we do not want that users capable of creating invoices can also authorise their payment;
- The computer operations responsible controls the authorisations related to production environment – his team has access to all (or most) information for operational purposes such as back-up and recovery;
- The systems programmer implements technical aspects, and controls the authorisations related to technical resources; and
- The auditor can independently review all systems.

Directories

The field of directories originated from ISO's X.500 directory, which aimed to be the global directory for a global X.400 email system. However, while the telephone operators were busy designing and building a well-crafted and scalable system that one day could become the world's email infrastructure, the Internet arrived. As a consequence, today we see the merging of many ideas of both worlds. Packet switching still lives on, but now in the form of the Internet Protocol. LDAP, originally a front-end to an X.500 directory, became quickly a stand-alone solution for directories. And with the proliferation of both standardised and proprietary directories, meta-directories arrived on the scene, allowing synchronisation and propagation of information. A typical situation is represented on figure 3 below.

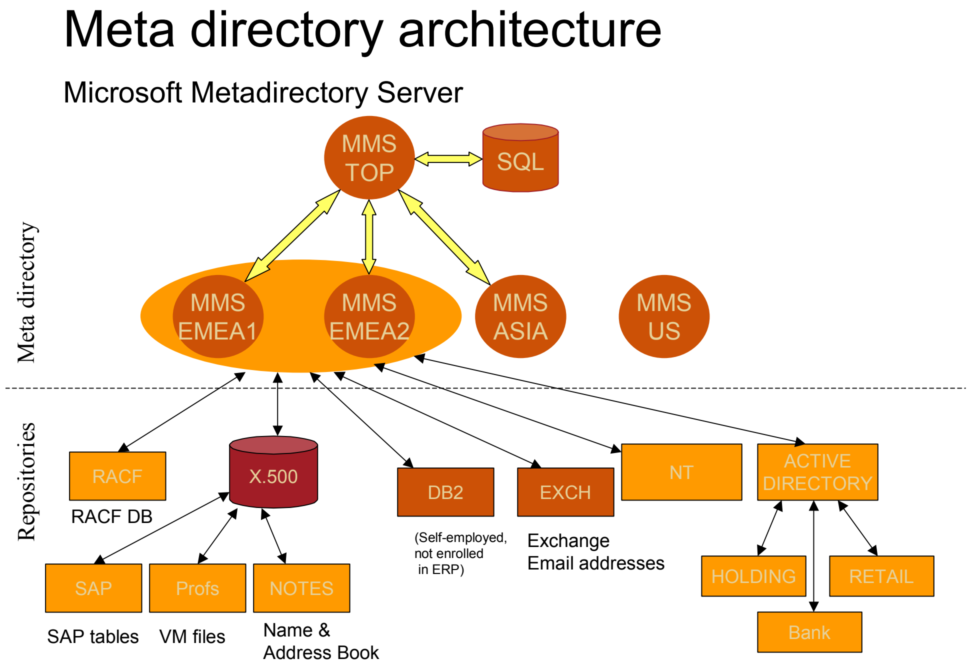


Figure 3 Sample use of MMS - Microsoft Meta directory Server

User management and provisioning

It is not sufficient for systems to offer authentication and authorisation features. In actual deployment situations, users and objects need to be enrolled, and access needs to be granted. Support for this (both in terms of supported systems through agents and for the process as a whole) has historically been rather weak, leading to much manual (and costly) activity. For example the user interface to IBM's RACF was (and today still is) quite powerful, but not really user-friendly. The complexity of the problem, combined with the diversity of systems implemented often led to a situation where many different version of an individual's information resided within an organisation. This in turn led to an unfavourable user experience, where users had to receive multiple user-ids and passwords, which were usually not synchronised, and behaved according to different rules. The expression ‘SSO – Single Sign-On’ was much used in the 1980’s and 1990’s, but mainly remained a marketing slogan. A significant breakthrough was probably Microsoft’s implementation of Windows security with the domain concept, which at least allowed SSO across all Windows servers within the organisation.

With the arrival of web technology, organisations started to open their applications for external users such as customers. As much of this access was HTTP-based, a new scheme for access control evolved. This is based on the following model:

- Users are registered in a central LDAP, and their privileges are recorded either in the LDAP or in a dedicated database;
- Access control is enforced through agents, either at the moment of entry through an inbound proxy server, or by installing an agent within the target application;
- These agents call upon a Policy Server, which will instruct them with regard to the authentication required;
- When the users supplies appropriate authentication credentials, an additional temporary credential is stored in the form of a protected cookie under his browser;
- On the basis of this new credential, the user can further access the target application.

Many systems for user management and provisioning have now been proposed and implemented, often leading to both a high degree of satisfaction of users and significant cost savings. Amongst the core functions of such systems we find data & password synchronisation, group & organisation management, support for different provisioning modes, often via workflow (centralised provisioning, delegated administration, self-registration), and self-service for end users.

IdM technology

IdM solutions can easily be represented in the three dimensional model introduced earlier on. Let us first consider some basic systems. A RACF system is limited to support the user management and provisioning on an IBM mainframe only. Users are enrolled in the RACF database (which is rather proprietary), and extending the scope of RACF beyond mainframe systems is rather difficult.

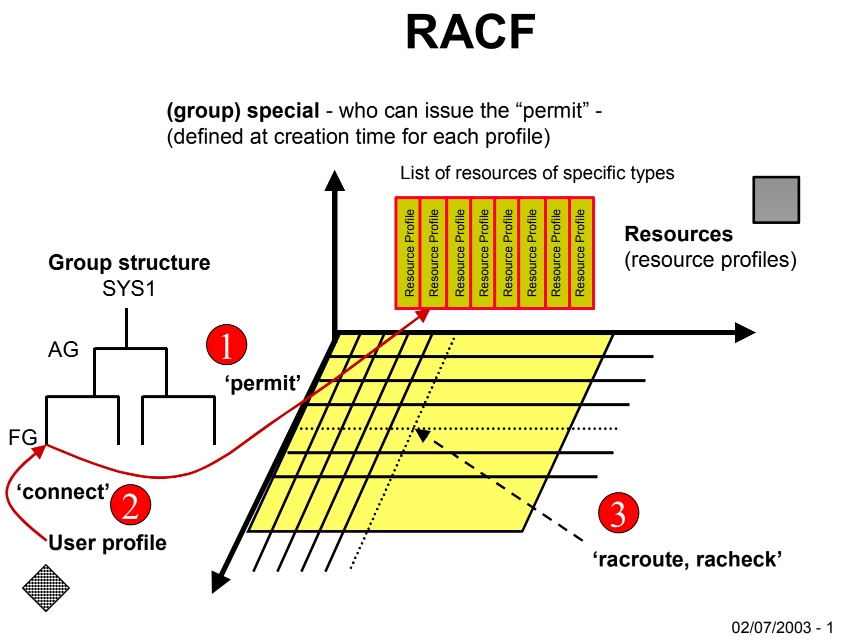


Figure 4 IBM's Resource Access Control Facility at a glance

It can be seen from figure 4 that there are three essential decision points:

- FG's (functional groups) are '*permitted'* access onto sets of resources;
- Users are '*connected'* to such functional groups;
- Finally, when the users requests an actual access to a specific resource, the Operating System's resource manager will call upon the SAF mechanism via the RACROUTE and RACHECK calls to request an advice from RACF (it remains the resource manager's responsibility to interpret the advice and to enforce the decision towards the requesting application).

As IBM's RACF was mainly limited to mainframes, complementary products such as those from Beta Systems arrived on the market. Their flagship Identity Management product is called SAM Jupiter (Schumann Access Manager). It is based on a central repository and agents for all the different platforms that are supported. Other products include for example BMC's Control-SA.

Unix is inherently simpler. Authorisations can be granted by the object owner or by root. Support for user management and provisioning is rather poor, and many vendor-specific extensions have been created (e.g. by SUN and IBM).

Unix

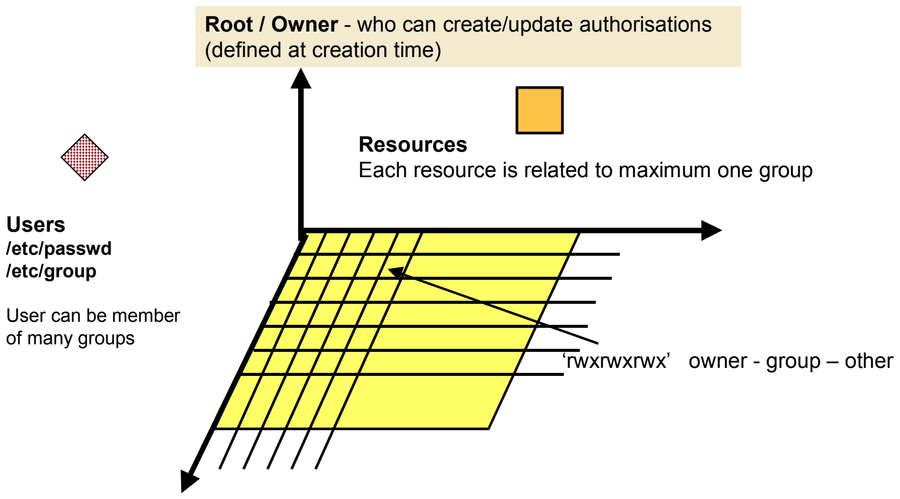
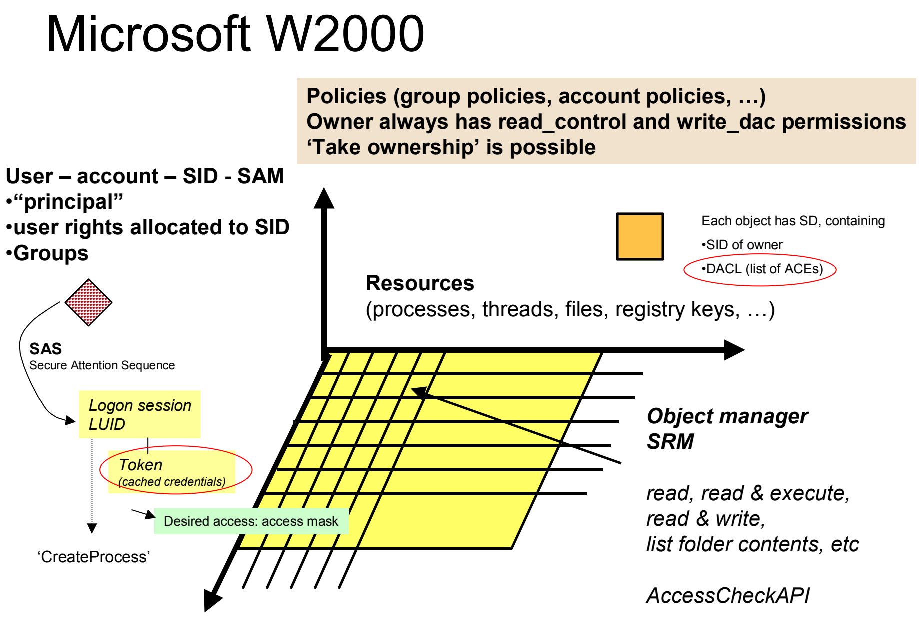

Root / Owner - who can create/update authorisations (defined at creation time)
Resources
Each resource is related to maximum one group
Users
/etc/passwd
/etc/group
User can be member of many groups
'rwxrwxrwx' owner - group – other

Figure 5 Unix authorisations at a glance

Microsoft's Windows model evolved significantly over time. The current Windows2000 model is depicted on figure 6 below.



Microsoft W2000

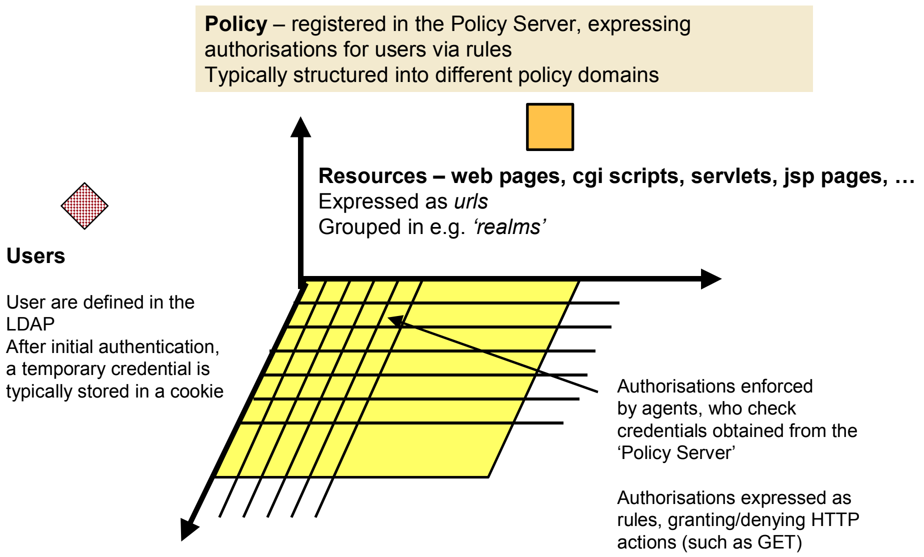
Policies (group policies, account policies, …)
Owner always has read_control and write_dac permissions
'Take ownership' is possible

User – account – SID - SAM
•"principal"
•user rights allocated to SID
•Groups

Figure 6 Windows2000 authorisations at a glance

Web technology-oriented solutions typically rely on a Policy Server. Their design starts from an organisational point of view, where the organisation is mapped into one or more LDAP trees. Authorisations are defined through rules, which are enforced via agents residing on the target systems. A user's first authentication (which can be implemented through user-id/password, SSL, smart card, etc) results in the storage of a cookie as a temporary credential. The agents on the basis of the cookie regulate further accesses. A typical set-up is depicted on figure 7 below.

Generic ‘Policy Server’ solution



Policy – registered in the Policy Server, expressing authorisations for users via rules
Typically structured into different policy domains

Resources – web pages, cgi scripts, servlets, jsp pages, …
Expressed as *urls*
Grouped in e.g. *‘realms’*

Users

User are defined in the LDAP
After initial authentication, a temporary credential is typically stored in a cookie

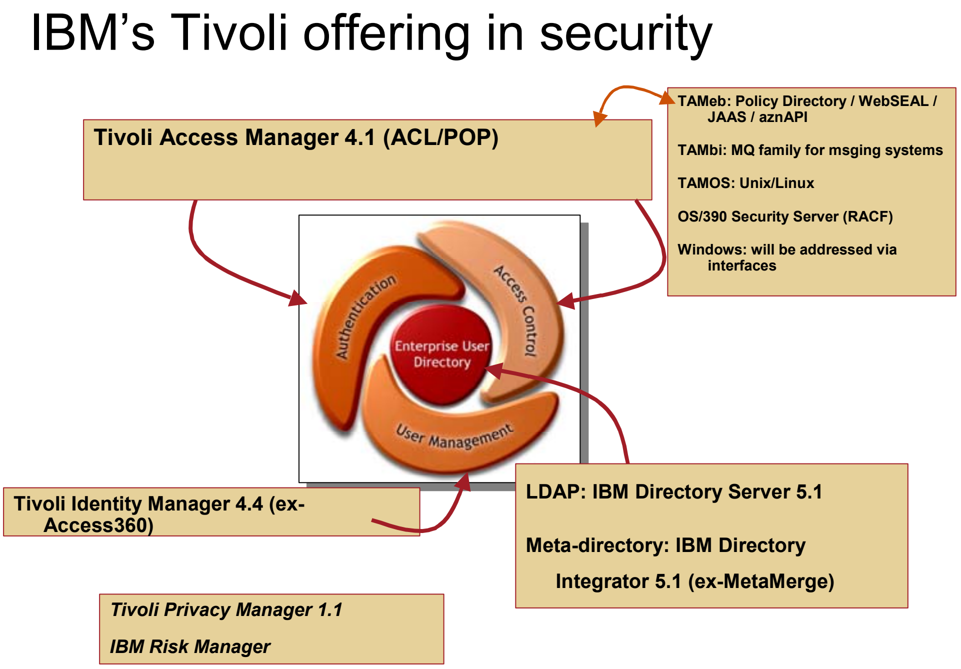
Authorisations enforced by agents, who check credentials obtained from the ‘Policy Server’

Authorisations expressed as rules, granting/denying HTTP actions (such as GET)

Figure 7 A generic web-oriented authorisation system at a glance

Many vendors ship such solutions, including but not limited to Oblix, Netegrity and IBM. Both Entrust and Baltimore have developed access control solutions on the basis of their PKI system (called GetAccess and SelectAccess respectively). In order to improve support for the user management and provisioning aspects, additional components have been developed or acquired.

It should be noted that IBM invested significantly in Identity Management, and rebuilt their product offering completely, almost entirely on the basis of acquisitions. We can represent the functionality of these acquisitions as follows:



IBM's Tivoli offering in security

Tivoli Access Manager 4.1 (ACL/POP)

TAMeb: Policy Directory / WebSEAL / JAAS / aznAPI

TAMbi: MQ family for msging systems

TAMOS: Unix/Linux

OS/390 Security Server (RACF)

Windows: will be addressed via interfaces

Tivoli Identity Manager 4.4 (ex-Access360)

LDAP: IBM Directory Server 5.1

Meta-directory: IBM Directory Integrator 5.1 (ex-MetaMerge)

Tivoli Privacy Manager 1.1

IBM Risk Manager

Figure 8 IBM's Tivoli offer in security

Research in the IdM field

Current body of knowledge

The main body of knowledge with regard to Identity Management evolved from the access control field. The most instrumental models in this field are Bell-Lapadula [BLP], Clark-Wilson [ClarkWilson] and Brewer-Nash [BrewerNash]. A relevant ISO standard is the ISO 10181 series.

With regard to RBAC, the work by Ravi Shandu is a landmark in the field. Within the IETF, RFC 2904 (Generic AAA Framework basic principles) and related RFCs illustrate a more Internet-focused approach.

Recent and interesting work includes the EC-sponsored research performed by David Chadwick in the context of PERMIS. Interesting work is also undertaken in the context of using XML technologies such as SAML and XACL.

With regard to authentication, there is a wealth of knowledge available. For a good description, refer to e.g. chapter 10 of [HAC].

Areas for further research

With regard to authorisation aspects, the following topics seem worthwhile to further investigate:

Balances

The many balances involved when establishing identities, including those between:

- centralised versus decentralised identity-establishing authorities;
- privacy and anonymity versus identity;
- public versus private identity-establishing authorities.

Actual permission set

Understanding and managing the actual permissions of a user in a cross-platform environment is typically a non-trivial case. Both the opening-up of systems to external users (customers, business partners, suppliers, …) and the changing organisational landscape due to mergers & acquisitions contribute to the complexity of the problem.

In this context, role-engineering and role-mining (ref [EUREK]) are techniques that become increasingly important.

A unified approach for modelling authorisations, transposable into the various technical and organisational universes is clearly lacking.

One possible way would be to structure the field into 3 sub-fields:

- The provisioning dimension – the up-front decisions, where parameters are set that influence the real-time decision; this includes both the user and the resource side;
- Real-time decision – the actual ‘yes/no’ decision when a subject requests access on an object; such a security decision is typically enforced by a Reference Monitor;
- Post-factum analysis to verify whether the real-time decisions are acceptable from a security policy point-of-view.

With regard to authentication aspects, the importance of zero-knowledge protocols cannot be overestimated, and we will probably see more sophisticated deployment as the need for anonymity and privacy increases.

Finally, it can be expected that the EC 6th framework will produce a number of interesting and relevant results with regard to Identity Management.

Conclusion

The IdM field gradually matured over the last 20 years, and picked up a lot of momentum the last 2-3 years with the deployment of web technology.

For most organisations, Identity Management addresses a pressing need which can only be ignored at the expense of effectiveness & efficiency. IdM has the potential:

- To reduce costs, for example by cutting down the sheer number of help-desk calls due to faster and more accurate provisioning, coupled to self-service features;
- To increase the quality of service (particularly ‘the user experience’), for example by increasing the data quality;

Key features of a good Identity Management system include multi-level delegation, role management, multi-platform support, workflow and self-service capabilities. A directory-based approach is a natural choice. Advances in Federated SSO models (SAML, .NET Passport, Liberty) hold a promise to federated identity management. Solving the Identity Management challenge today will give you more options tomorrow (and at a cheaper price).

Finally, it should not be forgotten that identity has other sides, such as anonymity and privacy, which should receive appropriate attention too.

References

[BLP] Bell, D.E. and LaPadula, L.J. (1974) Secure computer systems: mathematical foundations and model. The MITRE Corp, Bedford, MA.

[BrewerNash] D.F.C. Brewer and M.J. Nash. The Chinese Wall security policy, in Proceedings of the 1989 IEEE Symposium on Security and Privacy, pp 206-214, 1989.

[ClarkWilson] D.R. Clark and D.R. Wilson. A comparison of commercial and military computer security policies. Proceedings of the IEEE Symposium on Security and Privacy, pp 184-194, 1987

[COST1] Security Architecture for Open Distributed Systems (S. Muftic, A. Patel, P. Sanders, R. Colon, J. Heijnsdijk and U. Pulkkinen – Wiley Series in Communications and Distributed Systems, ISBN 0-471-93472-0, 1993)

[EUREK] www.eurekify.com - the Sage role analysis and data mining tools

[EXAMP1] Possible examples of crypto-based controls include amongst others:

- the browsers (Internet Explorer, Navigator, Konqueror, ….), where cryptographic keys can be stored, and where protocols such as SSL/TLS are supported;
- the web servers (Apache, IIS, …), where support for keystores and protocols such as SSL/TLS is also present;
- application servers such as Bea's WebLogic and WebPortal, or JBOSS/Tomcat, which can be controlled via PMI agents;
- SAP's ERP system, where SAP/R3's SNC (Secure Networking Communication – compliant to GSS API v2) and SSF (Secure Store and Forward – offering PKCS #7 support) are part of the offering;
- web portals, where authentication and access control can be based on certificates (either directly or via additional products);
- the Java language, where a full cryptographic API is part of the language, allowing developers to create applications that base authentication and access control decisions on cryptographic services; similar controls have now been built –in into .NET;
- VPN (Virtual Private Networks), where IPSEC certificates can be used to secure traffic between end-points of the network.

[HAC] Handbook of Applied Cryptography, by Menezes, van Oorschot and Vanstone, CRC Press 1996, available from www.cacr.math.uwaterloo.ca/hac

[ISO1] ISO - International Standards Organisation (www.iso.ch) - ISO 7498, the OSI model

[ISO2] ISO - International Standards Organisation (www.iso.ch) - ISO 7498-2, the OSI security model

[RACF1] IBM’s RACF (Resource Access Control Facility) is probably one of the world’s most popular access control packages for IBM mainframes. Competing products include ACF/2 and Top-Secret.