Introduction

Who are you? Are you who you claim to be? How do you prove you are who you say you are? Who approves your user profile and assignment? These are questions answered by Identity Management (IDM) systems. But these are only a few of the questions that must be answered when vetting users who are requesting access to systems housing sensitive, regulated data.

IDM systems, which usually work with user directories to authenticate users, represent only the authentication process of the AAA triad (Authentication, Authorization and Accounting) defined in Tech Target’s SearchSecurity site. All three elements of the AAA framework are required to intelligently control access to computer resources, enforce policies across multiple systems, audit usage, and provide the information necessary to bill for services.¹

Deployed alone, IDM lacks the granularity and authorization enforcement needed to protect against sensitive data leakage. This is particularly true when the system is operated by privileged users and administrators with root or other functional account access to systems. With IDM on its own, these privileged users are allowed to see and manipulate entire applications and their data sets, even if the users don't need access to those realms to do their jobs. IDM also cannot prevent the sharing of root and functional account passwords to these systems, as is commonly practiced among administrators for system and/or application management. This is in direct violation of many privacy mandates. IDM alone is also rather blind when it comes to the last A—Accounting (which translates to audit). For example, although they may be able to authenticate who a user is, IDM systems do not automatically consolidate user activity information and cannot capture details of a session to record what was actually touched and manipulated.

By combining Identity Management (IDM) with Enterprise Access Management (EAM), organizations can meet their AAA requirements in such a way that they adequately address compliance with their own internal policies and, at the same time, meet legal and regulatory requirements. This paper discusses the difference between IDM and EAM and explains how these two enterprise functions can work together for better control of access to operating systems, applications and related data.

The NIST SP-800-63 guidelines for implementing electronic authentication define IDM as a process of establishing confidence in user identities that are electronically presented to an information system. SP-800-63, then, covers four levels of identity assurance: registration, use of tokens, authentication protocols and related assertions.

Specifically, IDM is suited to support:

- Centralized provisioning for user authentication
- Centralized, local or remote authentication of users
- Single Sign-On (SSO) that incorporates tokens or other protocols to allow a user’s identity to be recognized by multiple systems
- Multifactor authentication—something they know (e.g., a password), something they have (e.g., a fob or smartphone application with a one-time password) and/or something they are (e.g., a biometric fingerprint, retinal scan or iris scan)

Figure 1 demonstrates how centralized provisioning works to create and control user authentication.

---

2 http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf
IDM represents a robust and well-adopted technology that has reached its functional limits. Identity Management systems are notoriously difficult to implement for reasons ranging from simple scalability to lack of functionality, as well as the need to customize or tailor the systems to each organization’s specific needs. Other common issues with using IDM alone include:

• **Lack of effective definition or enforcement of granular access controls:** Granular authority over user access and resources across multiple systems is key to controlling access. However, IDM typically lacks the ability to control access to specific servers and applications based on critical authorization factors, such as who the user is, which device they are using, when (date and time) they are accessing, and how they are gaining access.

• **Inability to control privileged access:** One of the most critical functions of any access management system is to control privileged access so that root, administrator and other special function account passwords do not have to be shared. Typically, IDM systems without EAM functionality that controls which commands a privileged user can run cannot eliminate privileged password sharing or control which commands can be executed by role and access route.

• **Inefficient user account management across disparate systems:** As employees are terminated or their roles change, IDM systems have difficulty keeping in sync. They typically cannot remove user accounts or change permissions across dissimilar operating systems easily. Tech centers of almost any size incorporate diverse systems, such as Windows, UNIX, Linux, AIX, HPUX (and likely multiple versions of each). Because of the number and diversity of systems, IDM systems may not know which accounts reside on which servers. As a result, they may leave vestiges of provisioning that should be changed or deleted when a user is terminated or transferred to a new department or location. The best security in the world cannot guard against a rogue ex-employee who chooses to access data using his or her user account when that account was not removed upon the employee’s termination.

• **Inability to capture the right information:** IDM systems operate at such a high level that they cannot see what is going on in a particular system. They also do not consolidate detailed user activity logs unless those logs pertain to the administrators of the IDM. Consolidated activity logs, which are critical for compliance reporting, auditing and forensics, cannot be accomplished with IDM alone. On the other hand, EAM can capture actual keystroke logs under some circumstances involving highly privileged users.
• **Lack of scalability:** One of the ongoing challenges for any organization is authentication and authorization across multiple operating systems and multiple applications. With the growing popularity of cloud computing, this issue becomes even more complicated, particularly if the organization is tapping into an external Internet cloud service rather than using an internal cloud. Even if an IDM system does scale, it might do so only partially—that is only for certain operating systems. Alternatively, the IDM might require custom programming in order to scale. Some IDM systems are geared more to web applications and might not play well with traditional client–server apps or large, feature rich “fat” applications, for example.

Some vendors classify their product as an IDM with EAM functionality, and readers might encounter products labeled as an IDM that can perform some access management functions. As a result, many of the above statements are qualified with the words *might* or *typically*. That being said, IDM without EAM is simply an incomplete solution when it comes to protecting access to sensitive systems. To illustrate this, an IDM project created by Dartmouth University describes identity management as being “concerned with storing and verifying the details of an individual’s identity and, in some cases, the associated rights and privileges that users are assigned within systems. [Emphasis is the author’s.]”

---

3 [www.dartmouth.edu/comp/systems/initiatives/idm/benefits.html](http://www.dartmouth.edu/comp/systems/initiatives/idm/benefits.html)
EAM Definition and Benefits

IDM systems focus on ways to determine whether a user is who he or she says and whether that user should have access to a physical or logical system specified in policy. EAM systems, on the other hand, focus on providing appropriate levels of access within the system to the identified user. NIST SP-800-53 defines Identity Based Access Control (IBAC), another way of saying access management, as access control based on the identity of the user. In essence, it controls a user’s access to the specific objects that are assigned based on his or her identity. The IBAC applies to access control by a single system (see Figure 2).

Instead of controlling access to a single system or a subset of systems based on a unifying factor (such as an operating system), EAM controls access (physical or logical) to the entire enterprise based on the IDM system’s authentication of the user.

There are many benefits of using EAM for user authorization and resource management. For example, the U.S. Department of Health and Human Services (HHS) developed their EAM system to provide user mapping and access management for HHS and its operating divisions. HHS has realized many benefits from its EAM environment, including the following:

- Improved authentication security
- Multifactor authentication support for privileged users, such as smart card, soft certificate and other authentication types
- Reduced manual effort required to complete FISMA and SAS70 audits due to robust identity auditing and reporting capabilities and access control models
- Reduced help desk costs by decreasing the number of accounts and passwords a user must manage
- Increased convenience for users by leveraging a single credential to access multiple applications
- Support for Personal Identity Verification (PIV) cards (rather than shared passwords) for privileged users
- Further Process definition/documentation around application administrators “on-boarding” process

Figure 2. Access Management by a Single System

5  www.hhs.gov/ocio/esolutions/hhsidentity/goals_for_enterprise_access_management_.html
Other advantages of combining EAM with IDM include the ability to control granular authentication and authorization enforcement based on access policies such as when the user logged in, how they logged in, the source of login, what protocols they can use (telnet, ftp, SSH), and granular command control. With the addition of EAM, organizations can also control user activity on a system, including which commands those privileged users can execute (see Figure 3).

You can set up combined EAM and IDM Systems to enforce against a variety of factors, including the following:

- Who the user is
- What role the user plays in the organization (particularly important for privileged users)
- What commands the user is authorized to perform (especially privileged users)
- What systems the user can access
- What method of access is approved for the user (local, remote, SSH, telnet, and so on)
- What source for access is approved for the user (IP address, workstation, VPN, and so on)
- What days and times are approved for the user to access the data
- What combined authentication methods must be used (single or multifactor)
- What authorization rules apply (based on source console or other factors)
- Whether or not the session should be deeply monitored with keystroke logging

Merging IDM and EAM ensures appropriate access for new users is automatically provisioned across all systems.
as approved and set up. Equally as important is the de-provisioning of user access, which can be centralized through the IDM plus EAM to remove accounts automatically across all systems without an additional cleanup requirement. Users transferred from one job function to another can be automatically re-provisioned to the appropriate systems via the EAM system in the same way with a single step, as shown in Figure 4.

![Figure 4. Provisioning Users with IDM + EAM](image-url)

Centrally controlling appropriate user access through such a system improves efficiency. Savings in time and resources are obvious; perhaps not as obvious, however, is the timely response to requested changes. Through improvements in coverage and operational efficiencies, these savings translate into higher productivity and/or higher revenue.

In addition to improved efficiency and lower costs, the benefits of combining EAM and IDM are the same as those listed in a recent SANS Analyst white paper on extending Role Based Access Controls (RBAC) for better granularity and control.

8 These benefits include the following:

- **Control and removal of dangerous shared roles/accounts:** Privileged access users (such as root, administrator and others) and their passwords can be protected, disabled or removed as appropriate to each system.

- **Protecting data and objects not just systems:** Access to sensitive data can be blocked from system administrators while still allowing them to do their jobs on those systems. Granular, segregated access and permissions can be achieved, down to operating system resource and object level, without kernel intrusion.

- **Scalability and ease of use:** Consistency can be created and maintained in roles across multiple systems in a way that is understood by business personnel who request those role assignments.

- **Role mapping:** Roles can be remapped painlessly once the system is fully operational, thus allowing the systems to keep up as roles and access rules change.

---

8 [www.sans.org/reading_room/analysts_program/access-control-foxt.pdf](http://www.sans.org/reading_room/analysts_program/access-control-foxt.pdf)
Implementing EIAM

One way to accomplish integration between IDM and EAM is to use a web services interface to link the IDM system to the EAM system, creating an EIAM. This configuration allows for secure data transfer between the two systems in real time so all add/change/delete activities can happen in real time from one console.

Consider a large financial organization with over $500 billion in annual revenue and almost 100,000 employees. In this case, the IDM and EAM are from two different vendors, but the organization was able to link the two together over a web services interface. With over 3,500 UNIX and Linux hosts and the need to comply with Graham-Leach Billey Act GLBA and other regulations—and with careful planning of all aspects (from assessment of practices and identity systems to full integration for seamless management) the EIAM was able to:

- Seamlessly integrate into the organization's multifactor authentication systems as part of the user authorization process
- Automate audit logging of all key events consolidated from across diverse servers for simplified compliance reporting
- Delegate privilege, including privileged access to commands or applications, removing sharing of privileged passwords
- Centralize administration, including password quality administration
- Avoid intrusion into the operating system without modification of the kernel or system libraries

This hypothetical case study shows the importance of careful planning. In addition, IT teams also need to work closely with existing vendors and locate other vendors who are willing to work with each other.

Many organizations already have an IDM vendor that typically works well in their environments. In these instances, the organization might want to add EAM without changing its IDM vendor, which would require additional interoperability to be established between two vendors. In other projects, it is possible to use a single vendor to change from internally developed or legacy access control systems to EIAM in a single solution, as discussed in the Dartmouth project mentioned previously.

An organization must make an early decision as to whether it intends to develop its own hybrid system or bring in a vendor or two with tested, mature, well-referenced plug-and-play tools. Once that decision is made, the organization must move forward in earnest with planning. An appropriate warning here is that creating one’s own EIAM system is incredibly complex. As an illustration, you might want to peruse the State of Louisiana’s Department of Children and Family Services RFP for their CAFÉ (Common Access Front End) system and the list of 1,072 requirements (on 73 pages) they included in their 338-page RFP.

Rolling out a new EIAM system to every user in the organization can be expensive in terms of time, money and staff resources. Therefore, Joel Dubin recommends phased-in stages, starting with selected groups of users (rather than with the entire company), and implementing the rollout over incremental periods of time. An incremental implementation allows technicians to work out kinks and bugs before they affect too many users and gives users time to get used to the new controls. In terms of time investment, Dubin’s Tech Target article predicts that a successful EIAM deployment for a large company can take anywhere from six months to two years to implement.

---

9  www.dartmouth.edu/comp/systems/initiatives/idm/
10  www.dss.state.la.us/assets/docs/searchable/LRS/ACCESS/rfcps/090810CAFEModernization_RFP.pdf
As said, converging IDM and EAM successfully requires careful planning. What follows is an organizational checklist for adding EAM to IDM:

1. **Planning—carefully determine the following:**
   a. The appropriate internal personnel to serve on an EIAM Design Team
      i. IT personnel—System administrators
      ii. Business personnel—Those who know the business roles
      iii. Management personnel—Those who will approve users for roles
      iv. Vendor personnel—Necessary where the existing third-party IDM will be used in the project
   b. User roles based on job functions—Separation of Duties (SoD) requirements
   c. Appropriate role hierarchy—Roles that can be inherited by other higher level roles
   d. Static attributes that might be applicable
      i. Physical location
      ii. Job function
      iii. Job position
      iv. Special clearances required to access certain data
   e. Dynamic attributes that might be applicable
      i. Time of day of access
      ii. Temporary work assignments
      iii. Circumstantial special clearances to access certain data
      iv. Other attributes appropriate to the environment
   f. Objects that need to be controlled, including mapping to Access Control Lists (ACLs)
   g. Commands required to perform the role upon the object
   h. Access and authorizations required for each role
      i. Mapping of users to roles to ACLs/objects
   j. Audit requirements
      i. System access
      ii. User add/change/delete
      iii. Object access
         1. File access/add/change/delete
         2. Records access/add/change/delete
         3. Printers

2. **Set requirements for the EIAM system.**
   a. Determine the preferred method(s) within the organization for controlling access across the environment:
      i. Standards that must be followed
      ii. Third-party software
      iii. Internally developed code
      iv. Web services adaptations or other translators to transfer data between the IDM and EAM
   b. Create detailed Request for Information/Proposal (RFI/RFP) document(s) to provide to prospective vendors and service providers

3. **Establish a team for vendor selection and Proof of Concept (PoC):**
   a. IT team (including the IDM vendor, if appropriate)
   b. Business team
   c. Management team
Conclusion

Using IDM without EAM leaves organizations vulnerable to data breaches and at risk of compliance violations. This type of model is not sustainable in most enterprises with diverse operating systems, applications, databases and other systems.

Today’s needs include protecting data from unauthorized user access, managing user resources to the object-level, and providing visibility and restrictions around user access that can scale for today’s complex enterprises. Access controls should be user-aware, context-aware, and able to support strong, multifactor authentication. The user activity data automatically consolidated from converged systems can show auditors, investigators and regulators who accessed what systems and data, from where, when, and more—thus achieving audit compliance.

Identity management is only part of the answer. Without implementing enterprise access management, the organization will not be able to achieve the enforcement of user access to systems and data needed to reduce the risk of insider fraud conducted by privileged users nor address key compliance requirements. Combining the two takes careful planning, but in the end, implementing an EIAM will result in more efficiency, improved compliance, and more comprehensive access and authentication systems.
J. Michael Butler, GSEC CISA GCFA EnCE, is an information security consultant with LPS, a leading provider of technical services to the mortgage industry. Butler's responsibilities have included computer forensics, information security policies (aligned to ISO and addressing federal and state disclosure laws), enterprise security incident management planning, internal auditing of information systems and infrastructure, service delivery, and distributed systems support. He has also been involved in authoring SANS security training courseware, position papers, articles and blogs. Butler has more than 28 years of experience in the computer industry.

SANS would like to thank its sponsor:

FOXT.
# Upcoming SANS App Sec Training

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Location</th>
<th>Dates</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS 2020</td>
<td>Orlando, FL</td>
<td>Apr 03, 2020 - Apr 10, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>Mentor Session - DEV522</td>
<td>Novi, MI</td>
<td>Apr 21, 2020 - May 21, 2020</td>
<td>Mentor</td>
</tr>
<tr>
<td>SANS Amsterdam May 2020</td>
<td>Amsterdam, Netherlands</td>
<td>May 11, 2020 - May 18, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Silicon Valley - Cupertino 2020</td>
<td>Cupertino, CA</td>
<td>Jun 22, 2020 - Jun 27, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Copenhagen August 2020</td>
<td>Copenhagen, Denmark</td>
<td>Aug 24, 2020 - Aug 29, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Network Security 2020</td>
<td>Las Vegas, NV</td>
<td>Sep 20, 2020 - Sep 27, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS OnDemand</td>
<td>Online</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
<tr>
<td>SANS SelfStudy</td>
<td>Books &amp; MP3s Only</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
</tbody>
</table>