Secure Coding. 
Practical steps to defend your web apps.

Copyright SANS Institute
Author Retains Full Rights

This paper is from the SANS Software Security site. Reposting is not permitted without express written permission.

Interested in learning more?
Check out the list of upcoming events offering "Secure Coding: Developing Defensible Applications (DEV536)"
at http://software-security.sans.orghttp://software-security.sans.org/events/
Introduction

Today’s security challenges warrant a more stringent focus on users and their privileges. In the past several years, we’ve repeatedly seen users targeted in social engineering attacks that prey on their (often overabundant) privileges to both operating systems and applications. Also, attackers know that large and complex organizations don’t always have a good handle on their users and applications.

Insiders with privileges have also played a role in numerous major attacks. For example, insiders with excessive privileges contributed to inadvertent exploitation in the high-profile RSA breach in 2011. Just this August, Saudi Aramco was reported to have been infected with malware that affected production systems. In this case, an insider apparently helped the attackers access and compromise systems.1

Organizations are trying to reduce the vulnerabilities caused by excess permissions and are developing and maintaining strategies for user privileges. Many have turned to self-service provisioning tools, often part of larger Identity and Access Management (IAM) suites, to provision complex use cases for user access and auditing control. These self-services tools should use approval workflows running behind the scenes and feature some sort of self-service access portal available to employees and other approved users.

This sounds nice in theory. However, user provisioning and IAM tools have traditionally been complex products primarily leveraged by IT rather than by users or their managers. These tools have been difficult for IT to customize, and even more difficult for end users to personalize. Also, workflows between IT staff and users have been difficult to achieve and manage.

Compliance requirements such as access certification are putting new importance on provisioning and IAM, which is sparking new interest in user-enabled provisioning tools that can be monitored and documented through approved workflow processes.

Against this backdrop, we assess Oracle Identity Manager (OIM) 11g R2, an enterprise IAM product that offers end users a personalized experience through a friendly interface, while managing workflow approvals and executing changes at both the user and administrator levels.

Overall, OIM provides a range of capabilities with ease-of-use superior to many IAM toolsets available today. The ability to create simple approval workflows, with easy-to-follow wizards for creating entitlements and access requests, granular interface customization, and legacy application provisioning (so-called “disconnected” provisioning), makes Oracle Identity Manager a robust choice for user provisioning across multiple environments.

1 http://in.reuters.com/article/2012/09/07/net-us-saudi-aramco-hack-idINBRE8860CR20120907
Review Goals

The self-service provisioning concept is solid: There’s real business value in streamlining existing workflows for granting and removing the proper level of access to systems and applications. Businesses can achieve improved governance and operational overhead by properly implementing IAM solutions, as well. They also can save money in the long run by reducing user error, complex approval requirements involving numerous parties and meetings, and more.

Our goal was to determine whether key capabilities and functionality existed, and to gauge ease of use for the following functions:

- Personalization and customization of the User Interface (UI)
- Provisioning entitlements based on use cases and user profiles of varying complexity
- Creating self-service permissions and workflow to legacy systems and applications
- Following an approval workflow for a corporate-supplied mobile device, where the employee’s manager and an asset inventory manager both need to identify and approve the request

Security teams will likely spend some time defining roles and privilege sets, but most of the functionality tested in these use cases involved end users and various approvers in the scenarios’ workflows.

The review environment consisted of Oracle Identity Manager 11g R2 in a virtualized server with Oracle Unified Directory and simulated applications. These mock applications and related scenarios took the form of the following:

- A human resources application
- A financial application
- A legacy supply chain application
- A scenario for provisioning users and mobile devices
Oracle Identity Manager (OIM) allows for personalization of user interfaces and customization of interfaces managed by the administrator, such as portals and extranets that customers will be accessing. Personalization is a useful capability for organizations that want to provide access to a large variety of internal and external users without having to set each permission individually. Customization is also important for organizations that want to set up extranets and private cloud environments with self-service provisioning and different sets of applications being based on use case and approval. Organizations can use these customization features to create personalized experiences for groups of employees and partners using the system in a manner that fits the needs of their business.

**Personalization**

In this functional review, we started by modifying and personalizing the OIM interface. After successfully logging in as a user (SELLISON), we started customizing the UI from the Home screen. The Home screen is straightforward by design and includes regions or portlets for a variety of actions and information, including Pending Requests, Direct Reports, Account and Password information and several others.

To personalize the look and feel of this interface to fit our mock user scenarios, we began by clicking the Personalize link in the upper left-hand corner of the main Home screen, as shown in Figure 1.

![Figure 1. OIM Home Screen for Personalizing the User Interface](image)

This presents the entire Home screen with editing commands and functions. Each portlet is clearly labeled, and the screen has new controls available for moving, removing, or changing attributes of the portlet.
Upon starting the personalization process, we had six stock-issue portlets in place for this user:

- Pending Requests
- Direct Reports
- My Accounts
- Change Password
- Getting Started Help Topics
- Pending Approvals

To personalize the interface, we removed the Getting Started Help Topics, Direct Reports, and My Accounts portlets by clicking the X button in the top right-hand corner of their windows. Each time, OIM prompted us to make sure we really wanted to perform this action.

In this user account, we wanted to focus on the Pending Approvals portlet, so the ability to see this quickly within the interface was a priority. Moving this portlet was as simple as dragging and dropping it into the top left-hand corner, where it would be most visible upon initial login.

Putting things back is easy as well. Say we deleted the Direct Reports portlet and later changed our mind. We can add it into the interface by clicking the Add Content button, which displays a menu that asks which content we want to see, as shown in Figure 2.

![Figure 2. The Add Content Menu](image-url)
After clicking Add for the desired sections, then Close, we saved our new personalized interface and were able to see it in the new screen, as shown in Figure 3.

![New Personalized Home Screen](image-url)

**Figure 3. New Personalized Home Screen**
Customization

This system is also used by administrators developing access controls and rights around web, portal and other applications. Managers also use it to check on access requests and workflow. For access requests and workflow, administrators and managers can customize their interfaces with company logos, data sets that can be viewed and more.

In our first review of this feature, we tested the ability to customize searches for user accounts and information. Within the OIM console, we clicked the Users button at the top of the screen, which presented us with the Search Users screen. We entered a generic search for users in the Information Systems organization unit, which returned a page of results, as shown in Figure 4.

We can easily customize the results by using the individual search filters above each column in the results set. To filter for only active users within OIM, we entered the keyword Active above the Identity Status column. We can save such searches and modify the columns that appear in the results, among other things.

We also experimented with UI customization, changing the logo to the SANS Institute logo and adding text at the top of the page. In our final UI customization task, we added a User Defined Field (UDF) for a new user attribute we would need. Then we enabled the UDF within the user’s personalized interface as well—without having to write code, manipulate XML or perform other work usually involved in such endeavors.

This type of customization makes it easy to identify the user interface as a SANS employee application. Also, it can be used to push out new applications to the user that weren’t self-initiated, or change applications when needed.
Protecting User Interfaces during Changes

But how do you make changes without interrupting the end user’s access to applications through their personalized interface? To protect user applications during the customization process, customization tasks take place within a protected sandbox.

Each sandbox represents a set of changes that can be published for the organizational groups without interrupting their use of these applications or changing their personalized settings. Sandboxes focus primarily on the end-user experience, providing specific interface attributes and options that users and groups need to perform certain actions and activities for business.

The level of detail and control in these sandboxes is very granular, allowing an administrator to modify HTML tags and specific markup language when creating user sandboxes, as well as modify forms and other user-specific attributes that can be embedded.

Overall, we found the customization and personalization capabilities in OIM to be flexible and simple to learn, and because everything was browser-based, we didn’t need to do any scripting.
Self-Service Access Requests and Workflow

When it comes to self-service, user credential requests are among the most fundamental capabilities needed in access management. Supporting these processes must be workflow approvals in which sign off by business managers or administrators is allowed before changes in access are allowed. Such approvals should be as automated as possible.

In this case, we wanted to see if OIM would follow the chain of command in a human resources self-service application in which an employee requests access to a timecard application and gets his manager’s approval before the requested application privileges are made available.

User Perspective

The self-service features, including searches for HR applications and approval from the user’s manager, worked seamlessly from the user perspective. We started by searching applications available to our new employee user. Then we searched for employee timecards.

Entering keywords turned up two timecard applications to choose from, as shown in Figure 5.

![Figure 5. Self-Help Resources Available for User to Check Out](image_url)

Still emulating the employee user, we clicked the timecard application we wanted, which placed it into a shopping cart, and finalized the selection through a Checkout capability. The system sent immediate feedback to the employee that the request had been submitted to his manager and was waiting approval. If the user is not allowed to access or request a specific entitlement, this is immediately made clear to him through a popup message that explains why.
Manager Perspective

Once the application request was made, the system sent an e-mail to the employee’s manager who went to her portal, evaluated the request, and processed it through her own interface. In this case, the manager approved the request, as shown in Figure 6.

![Figure 6. Manager Approval for a Request](image)
Approved or Denied

When the user checked his entitlements, the new timecard application was there among his resources, as shown in Figure 7.

![Figure 7. Timecard Application Entitlement](image)

Access Requests and Regulated Data

The second type of entitlement provisioning in our mock environment was a bit more involved. This entitlement was for an expense reporting application that falls under Sarbanes-Oxley (SOX) compliance rules.

In this scenario, various risk levels for different employees and use cases were involved, and several users had requested more complex review or approval entitlements. So this workflow included users making the request, their managers processing requests, and a separate audit staff finalizing the requests, as shown in Figure 8.

![Figure 8. Privacy-Protected Expense Entitlements Requested by Employee](image)

Each entitlement is assigned its specific risk level, and that risk level is used to determine the approval workflow. For expense report viewing, a medium risk is assigned, so either the assignees’ manager or an Audit Review team could approve the request. If this were a healthcare environment, that risk assignment would surely be high and could call for multiple approvals before being granted.
Approvals and Workflow

Approval workflows can run in parallel or serial form. Parallel workflows have multiple parties who can sign off on the request, and any of them can act first and officially approve the request. Serial workflows require approvals to occur in order, where the first party approves the request and then sends it to the next party in line.

Once the manager approved this parallel request, some of the associated requests also advanced in the workflow. After additional approvals in the workflow, primarily from the Audit Review team, the specified user was granted access to all requested entitlements, as shown in Figure 9.

![Figure 9. User Jarvis Galas’ Entitlements after Approval](image)

The interface also provides workflow visualization and the ability to provide comments and send attachments. As demonstrated in this review, entitlements are easy to change as required by business needs; for example, when someone needed access to more-protected resources than normal, we were able to demonstrate additional approval points along the workflow.

From the user standpoint, OIM’s Shopping Cart request tool made complex requests easy to understand by users, their managers and auditors. The Shopping Cart tool provides users with a view of what requests they have in the works and what approvals are required at any time during the approval process.
Providing self-service access to legacy systems has been problematic for organizations because these systems cannot be fully integrated into the organization’s identity management solution. Often they have to be handled manually. Such systems can be thought of as disconnected resources.

In lieu of automated workflow, IT groups often resort to using trouble tickets and other manual means to change access to these disconnected systems. Rather than taking a day to get approval for new access with an automated process, users could end up waiting weeks for approvals using a manual approach!

Even though these applications aren’t connected to the access control system, requests and approvals still need to be centrally controlled. Oracle Identity Manager’s workflow capabilities carry over to these orphan applications with a disconnected workbench that can be leveraged to grant access to these systems without the need to build and maintain connectors to them.

To review this feature in OIM, we first needed to create a new sandbox or customized user interface environment. Using OIM Form Manager, we requested access to a mock supply chain management (SCM) application that we used as the legacy system.

Once the sandbox was created, we set up a new application instance called Legacy SCM, set it as disconnected, and saved it, as shown in Figure 10.

![Figure 10. Legacy SCM Sandbox App Instance](image_url)
Having set up this instance, which identified and linked us to the legacy application system, we needed to create a custom form for OIM that listed the attributes we wanted to control and maintain. Using the OIM Form Editing Wizard, we were able to create a relatively simple form that included the following information:

- Application User ID
- User Type (in this case, the Business Unit relationship)
- Work Location

One of the Form Creation Wizard’s pages from our example appears in Figure 11.
Once this was completed, we assigned the new sandbox and application to the Information Systems group. The next task was to create new catalog items for the various groups that would need to be involved in the self-service access request workflows. The first step was provisioning the Fulfillment Role to the Asset Management Team to make that manager responsible for approving access to the SCM (see Figure 12).

![Figure 12. Establishing the Access Request Fulfillment Role for a Legacy App](image-url)
At this point, users can begin requesting access to the Legacy SCM as if it were any other item in OIM’s catalog. These new access requests will contain the form we created, as shown in Figure 13.

![Figure 13. New User Catalog Request for Legacy SCM](image)

In this case, workflow management works the way it does with any other provisioning case. It sends the request to the user’s manager, who can then go to his portal and deny or approve it.

Once the management approval has been completed and forwarded in the workflow, the Asset Management team needs to provision the user within the system. In our example, this is a manual task that someone in that group must claim and manage because, as with other legacy apps, this application lacks the necessary Application Programming Interfaces (APIs) to enable integration. Even with APIs, there are often manual tasks involved with provisioning legacy systems. The benefit of this system is that there is no need to build the connectors. Although manually executed tasks are less than ideal for true end-to-end management, Oracle Identity Manager presents a huge improvement in automation of workflow requests, approvals and other auditable activities.
Once our Asset Management team members claimed the task and completed the access request, they marked the request Complete, as shown in Figure 14.

![Image of Asset Manager Completing a Manual Task for Provisioning](image_url)

**Figure 14. Asset Manager Completing a Manual Task for Provisioning**

The user who initially requested access to the legacy application will then receive a notification that the system access provisioning is complete.

We experimented with several of these disconnected legacy applications, using various user roles and workflow scenarios. As long as the appropriate groups and roles were set up, creating workflows for these applications was relatively simple and easy to amend.

The Form Manager was an incredibly convenient feature that saved quite a bit of time for customizing the specific user data input into the entitlement request and would simplify provisioning for much more complex legacy systems. It is also much faster than trying to perform more in-depth integration activities via APIs. Every time a user requests access in this scenario, she simply needs to enter any required data for the access request into additional form fields defined in the sandbox.
The final scenario we worked through simulated a new user to the organization requesting a corporate-supplied mobile device and application access.

Our previous entitlements focused on software, some of which were fully integrated with Oracle Identity Manager (as connected requests) and some that weren't (disconnected requests). In many ways, requesting a device is no different from requesting any other disconnected asset or application, such as the legacy SCM application in the previous scenario.

In this case, we just reviewed the workflow request rather than create our own workflow and groups. In our example, illustrated in Figure 15, user Brian Jones has requested application access through a new company-issued Apple iPhone 4S.

---

Figure 15. Manager View: Request for Mobile Access Request from new iPhone
Once again, the user’s manager will need to approve the request, and the Asset Management team will need to claim and then complete the request. While this is happening, the user sees the request Provisioning, as shown in Figure 16.

![Figure 16. User View: Pending Provisioning Request](image)

After the Asset Management group completed its role in the provisioning workflow, the request changed to the Provisioned state in the status column of the employee UI.

As with the other requests we reviewed, OIM maintained an audit trail of all these activities for later review by auditors.

Because OIM emulates a workflow methodology common to most organizations, requests are completed in the same time it takes to complete access requests to non-legacy systems. To the end users, the process appears no different, and to the administrators it is more efficient and centralized than other options available to them.
Conclusion

In large, complex organizations, user-driven or self-service application provisioning has historically been difficult to deploy into an existing matrix of access management policies.

Oracle Identity Manager 11g R2 is a broadly capable tool for enabling and centralizing self-service identity and access provisioning and entitlement. The system is easily modified and customized, giving users the ability to create highly personalized interfaces and giving administrators and managers the ability to customize their user experiences as well.

With OIM, workflow approval processes are automated, and minimal manual intervention is needed to grant access requests to disconnected legacy systems. You don’t need to be a development expert to use OIM, nor do you need to build any APIs or connectors.

When it comes to self-service, the first feature organizations want is ease of use. In pursuit of this quest, OIM presents entitlement provisioning to end users through something they are already familiar and comfortable with: a self-service shopping cart interface.

This ease of use is the biggest value OIM brings to the enterprise. The tool presents familiar and straightforward interfaces easily personalized by end users and customized by administrators without the need for coding and development. End users can easily search for entitlements, request them by adding them to the cart, and submit the request by “checking out.” Administrators can easily make changes to user rights and interfaces in the safety of sandboxes. Once those changes are accepted, they show up within the user interface without the need for any other applications or coding.

This is also true with disconnected legacy applications. The ability to create and publish custom user interfaces, with specific customization attributes and capabilities for different employees or user roles, makes it possible to provide self-provisioning to legacy and mobile systems. These are historically problematic to integrate into provisioning solutions without developing or building and integrating (with coding) connectors to the legacy applications. The ability to sandbox UIs before they’re generally deployed makes it possible to test UI customizations without fear of affecting the entire business. Oracle Identity Manager is a good fit for complex organizations looking to simplify their self-service access programs. It is also a good tool to start your self-service access management program on the right foot, if you haven’t already started one.
About the Author

Dave Shackleford, founder and principal consultant with Voodoo Security, is a SANS analyst, instructor and course author, as well as a GIAC technical director. He has consulted with hundreds of organizations in the areas of security, regulatory compliance, and network architecture and engineering. He is a VMware vExpert, and has extensive experience designing and configuring secure virtualized infrastructures. He has previously worked as CSO for Configuresoft and CTO for the Center for Internet Security. Dave is the co-author of “Hands-On Information Security” from Course Technology as well as the “Managing Incident Response” chapter in the Course Technology book “Readings and Cases in the Management of Information Security.” Recently, Dave co-authored the first published course on virtualization security for the SANS Institute. Dave currently serves on the board of directors at the Technology Association of Georgia’s Information Security Society and the SANS Technology Institute.

SANS would like to thank its sponsor:

ORACLE
# Upcoming SANS App Sec Training

<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Dates</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS 2015</td>
<td>Orlando, FL</td>
<td>Apr 11, 2015 - Apr 18, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Security West 2015</td>
<td>San Diego, CA</td>
<td>May 03, 2015 - May 12, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Secure Europe 2015</td>
<td>Amsterdam, Netherlands</td>
<td>May 05, 2015 - May 25, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>Community SANS Seattle DEV522</td>
<td>Seattle, WA</td>
<td>Jun 08, 2015 - Jun 13, 2015</td>
<td>Community SANS</td>
</tr>
<tr>
<td>Community SANS Salt Lake City DEV541</td>
<td>Salt Lake City, UT</td>
<td>Jun 15, 2015 - Jun 18, 2015</td>
<td>Community SANS</td>
</tr>
<tr>
<td>SANS Capital City 2015</td>
<td>Washington, DC</td>
<td>Jul 06, 2015 - Jul 11, 2015</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS London in the Summer</td>
<td>London, United Kingdom</td>
<td>Jul 13, 2015 - Jul 18, 2015</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>

Last Updated: March 27th, 2015