ORACLE FUSION MIDDLEWARE 11g
IDENTITY MANAGEMENT

# Oracle Privileged Account Manager

## Protecting and Auditing Access to Sensitive Resources

ORACLE WHITE PAPER | APRIL 2015

ORACLE

## Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

## Table of Contents

## Introduction

The seemingly endless stream of highly visible security breaches and public disclosure of classified information at major organizations and corporations conspicuously exposed the existing problems with privileged user management.

Privileged users perform sensitive activities that involve access to strategic corporate and federal assets. In most organizations, privileged accounts are not clearly defined and are often shared by multiple users.

Privileged accounts are evident across a multitude of critical Information Technology (IT) resources such as operating systems (e.g., UNIX, Windows), database servers (e.g., Oracle Database, Microsoft SQL Server, or IBM DB2), user directories (e.g., Oracle Directory Services or Microsoft Windows Active Directory Services), network devices (routers, load balancers, firewalls), Hypervisors and enterprise applications (e.g., human capital management, supply chain management). Privileged users include system, database, and network administrators, support personnel (e.g., help desk), as well as application owners.

Detecting inappropriate access to these accounts and determining which individuals in a team of administrators participated in unauthorized activities is extremely challenging, because privileged accounts are not necessarily tied to individual end users. The number of privileged accounts grows with the number of servers, devices, and applications to manage. In most large organizations, there are hundreds, sometimes thousands of privileged accounts for which multiple individuals know the credentials without a way to track who actually used an account at a specific time. Because an inflation of privileged accounts is hard to manage, passwords are rarely changed thus compromising overall security and violating compliance regulations.

This document describes how Oracle Privileged Account Manager addresses the challenges expressed above. Oracle Privileged Account Manager is designed to enable the separation of privileges, manage self-service requests to privileged accounts, and provide auditing and reporting of password usage. As an integral part of the Oracle Identity Governance Suite, Oracle Privileged Account Manager contributes to securing Global Trade Management (GTM) strategies in particular as they relate to compliance with regulations such as the Sarbanes-Oxley Act.

## Introducing Oracle Privileged Account Manager

Oracle Privileged Account Manager (OPAM) is a server-based password vault designed to generate and manage passwords and sessions for privileged users accessing specific resources. Through secure session management, control and reporting, OPAM enables historical records to support forensic analysis and auditing.

OPAM is an Oracle Fusion Middleware application running on Oracle's WebLogic Server leveraging the Oracle Platform Security Services (OPSS) framework as a security foundation and Oracle Database as backend data storage. OPAM is integral part of Oracle's Identity Governance Suite, which provides user provisioning and de-provisioning, user self-service (such as delegation, access request, and password reset), approval workflow, risk-based, business-user-friendly identity certification, and advanced role life cycle management. In order to continuously monitor and effectively enforce compliance controls, Oracle Identity Governance provides automated periodic review of users' access rights (together with closed-loop remediation of these access rights) as well as monitoring of exceptions to corporate audit policies.

Oracle Identity Governance platform also provides rich audit and reporting capabilities allowing lines of business, IT administrators, and auditors to not only review who has/had access to what, but also how users acquired access and when they used an exclusive privileged account. In addition the OPAM dashboard provides a real time status of ongoing user activites.

Oracle Identity Governance provides automated provisioning to managed applications and target systems upon grant for both standard and privileged access using a robust set of connectors between OPAM and managed systems and resources. If these grants need to be revoked as a result of monitoring controls, automatic de-provisioning occurs using the same connectors and a comprehensive audit trail is provided.

In addition to actively managing privileged accounts, OPAM provides the concept of a "lockbox". Assuming accounts are associated with target systems located in a network subnet which cannot be reached from OPAM, the OPAM security administrator can decide to create a OPAM "lockbox" to centrally store and securely grant access to privileged account passwords, thus preventing the requirement for administrators to note down these accounts.

### OPAM Architecture

The OPAM server, a Java EE application, securely stores the passwords (and metadata) it manages within an Oracle database (Figure 1). OPAM resets the passwords on the target systems as required, through connectors during the checkout and/or check in process. OPAM leverages common Oracle Identity Governance connectors, thus reducing administrative overhead and total cost of ownership.

Based on Oracle's Integrated Connector Framework (ICF), connectors link OPAM to the external stores of target enterprise applications, user directories, database servers, operating systems, hypervisors and network devices and enables OPAM's discovery of privileged accounts and password management features.

OPAM ships with a set of connectors for UNIX, LDAP, Database, Windows, SSH based targets as well as SAP. These connectors enable OPAM to manage a broad range of target systems: any UNIX / Linux server, Oracle Database (from Oracle Database 9 to Oracle Database 11g), Microsoft SQL Server, SAP, Sybase Adaptive Server Enterprise 15.x, IBM DB2, LDAP directories including Microsoft Active Directory, Network Devices accessible through SSH or managed through RADIUS, and Hypervisors. In addition OPAM is certified with Oracle Exadata and Oracle Exalogic engineered systems.

Privileged accounts are discovered and identified through connectors. A connector queries the target systems for a list of accounts on these systems. The OPAM administrators then determine which of these accounts they want to manage as privileged accounts. In general OPAM can manage any account that can be discovered on a target.

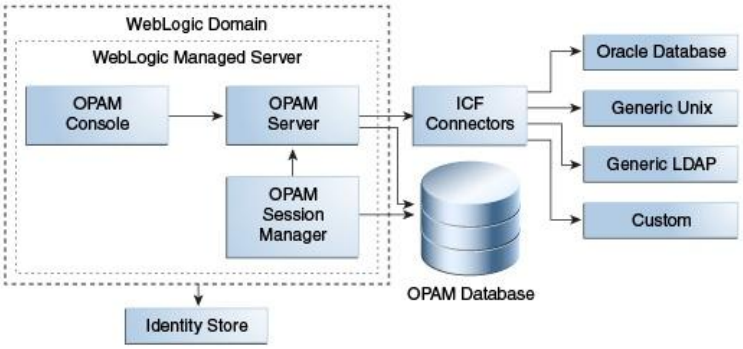


Figure 1: Oracle Privileged Account Manager Architecture

The OPAM server exposes a RESTful API (REST is a lightweight HTTP-based alternative to SOAP-based web services). The RESTful API is used by OPAM client applications including the OPAM console and the OPAM command-line client. OPAM's RESTful API is secured over SSL and requires HTTP authentication via username / password or certificate based authentication. Customers and Service Integrators can build custom integrations using OPAM's public (and fully documented) RESTful API. In addition OPAM supports OPSS Trust for Identity Propagation, allowing the OPAM console to make REST API invocations against the backend OPAM server on behalf of an end user authenticated against the OPAM console.

Metadata information and passwords managed by OPAM are persisted in an Oracle Database. Customers can leverage Oracle Database Vault and Transparent Data Encryption (TDE) for additional security. The use of TDE is a recommended best practice by Oracle (OPAM ships with a restricted use license for TDE). More details about TDE can be found on Oracle Technology Network (OTN) under Database Advanced Security.

## OPAM Process Flow

OPAM supports privileged account consumption using a password or a session. OPAM allows a user, for example a database administrator, to use a privileged account by "checking out" a password for a particular enterprise application, operating system, or database server. Once OPAM successfully resets the account password, a password is issued to the administrator. The administrator uses the password, then "checks in" the password when the administrative task is complete. As a standard behavior (default configurable policy), the system is set to automatically change the password on check-in, thereby precluding the administrator from reusing the same password again.

**Note:** When requesting a session the flow is slightly different see OPAM session management below.

OPAM Password Check Out Process

A password can be checked out using the OPAM console (as shown in Figure 3), via OPAM command line, or using a custom client application leveraging the OPAM RESTful API. For instance, an administrator is requesting a password for the "HR_ADMIN" user to access the HR application database (the same process would apply to any other type of target system managed by OPAM).

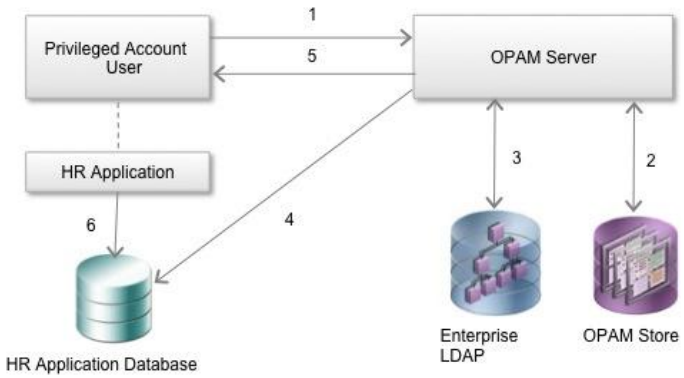


Figure 2: Simplified OPAM Check-Out Process

1. The administrator authenticates to the OPAM server and requests the HR_ADMIN password.
2. The OPAM server checks that the requester has been granted the HR_ADMIN account directly or through group membership (HR application administrators are in a group and the HR_ADMIN account is granted to that group).
3. The enterprise LDAP server (identity store) verifies the requester's identity and role.
4. The OPAM server generates a password based on the specific password policy for the account and resets the password for the HR_ADMIN account in the database.

   **Note:** multiple password policies can be created and used
5. The OPAM server returns the password to the requester.
6. The requester logs into the HR application database as an administrator with the OPAM-generated HR_ADMIN password.

Once the task is complete, the administrator checks the password back into OPAM. OPAM is configured to automatically change the password, thereby eliminating the possibility of the password being reused (default behavior). Additionally OPAM provides a configurable password history, which is required if for example you run a backup as a particular user and you need to retrieve the user password for that point in time.

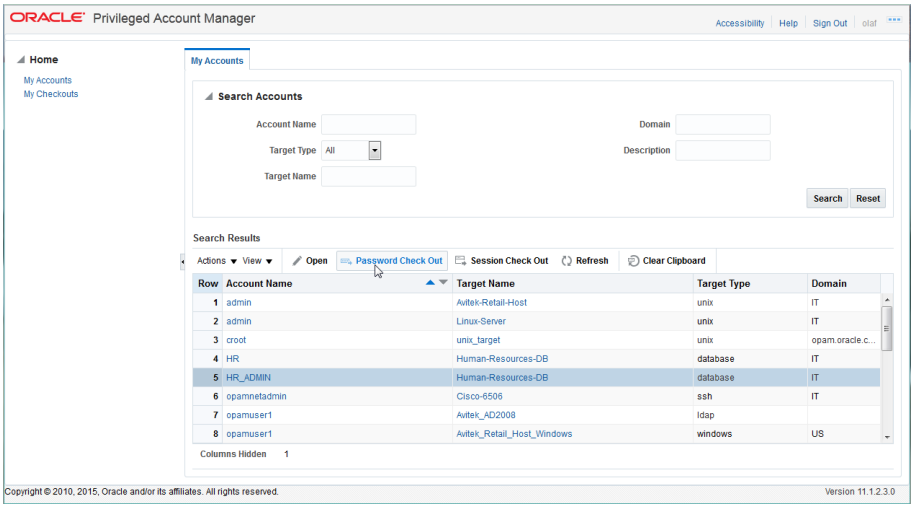


Figure 3: Checking out the HR_ADMIN password using the OPAM Console

## OPAM Application Account Management

Service or application accounts are another type of accounts OPAM can manage. Applications use these accounts to connect to target systems at run time. Traditionally, administrators set up these accounts once during installation and never use them again. As a result, application accounts can potentially cause hidden vulnerabilities in your deployment. For example, passwords might become less secure over time because they were created using outdated policies, or commonly used deployment passwords might be compromised.

Service or application passwords are more challenging to manage because not only does the password need to be changed in the target system, the client (application) needs to be notified as well. Applications built with Oracle Fusion Middleware (for example Oracle Fusion Applications) by default use Oracle Platform Security Service's (OPSS) credential store framework (CSF) to externalize passwords. OPAM can update the passwords stored in the credential store directly without the requirement to modify the client application. The password is changed transparently to the application, based on corporate password policies. The application will always checkout/ access the password without having to deal with password lifecycle management.

For applications which do not leverage the OPSS credential store framework, the OPAM plug-in framework can be used as an alternative to synchronize passwords into application specific external storage containers. This can be achieved through a plug-in with custom logic that executes on account password change operations and pushes the new password for example into a properties file. Additionally, a password policy can be applied for periodically cycling the account password. Cycling the password ensures that the application accounts are always compliant with the latest corporate policies and remain secure. OPAM performs this task with no service interruption.

For more information on application account passwords see the blog "Oracle Privileged Account Manager".

## OPAM Session Management

Recent front-page security breaches have emphasized the fact that access control and monitoring of privileged accounts is critical. In some cases, privileged account password management alone is not enough. OPAM additionally provides session management and auditing capabilities to address extreme use cases. By creating a single access point to the target resources, OPAM's Privileged Session Manager (OPSM see Figure 4) helps administrators to control and monitor all the activities within a privileged session.

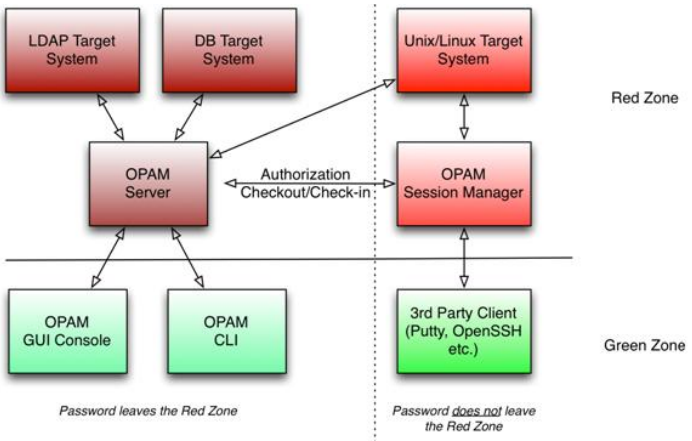


Figure 4: OPAM Privileged Session Manager Deployment Architecture

### Session Initiation

Through OPSM, the end user gains access to the targets. If access is granted in OPAM, OPSM acts like a gateway and can elevate the access level on the target for the user. For example, user "Joe" can access a Linux system as user "root." During session initiation the privileged credentials, for example "root" password, will never be revealed to the end user. Before a session is initiated, a real time access approval can be implemented using the plug-in framework, for example to enable emergency access to a system when an administrator is not available. Additionally when a session access is granted, a notification (text message or email) can be sent to an auditor. Currently any compliant third-party client (e.g. Putty, OpenSSH, etc.) is supported. Support for X11, VNC and other protocols are planned for future releases.

### Session Control and Command Control

Sessions are controlled through usage policies that define how long a session will be open before being terminated. Users can request a session that won't be terminated, for example for a long-running backup job, however OPAM administrators can force a session termination at any time.

Through elevated access control, users can be limited in terms of commands they're allowed to execute. Command restrictions are enforced through OPSM using a "white list" based approach, which means a user can only execute commands from a predefined list of commands.

No "agent" on the target system is required when OPSM is used, thus eliminating the lifecycle management for an additional software component on the target system.

### Session Monitoring and Auditing

When the session is established, OPSM will monitor SSH session activities through keystroke logging and record the input / output for each session into searchable historical records (transcripts) to support forensic analysis and audit data (see Figure 5).

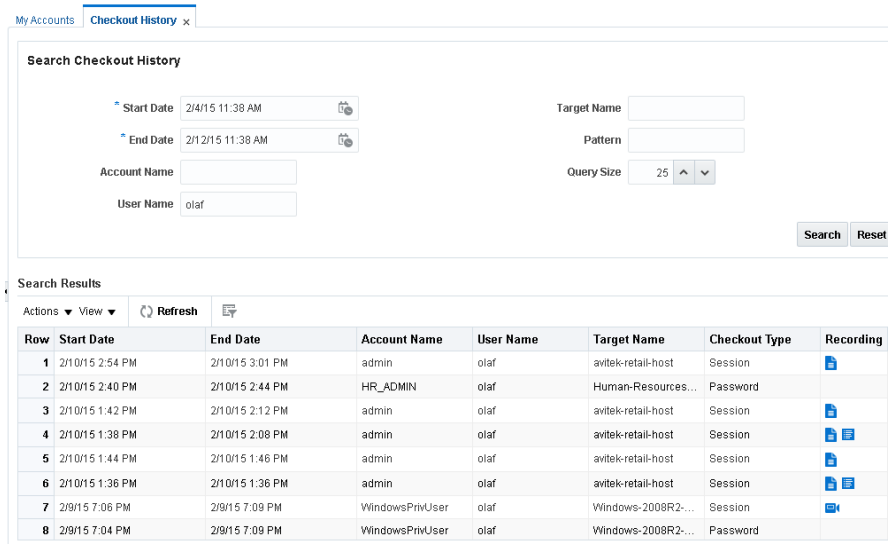


Figure 5: OPAM checkout history

The SSH session recording provides individual accountability for all user-initiated actions even if for instance a shared account is used simultaneously by more than one user.

### Windows Session Recording and Auditing

For Windows based target systems OPAM leverages an OPAM agent on the target to capture and record user activities into a MPEG-4 encoded video (Figure 6). This video can be reviewed in a standard HTML5 browser. Similar to a DVD or DVR you can move back and forth in time or directly jump to specific events occurred.

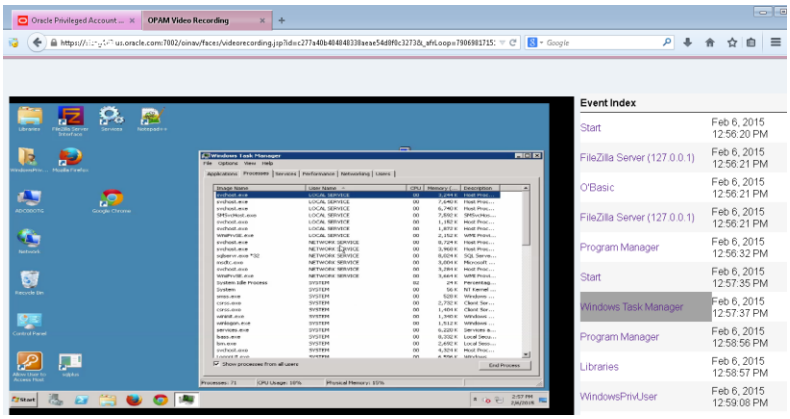


Figure 6: OPAM Windows recording

## OPAM Audit and Reporting

Auditing in the context of OPAM is the ability to determine who checked out privileged account passwords and/or sessions (as shown in Figure 5). In addition to password check-outs, OPAM audits and logs all operations and

provides its own built-in audit reports. In fact all OPAM operations are stored in an audit database. The OPAM dashboard (Figure 7) provides a realtime view of accounts currently in use and active sessions.

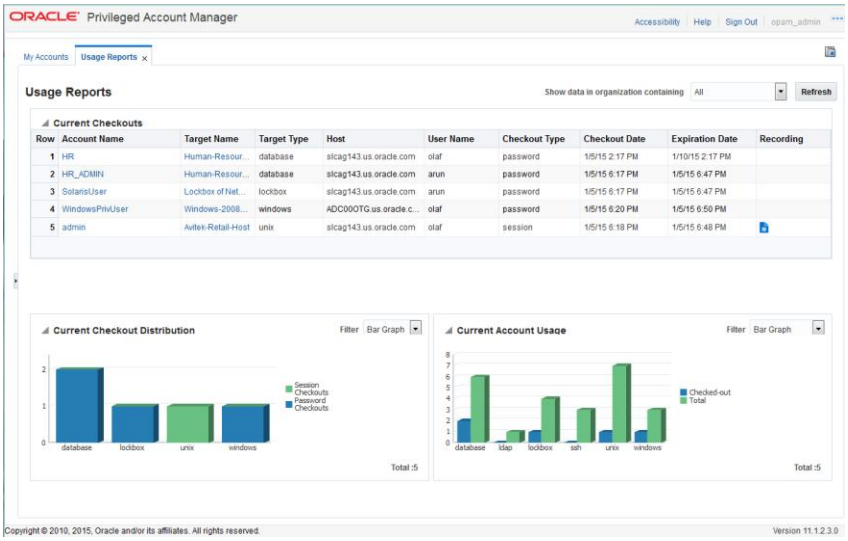


Figure 7: OPAM dashboard

OPAM's auditing and reporting can be combined with other Oracle Fusion Middleware components such as the Oracle Fusion Middleware audit service and Oracle Business Intelligence (BI) Publisher to generate customized reports (Figure 8).

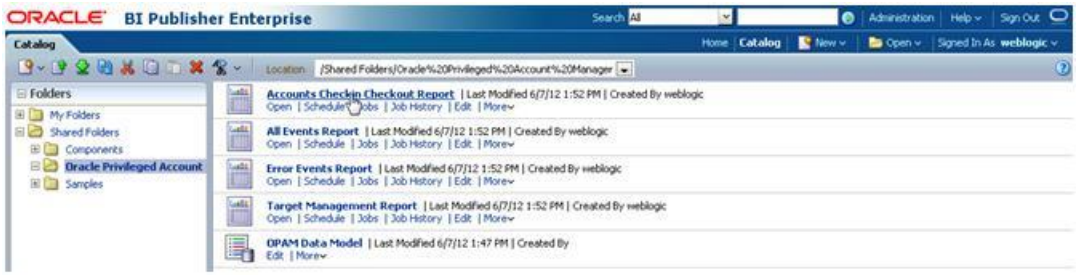


Figure 8: OPAM Audit and Reporting with Oracle BI Publisher Enterprise

## OPAM Resource Groups and Delegated Administration

In Oracle Privileged Account Manager, all targets and accounts are considered resources. A resource group is a collection of resources that can include targets, accounts, and other resource groups (Figure 9). Resource groups facilitate easier and better administration of resources in your deployment and are particular helpful when managing thousands of systems in different geographies, or Cloud deployments.

Resource groups simplify management by organizing data into groups and delegating administration to specific users or user groups. In Oracle Privileged Account Manager a user with a global administrative role such as Security Administrator role has administrative access to all resources, i.e. all targets and accounts. Deployment needs

necessitate administrative access to be provided for users to a subset of resources rather than a global access. For example a regional admin may need access to manage only the resources within his region. Resource groups provide the mechanism to create such sub sets of resources and delegate administration privileges for that resource group to specific users, user groups, or both.

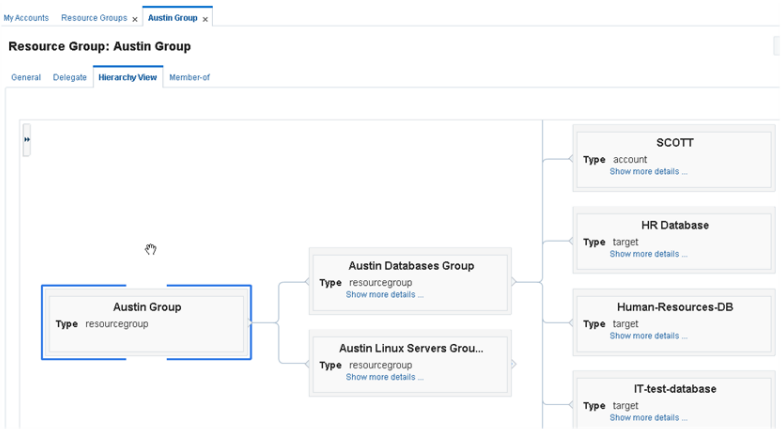


Figure 9: OPAM Resource Group Viewer

## OPAM Extensibility and Customization Framework

OPAM provides a Java based plug-in framework that enables you to extend and customize OPAM functionality and operations to better meet your specific requirements. This framework enables you to:

- validate and manipulate data before OPAM performs operations for instance provide real time emergency approval via phone
- perform specific actions after OPAM completes its operations, for example send notifications when an account has been checked out
- register and manage plug-ins through the OPAM console, command line, or RESTful interface
- integrate OPAM with third-party systems such as wallets, ticket management systems, and audit systems

## OPAM and Identity Governance

Although OPAM may run as a standalone component, the real value of any privileged account management (PAM) system is its synergy with the enterprise's identity management environment. Indeed, one of OPAM's key benefits is its integration into the Oracle Identity Governance Suite. This integration provides you with a complete governance solution to support both regular and privileged users in order to meet enterprise compliance requirements such as certification and auditing users' access to individual and shared accounts. This unique benefit delivers a single user experience for both standard and privileged access, and a common certification and reporting process.

Following are typical use cases demonstrating the benefits of the close integration of OPAM with the Oracle Identity Governance platform.

- **Provisioning users to the OPAM directory through the Identity Governance Suite's provisioning capability:** Leveraging the Identity Governance policy- and role-based provisioning, a system administrator may be provisioned to the specific LDAP groups that OPAM uses for privileged account access with full support for approval workflow.

- **Requesting privileged account access through the Identity Governance platform:** Account access is typically governed by LDAP. The best practice with OPAM is to tie a privileged account access to an LDAP group membership. This enables any existing LDAP group membership process such as provisioning to manage access to OPAM. In this use case, an OPAM user needs to request access to an account for which he is not currently approved.

- **The Identity Governance platform publishes privileged account entitlements in its request catalog**: (see Figure 9). An administrator uses the Identity Governance platform's access request self-service, searches the catalog, picks the group he needs, and submits the request for approval. The user is provisioned with group membership after approval, and can now access OPAM for checking out a privileged password. This is a pre-integrated option, however any other mechanism for requesting access and adding users to the proper LDAP group(s) can alternatively be used.


Figure 9: Identity Governance Platform's Request Catalog

- **Requesting "break-glass" access through the Identity Governance platform:** Break-glass access enables administrators to request emergency access to privileged accounts they are not normally entitled to (the break-glass metaphor comes from breaking the glass to pull a fire alarm). Such a situation may happen when a critical server is down and the designated server administrator is not available. In this case, the administrator goes through the Identity Governance platform's request process indicating this is a break-glass emergency request. Submission of the request kicks off a break-glass workflow with minimal or automatic approval (based on the customer's processes and policies). The administrator is provisioned to the OPAM LDAP group and can access privileged credentials. A special alert is generated and audited for the event and is sent to security administrators. The access is automatically de-provisioned based on the security policies defined by the customer.

- **Delegating access to privileged accounts:** Leveraging the Identity Governance platform's self-service delegation functionality, a privileged user leaving work for a period of time can delegate her entitlements to another user during her absence and reclaim her entitlements upon her return within the organization's policy guidelines.

- **Leveraging risk-based certification and closed-loop remediation:** Taking advantage standard Identity Governance platform functionality, privileged access information is made available for access certification. Risks can be calculated based on the privileged access status and other data such as the provisioning method. If access violation is detected, access can be revoked leveraging closed-loop remediation, a core feature of the Identity Governance platform.

Additionally, OPAM benefits from Oracle's Access Management platform to simplify access to the OPAM console and fraud detection.

The Oracle Access Management platform is a server-based solution that provides centralized, policy-driven services for web application authentication, web single sign-on, identity assertion, and session management. The Oracle Access Management platform also provides resource protection through real-time fraud prevention, software-based multifactor authentication, and unique authentication strengthening. OPAM leverages the Oracle Access Management platform for single sign-on and layered access control to the OPAM console.

## OPAM and Complementary Security Technologies

OPAM works in conjunction with other security components such as Oracle Enterprise User Security, Oracle Database Vault, and UNIX / Linux user management to provide a full-fledged security solution supporting both regular and privileged users. The following sections describe how OPAM complements these technologies.

### OPAM and Oracle Database Enterprise User Security

Enterprise User Security (EUS) is a feature of Oracle Database that enables administrators to centrally manage database users across the enterprise. Enterprise users are created in an LDAP directory and are assigned roles and privileges across various enterprise databases registered with one or more user directories.

One major differentiator of Oracle's identity management offering is the ability to provide customers with greater flexibility and choice of user directories by integrating EUS with Oracle Unified Directory (OUD). OUD's virtualiation capabilities enables organizations to centrally manage database-user identities through existing corporate directories such as Oracle Internet Directory (OID), Oracle Unified Directory (OUD), Oracle Directory Services Enterprise Edition (ODSEE), and third-party directory services such as Microsoft Active Directory (AD).

OPAM provides services that are complementary to EUS as shown in Table 1:

| Feature Description | Supported By |
|---|---|
| Use existing enterprise LDAP passwords for end-user passwords | EUS |
| Map database usernames to LDAP users and database roles to LDAP groups | EUS |
| Manage SYS / SYSTEM passwords | OPAM |
| Manage passwords for privileged application accounts | OPAM |
| Manage non-Oracle database administrators passwords | OPAM |

Table 1: Oracle Enterprise User Security and OPAM

## OPAM and Oracle Database Vault

Oracle Database (DB) Vault is designed to enforce separation of duties for Oracle DB administrators, least privilege, and other preventive controls to ensure data integrity and data privacy. Oracle DB Vault proactively protects application data stored in Oracle DB from being inappropriately accessed by privileged database users.

OPAM provides services that are complementary to Oracle DB Vault as shown in Table 2:

| Feature Description | Supported By |
|---|---|
| Privileged user access control to limit access to application data | DB Vault |
| Multi-factor authorization for enforcing enterprise security policies | DB Vault |
| Secure application consolidation | DB Vault |
| Manage DB Vault privileged accounts passwords (e.g., SEC_ADMIN) | OPAM |
| Manage database privileged accounts passwords (e.g., SYS) | OPAM |

Table 1: Oracle Database Vault and OPAM

## OPAM and UNIX / Linux User Management

Oracle Authentication Services for Operating Systems (OAS4OS) enables enterprises to centralize the management of UNIX and Linux authentication, user accounts, password policies, and sudo (UNIX “superuser”) authorization policies leveraging Oracle Internet Directory (OID). Based on open, standard interfaces, OAS4OS provides full automation of client configuration and user migration, and serves as an LDAP-based naming server to replace the (now deprecated) Network Information System (NIS) naming service. (Note: Without deploying OAS4OS, a centralized UNIX / Linux environment can also be created using other Oracle directory services such as OUD and ODSEE.)

OPAM provides services that are complementary to OAS4OS as shown in Table 3:

| Feature Description | Supported By |
|---|---|
| Use existing enterprise LDAP for end-user passwords | OAS4OS |
| Map UNIX groups and NIS maps to LDAP | OAS4OS |
| Manage ROOT passwords | OPAM |
| Manage privileged application accounts | OPAM |
| Manage Microsoft Active Directory passwords (i.e., Microsoft Windows machines authenticating with an AD domain) | OPAM |

Table 3: Oracle Authentication Services for OS and OPAM

## Conclusion

Organizations are struggling to manage a large number of administrative accounts in a secure, efficient, and scalable way. Privileged accounts are the most critical accounts to manage since they provide broad access to systems and sensitive corporate and state information. Failure to manage privileged accounts can result in data breaches, theft, compliance violations, and service outages. By creating a single access point to target resources, Oracle Privileged Session Manager helps administrators easily control and monitor all activities within a privileged session.

Oracle Privileged Account Manager (OPAM) provides a versatile and integrated solution that allows organizations to manage their privileged account passwords. OPAM can operate as a standalone solution, but delivers more value when used in conjunction with other Oracle Identity Management components. As part of the Oracle Identity Governance platform, OPAM provides secure, self-service access to database servers, operating systems, and enterprise applications accounts with full auditing and reporting capabilities.

For more information on OPAM or other Oracle Identity Management solutions please visit Oracle's website at http://www.oracle.com/identity.

**Oracle Corporation, World Headquarters**
500 Oracle Parkway
Redwood Shores, CA 94065, USA

**Worldwide Inquiries**
Phone: +1.650.506.7000
Fax: +1.650.506.7200

ORACLE

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

**Hardware and Software, Engineered to Work Together**



Oracle Privileged Account Manager
Protecting and Auditing Access to Sensitive Resources
April 2015
Author: Olaf.Stullich@oracle.com
Contributing Authors: Marc Chanliau, Buddhika Kottahachchi, Arun Theebaprakasam