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/
Database Activity Monitoring and Audit: A Review of Oracle Audit Vault and Database Firewall

January 2014
A SANS Analyst Product Review

Written by Tanya Baccam

The Benefits PAGE 2
Database Activity Monitoring PAGE 5
Centralized Reporting for Forensics and Audit PAGE 11
Introduction

Databases are at the heart of the datacenter and contain organizations’ most sensitive and regulated information; thus, they are prime targets for today’s advanced threats. They require continuous monitoring and layered protection to prevent the loss of sensitive customer, financial and other business data.

Oracle has, over the past few years, continued to grow and integrate security capabilities both for Oracle and non-Oracle database products. The company’s tools go well beyond securing the Oracle database and can monitor and protect non-Oracle databases such as Microsoft SQL Server, IBM DB2 and Sybase; Linux, Solaris and Windows operating systems; or even directory services, filesystems and custom applications.

Oracle recently merged its database activity monitoring, auditing and reporting tools into a comprehensive heterogeneous enterprise data security platform, Oracle Audit Vault and Database Firewall (AVDF). So now, with one tool, organizations can monitor their databases for suspicious activity, block threats and provide audit reports across the enterprise.

In this review, we evaluated a functional subset of AVDF under simulated real-world scenarios. Our review concluded that AVDF did what it claimed to do and is a valuable solution for organizations looking for a first line of defense that protects their data and databases. The functionality of AVDF became apparent through the review as we looked at audit trails and reports and tested SQL injections. AVDF comes with a variety of useful audit reports based on multiple regulations and standards, and it can monitor for malicious activity out of the box.

At a High Level

Oracle Audit Vault and Database Firewall (AVDF) provides organizations with three key database security requirements: audit collection, SQL traffic monitoring and security event reporting. With an easy-to-use interface, AVDF is for organizations looking to increase security with enterprisewide database activity monitoring, auditing and reporting.
AVDF provides a holistic approach to database protection, threat management, forensics and auditing. By collecting and consolidating audit data, firewall traffic and logs generated by databases, operating systems, directories and more, AVDF gives database administrators and IT security staff the ability to monitor attacks and discover what the scope of the attack was, including the IP address(es) involved, and wraps fundamental controls around the systems it monitors and protects.

**Key Differentiators**

- Transparent database activity monitoring over the network—no changes to database client and server configuration, and no additional load on network or the database servers being monitored
- Capability to block unauthorized database activity (such as SQL injection attacks) using a specialized grammar analysis that allows accurate enforcement of activity whitelists and blacklists
- Comprehensive database activity based on consolidated database logs, securely stored in a centralized, enterprise-scale repository
- Centralized data security auditing across the enterprise, achieved by consolidating OS, directory and other logs into the same centralized repository
- Fine-grained, correlated alerting based on analysis and policy enforcement of consolidated logs
- Out-of-the-box audit reporting across multiple sources (e.g., Oracle and non-Oracle databases, directory and OS) to satisfy common regulatory requirements such as PCI DSS, SOX and other compliance regimes
- Custom reports and powerful BI tools that allow organizations to go as deep as necessary for forensic analysis or e-discovery purposes
- Easy-to-deploy software appliance based on hardened operating system and database that does not require database administrator (DBA) skills, allowing the solution to be owned and managed by IT security staff
**Audit and Detection Setup**

During the evaluation, we examined the Audit Vault aspects of AVDF, which leverage logging as a detective measure, as well as the Database Firewall, which monitors SQL traffic and prevents common attack methods such as SQL injection, privilege escalation and other popular exploitations.

Our evaluation began with setting up an enforcement point—as shown in Figure 1—a Database Firewall that monitors and logs SQL statements.

![Figure 1. Creating a Database Enforcement Point](image)

In our case, we selected to monitor a database in the cloud with our selected database firewall.

Multiple targets for monitoring and multiple monitoring devices can be selected based on policies such as type of database, location of database and more. This way, the same console can monitor different databases or systems based on policy created in the interface.
Monitoring and Reporting Options

Most organizations have multiple database platforms, but this invariably leads to disparate reporting tools, complicating the risk management process. AVDF provides the capability to centralize this information from many database types—in addition to multiple operating systems, file systems, directories and other sources—via audit collection plug-ins, and then view results through a web console. This is useful for monitoring privileged user activities, audit and compliance reports, and collecting evidence of attacks that organizations can use to identify and suppress future threats.

Optional deployment options include inline blocking, out-of-band monitoring and remote monitoring of inbound SQL traffic to support a number of business use cases. Figure 2 shows an overview of AVDF components.

Figure 2. Typical AVDF Deployment
AVDF can perform inline blocking and monitoring, as well as remote monitoring, out-of-band monitoring and storage of the audit data itself. We began our evaluation by examining database activity monitoring, as described in the following section.

**Database Firewall Modes**

Companies can maintain visibility into potential attacks by monitoring database traffic. AVDF’s firewall typically operates in one of two modes. The first is DPE (Database Policy Enforcement) Mode, in which AVDF applies its firewall rules to SQL traffic and blocks potentially harmful traffic or substitutes a permitted SQL statement for a dangerous one.

For example, we created the policies shown in Figure 3 as part of a whitelist that meant these were the only SQL statements that, when submitted by a normal user, would be allowed through to the database.

![Figure 3. Firewall Policies Created for Monitoring/Reporting](image)

(Of course, the list of allowed SQL statements in a live deployment would likely be much larger than what is shown here.)

As an alternative, DAM (Database Activity Monitoring) Mode applies policies to monitor SQL traffic and raise alerts about potentially harmful activity. The same policies are used as shown above, but instead of actively blocking, we are now simply monitoring the traffic rather than enforcing policies.
AVDF analyzed the SQL statement grammar based on the security policy associated with the application and database we used. To further illustrate, consider our environment with the whitelist of permitted queries from Figure 3. We added the SQL statement `SELECT name FROM v$database` to the whitelist and set AVDF’s action type for these statements to “Pass” them to the database because we wished to explicitly allow this statement. Then we created a policy called “Block Unseen Statements” in the Firewall Policies section of the AVDF administrative interface, as shown in Figure 4.

Figure 4. Block Unseen Statements Policy

AVDF will now block any statements other than listed statements such as `SELECT name FROM v$database` or those from the policy rules dialog box shown in Figure 3.

The Value of Substitute Statements

A statement substitution can be used to give the user or application a response instead of “dead air”—no response at all. The substitute statement that may be used is up to the AVDF administrator. The statement is a SQL query that can return error messages to the user or simply execute a different form of the original, forbidden SQL statement. This enables the database to return a response to the client side instead of leaving the client hanging.
In our examples, we used a substitute statement of `SELECT 100 FROM DUAL` for such blocks; a return value of “100” indicated that the rule had been triggered and the substitute statement was run instead of the original query.

Figure 5 shows the SQL console with the blocking policy in place.

![Oracle Database DB06 SQL Console](image)

**Figure 5. Results of Blocking Unseen Statements**

Queries such as `SELECT username FROM dba_users` are now blocked, as indicated by the return value “100”; authorized statements such as `SELECT name FROM v$database` are allowed and return real data.

**Detect, Block and/or Log**

AVDF sits in front of the database engine and monitors SQL statements, access attempts and other activity, while needing no additional setup within the databases being protected.

When monitoring for attacks, AVDF supports both positive (whitelist-driven) and negative (blacklist-driven) models of IT security to detect and block malicious activity from reaching the databases it protects.

When setting up whitelists and blacklists, characteristics such as network, application, username and other attributes can be used to evaluate a potential threat. In this case, we created a user set that granted the user `sys` an exception to the firewall policy. Note that statements from `sys` would be logged. This user could issue any statements, while the other users would have traffic blocked according to the firewall rules. Figure 6 shows the user set that was created, which allowed `sys` to issue any SQL queries; `sys` is the only user granted an exception to the firewall policy whitelist.
While this policy is active, the user sys can query from the **dba_users** database; any other users are blocked. As seen in Figure 7, the query was processed for the user sys and a username returned, but the same query was blocked for the user system and “100” returned instead, indicating a block.

**Figure 6. Making User sys an Exception to Firewall Policy**

**Figure 7. Blocking a Query by User system**
After this activity was blocked, we could also view the activity via the logging capabilities. Note the first entry in Figure 8 is the query issued by system to the dba_users database.

Additionally, as the log records show, we were also able to further drill down to the event level and investigate details without having to change anything in the application or database. If you click on the individual event record indicated in Figure 8, additional detail about the event will appear, as shown in Figure 9.

SQL injection is one of the common attacks used to compromise data from databases today. Typically, SQL injection attacks can be hard to block because the actual SQL being sent to the database is not typically monitored and the vulnerability is a result of poorly written application code. AVDF can both monitor and block SQL injection attacks.
To test this, we ran multiple queries against the database. The first query was authorized and, therefore, AVDF allowed the statement to be executed and it returned results, while the second query, which included SQL injection syntax, was blocked because it was an unauthorized query, as shown in Figure 10.

```
SQL> select name from v$database;
NAME
-------
DB06

SQL> select username from dba_users where user = 'TANYA' or '1' = '1';
100
100

SQL>
```

*Figure 10. Blocked SQL Query*

Again, the substitute statement of `SELECT 100 FROM DUAL` was used; when the query was blocked, “100” was returned.
AVDF provides centralized, consolidated and secure storage of audit and log data. This can make quick and easy work of compliance reports and other functions that support auditors or investigators during assessment and forensic follow-up.

**System-Agnostic Data Collection**

AVDF collects data from Oracle environments as well as Microsoft SQL Server, SAP Sybase Adaptive Server Enterprise (ASE) and IBM DB2 for Linux, Unix and Windows (LUW), as well as OSes and directory and file services, such as Oracle Big Data Appliance (BDA), Oracle Enterprise Linux, Microsoft Windows, Solaris, Active Directory and Oracle Automatic Storage Management Cluster File System (ACFS).

In our review, we targeted a single Oracle database, although additional systems can be monitored by AVDF if needed in an environment. Figure 11 shows the system targeted for audit reporting in our evaluation.

In addition to consolidating audit data from multiple platforms, Audit Collection Plug-ins can be used to collect data from application tables and XML files and then send that data to the Audit Vault Server. The plug-in expands the number of audit logs AVDF can capture, providing organizations with the capability to collect, consolidate, and manage event logs across many platforms, databases, file systems and directories.
Built-In Reporting

Many organizations use manual audit processes that are tedious and time consuming, which means that few, if any, reviews of the audit data will be very thorough. Using the data it collects, AVDF can report on database activity characteristics such as:

- **Database Firewall action and threat level.** What did the firewall allow (or block) and why?
- **Database username.** Who generated the database activity?
- **OS username.** Which user?
- **Statement type.** What is the nature of the statement?
- **Client application name.** Which application sent the request?
- **IP address.** Which IP address is associated with the request?
- **SQL statement.** What was the actual command the database engine executed?

The AVDF graphical user interface (GUI) gave us a centralized view of the monitored system, and the built-in reports provide data that can be used for compliance purposes. Figure 12 shows some of the reports that were available during our review.

![Figure 12. Compliance Reports in AVDF](image-url)
By clicking on any of the reports, we could observe the individual events. Figure 13 shows a report giving an overview of database activity from the previous 24 hours.

Each line item was another event in the life of the database. By clicking on the individual event, we could get information such as host name or system type, with the latter based on the installed database engine, OS or similar characteristics.

With a few clicks, we easily drilled down to audit information, such as the IP address and database the activity was conducted against. Figure 14 shows a detailed audit record.
AVDF gives you the ability to view any data audited in the database such as privileges and stored procedures, which enables auditors to track the use of, and changes to, procedures by applications as well as their users. The Stored Procedure Audit Reports provide a listing of how the stored procedures have been modified, when stored procedures are created or deleted, and other related activity, as shown in Figure 15.

The reporting interface is built on Oracle's Application Express technology, which enables quick and easy ad hoc reports in addition to AVDF’s built-in reports. The interface enabled us to easily customize reports with colorful charts and graphs, as well as supported more elaborate changes in format, such as adding, removing or relocating report columns.

Compliance requires solid reporting that addresses threats, is tuned to applicable risks and reinforces the organization's security posture; AVDF’s built-in reports provide a quick summary of stored data and information that may be needed for compliance purposes. The basic audit reports include information such as failed logins, database startup and shutdown, data access and data modification. A view of the reporting interface appears in Figure 16.
Scheduling and Attestation

As one would expect, AVDF enables report scheduling. It also goes further by enabling users to attest to the contents. Parameters can be set around how often the report is to run, to whom the report should go, and whether to include any notes, as shown in Figure 17.
The ability to attest to a report’s content is an essential part of audits; AVDF enables its users to add a message such as, "I have reviewed and attest to the data in this report," with the user’s name and the date.

The final step of the report scheduling allowed us to specify which users needed to attest to the report, as shown in Figure 18.

![Figure 18. Attestation of a Report](image)

**Compliance Reporting**

AVDF’s reporting tools also address the requirements of specific compliance regimes, including Data Protection Act (DPA), Gramm-Leach-Bliley Act (GLBA), HIPAA, PCI and SOX. Figure 19 shows the different categories of reports we found in the interface.

![Figure 19. Compliance Reports](image)

We were able to expand the relevant compliance sections and view the available reports, then select and view specific reports. This helps administrators and auditors find evidence of policy violations, such as the misuse or inappropriate access of confidential data by an insider.
Conditional Alerting

AVDF includes a GUI for creating customized alerts based on audited activity or data. This facilitates the proper separation of duties.

AVDF can send alerts via email and SMS, and comes with notification templates that make alerts simple to implement. For example, we used a template to create an email that would be sent when a new user is created, as shown in Figure 20.

![Figure 20. Alert Creation with Conditional Filtering](image)

This shows an example of how an alert can use filtering to control when it will activate; AVDF users can easily filter by the type or severity of event, type of target, asset owner and action taken, among other conditions.

Putting It All Together

The centralized web console makes it easy to set up and view the systems that AVDF monitors. The built-in reports provide details about database activity, and when needed, customized reports can be created. Alerts could also be easily set up to communicate relevant data in a more timely fashion.
Oracle Audit Vault and Database Firewall provides enterprisewide monitoring of audit and log data from database engines, underlying operating systems and even custom systems. The ability to monitor and protect enterprisewide systems in a single console can help break down the information silos that leave IT environments vulnerable.

With AVDF, consolidated reports and activity can be drawn from multiple brands and types of databases, OSes and filesystems. ADVF’s capabilities provide protection that enhances the database infrastructure, and its reporting facilities are used for common compliance regulations. The out-of-box compliance reports are helpful, and automatic alerting against prohibited behavior functioned as tested, while writing custom reports and alerts is also intuitive.

In our review, we were able to observe it detecting a SQL injection attempt and block it. AVDF can help identify and/or prevent many of the common threats pointed at database targets, while allowing the monitoring of multiple hosts and providing visibility into database traffic, as well as simplified compliance reporting for databases and other systems.

Overall, AVDF provides efficiencies for organizations looking for a way to quickly detect what actions are occurring in their databases—and what actions should not be allowed—and streamline their protection and audit capabilities.
Tanya Baccam is a SANS senior instructor, as well as a SANS courseware author for courses such as AUD445: Auditing the Security and Controls of Oracle Databases. Tanya has consulted with a variety of clients about their security architecture in areas such as perimeter security, network infrastructure design, system audits, web server security and database security. She provides a variety of security consulting services for clients, including system audits, vulnerability and risk assessments, database assessments, web application assessments and penetration testing. Tanya has previously worked as the director of assurance services for a security services consulting firm, was the manager of infrastructure security for a health care organization and was a manager at Deloitte & Touche in its security services practice. Tanya has played an integral role in developing multiple business applications and currently holds the CPA, CITP, GIAC GCFW, GIAC GCIH, CISSP, CISM, CISA and OCP DBA certifications.

SANS would like to thank its sponsor:

ORACLE®
# Upcoming SANS App Sec Training

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Location</th>
<th>Dates</th>
<th>Type</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>