

Research Note

Date: June 4, 2009

Title: Security Criteria For Selecting A Database: Oracle Database Server 11g vs. Microsoft SQL Server 2008

Written by: Bob West and David Mortman

Summary

In response to several requests for real life examples of how to apply the research note, this note compares and contrasts Oracle and Microsoft practices using the criteria advanced in Echelon One's 2008 Research Note "Security Criteria For Selecting A Database"

The Issue

At the beginning of 2008, Echelon One published "Security Criteria For Selecting A Database" which provided seven areas to investigate when deciding which database to purchase and deploy. This paper was intended to provide customers a set of meaningful criteria to use when assessing the inherent security of database software before its purchase. While the paper specifically mentioned database software, the criteria advanced in the paper can be used for any software purchase considerations. The initial focus on database software was –in large part- influenced by the debate created by certain vendors and security researchers whose claims about the number of published vulnerabilities in database software were misleading or self-serving. In other words, the intent of the paper was to provide a set of meaningful criteria to evaluate the security posture provided by database software, and debunk claims solely based on the number of published vulnerabilities. The initial paper stated: *"Opinions and vendor hype about the inherent security in databases abound. Many organizations only look at the number of security vulnerabilities that have been published when, in fact, much more needs to be considered, particularly in the realm of security assurance practices. This problem is exacerbated by the fact that some vendors actively promote vulnerability counts as an objective measure of quality."*

In response to several requests for real life examples of how to apply the research note, this note compares and contrasts Oracle and Microsoft practices using the criteria advanced in the initial Research Note, with a certain focus on Oracle's Database 11g and Microsoft's SQLServer 2008 products. As a refresher, those seven criteria were:

1. Product Security Features
2. Vendor Disclosure of Vulnerabilities
3. Secure Development Practices
4. Ongoing Software Assurance

5. Vulnerability Remediation
6. Independent Vulnerability Assessment and
7. Product Administration Usability

Discussion

The earlier research note then applied a series of questions to each criterion to highlight key security concerns.

Criterion 1: Product Security Features

In the Product Security Features area, customers should investigate the implications of the following four basic questions:

1. What are the security features and are they relevant? (N.B., the existence of a feature is no indication of its usefulness.)
2. How flexible are the security features, that is, can they be fine-tuned to a level of granularity that satisfies current and prospective information security and availability needs?
3. Is security turned on by default, and if not, is it easy to lock the product down?
4. Are specifically desired features, such as encryption, designed and implemented well – and in accordance with published standards?

The relevance of the security features is difficult for us to answer generically as the needs of each organization vary considerably from each other. However, there are operational differences between Oracle Database 11g and SQL Server 2008.

Oracle's ability to control privileged user access

A number of regulations require enforcing controls over access to sensitive resources such as information that is material to financial reporting (Sarbanes-Oxley) or personally identifiable information (PCI, HIPAA, GLBA). In addition, recent security incidents have validated the long-known but often-ignored issue of insider crime. This is an issue complicated by the fact that many applications were initially built to address high availability and performance requirements and not necessarily security. With trends such as out-sourcing continuing to increase, the need to put in place preventive controls over access to sensitive systems by privileged users such as system administrators, database administrators or individuals who have access to media including backups, become mandatory. The challenge of “application bypass”, which occurs when a perpetrator circumvents application controls by attempting to access sensitive data directly from the database, needs to be addressed while not breaking the existing application functionality.

Microsoft SQL Server 2008 provides little in the way of built-in access controls and relies heavily on roles combined with protection programmed via database triggers and database views. Microsoft's approach adds complexity to existing

systems and makes adding security to existing applications more complex.

Oracle provides options to minimize the “Application Bypass” problem. Oracle Database Vault enables organizations to transparently create what are known as “Realms” around application tables to prevent privileged users, such as DBAs, from using their privileges to access application data. Database Vault realms are transparent to the existing application. Database Vault also provides command rules and multi-factor authorization to enable organization to further control the conditions under which the database and application data are accessed, basically giving organizations the ability to create trusted paths to the database and application data, restricting access to specific subnets, IP addresses, program names, and many other conditions.

For government and public sector customers Oracle provides the ability to assign data classification labels (sensitive, highly sensitive) to data rows and restrict access based on the data classification label. This ability is provided by Oracle Label Security and is used today by both government and commercial organizations. Oracle also provides a feature known as virtual private database (VPD) that allows organizations to create custom policies that perform real time query modification – restricting the rows returned. When VPD policies are table column specific the option also exists to mask out column values on a real time basis.

Oracle’s ability to enforce advanced access control policies and transparently encrypt data allows organizations to meet strict security policies and meet regulatory requirements.

Today, Microsoft does not provide data masking capabilities. Customers need to rely on third party masking solutions, or attempt to implement SQL Server crypto functions for masking purposes (thus running the risk to see the data decrypted and reversed to its original value).

Oracle Data Masking provides a comprehensive solution for transferring copies of production data to development and test organizations. Oracle Data Masking can help organizations comply with privacy and confidentiality laws by masking sensitive or confidential data in development, test or staging environments. This helps maintain the integrity of the application while masking data. The data masking pack provides a central definition for common masking formats allowing organizations to apply common data privacy rules consistently to all production data and thus ensure compliance with regulations. Many organizations struggle with this when managing databases in-house, but this becomes a significant issue when business processes are outsourced either onshore or offshore.

Oracle provides Oracle Advanced Security to encrypt data-at-rest and on the wire. Oracle Advanced Security Transparent Data Encryption (TDE) enables organizations to transparently encrypt specific sensitive columns (ssn, credit

card) or encrypt entire applications using tablespace encryption. TDE also provides media protection, preventing privileged users at the operating system level from using their status to browse data files using OS level commands. In addition, database backups retain the data as encrypted so sensitive data remains protected on backup devices/tapes.

A template for analysis could be developed along the following lines, where one would weight the granularity of each feature by its relevance to the decision-maker's company. The general idea is for each feature considered, such as the ones below, to be rated based on their relevance to the organization (1 for Low, 2 for Medium, 3 for High), whether a specific vendor supports them (2 for yes, 1 for no), and weighted against a granularity score (decimal amount between 0 and 1) for how well the feature meets the organization's need. The multiplication of these three numbers provides a relative score, which provides for comparing the different production options.

| Feature | Relevance (L/M/H) | Oracle | Granularity Score | Microsoft | Granularity Score |
|-----------------------------|--------------------------|---------------|--------------------------|------------------|--------------------------|
| Table Access | | | | | |
| Login & Auditing | | | | | |
| Resource Control | | | | | |
| Session Control | | | | | |
| Encryption | | | | | |
| Change Tracking | | | | | |
| Data Masking | | | | | |

Criterion 2, Vulnerability Disclosure Practices

Vendor Disclosure of Vulnerabilities brought us to the core of the security assurance discussion by asking:

1. Is the vendor open to discussing the security issues the customer is facing?
2. Does the vendor realistically and consistently rate the severity of their vulnerabilities (e.g., by using Common Vulnerability Scoring System (CVSS))?
3. Are the vendor vulnerability remediation practices well documented and publicly available?

Both Oracle and Microsoft have published disclosure policies on their websites and, until recently, both had similar policies of not sharing vulnerability

information any earlier, even with "priority" customers. While Oracle has maintained this stance, Microsoft announced its Active Protections Program (MAPP) at Blackhat 2008 where it is now sharing vulnerability information with a select set of security companies. This practice by Microsoft does raise potential concerns about vulnerability details leaking prior to the availability of fixes.

Both vendors have been using vulnerability scoring on their advisories for a number of years. While Microsoft has been using a proprietary scheme since 2002, Oracle has used CVSS since 2006, and in 2007 adopted the second version of the standard. The use of different schemes by vendors makes it difficult for customers faced with heterogeneous environment to assess vulnerability severities. Ideally everyone would use the same system (such as CVSS), which would allow for a more consistent interpretation of the criticality of vulnerabilities for similar products and interfaces. Microsoft's vulnerability ratings system, while long standing and familiar to many users lacks sufficient granularity to help customers assess their risk and make a proper prioritization of patches possible. This lack of clarity is particularly challenging when one needs to deal with a heterogeneous environment. As a result we prefer the more specific, well-documented, and vendor-neutral Common Vulnerability Scoring System (CVSS).

While CVSS gives one the ability to map scores back to the National Vulnerability Database NVD ratings of Low (0.0-3.9), Medium (4.0-6.9) and High (7.0-10.0), these non-numerical scores need to be taken with a grain of salt. This is due to CVSS's application independent model, which requires a full operating system level exploit to yield a 10.0 Base Score. Under CVSS, even a severe database vulnerability such as full compromise of the confidentiality of a database (full read-access against the database) possible remotely without authentication wouldn't rate more than a 5.0 ("medium severity" according to NVD!) Oracle has highlighted the limitation of the Base Score rating during various CVSS working group meetings. In addition, while Oracle is strictly applying the CVSS formula in the calculation of the published CVSS Base Score in the CPU documentation, the company highlights those instances when a given vulnerability could result in compromising large number of records. These vulnerabilities are reported by Oracle with the value of "Partial +" rating in the security risk matrices in an attempt to help customers develop the proper context when reviewing the CVSS base score of the vulnerabilities. For more information about the limitations of CVSS and Oracle's use of the "Partial +" rating in its advisory, see <http://www.oracle.com/technology/deploy/security/cpu/cvssscoringsystem.htm>. As a best practice, customers should always ensure that they apply the appropriate context when evaluating CVSS scores.

The following table provides an overview of the respective practices of Oracle and Microsoft

| | Oracle¹ | Microsoft² |
|---|---|---|
| Dissemination Practice | All customers treated equally | Preferential announcement to security vendors. |
| Advisory Documentation | Prioritized matrix | Prioritized matrix |
| Vulnerability Severity Ratings | CVSS 2.0 Base Score | Proprietary system Low, moderate, important, and critical. |
| Usability | “Plain English” executive summary | “Plain English” executive summary |
| Patch Release Schedule | Quarterly with pre-release announcement the week before. | Monthly with pre-release announcement the week before. |
| Published Responsible Disclosure Policy | Yes | Yes |
| Working with External Vulnerability Reporters | Credits all researchers who are not Oracle employees or contractors and who follow responsible disclosure practices. In addition, Oracle started to recognize special contributions by researchers with the “Security In Depth” recognition in the Critical Patch Update documentation. | Credits all researchers who are not Microsoft employees or contractors and who follow responsible disclosure practices. |
| Publish Sample Code | No | No |
| Prioritization of the fixes | Vulnerabilities fixed in severity order Fixes included in the main code line first | Vulnerabilities fixed in severity order Fixes included in the main code line first |
| Are patches cumulative? | Yes | No |
| Inclusion of fixes in | Yes | Yes |

¹ For further information, see:

<http://www.oracle.com/technology/deploy/security/securityfixlifecycle.html>
<http://www.oracle.com/technology/deploy/security/alerts.htm>

² For further information, see:

http://www.microsoft.com/security/msrc/incident_response.msp#EPC
<http://www.microsoft.com/technet/security/bulletin/policy.msp>
<http://www.microsoft.com/technet/security/bulletinsandadvisories/default.msp>

| | | |
|--|---|--|
| service packs and sub-releases | | |
| <p>Does vendor provide utilities for patch management:</p> <ul style="list-style-type: none"> • Inventory collection • Unattended patches • CLI • Rollback • Grid support | <p style="text-align: center;">Yes</p> <p>Opatch : a CLI tool that supports rollback and RAC. My Oracle Support provides free utility to gather environment information and displays recommendation to customers, including patch application and configuration (Available to all customers under maintenance). Option: Oracle Enterprise Manager Provisioning Pack Grid/Live Patching : Yes</p> | <p style="text-align: center;">Yes</p> <p>Microsoft Baseline Security Analyzer (MBSA), Microsoft Update (MU), Windows Server Update Services (WSUS), Systems Management Server (SMS), System Center Configuration Manager (SCCM) 2007 Grid/Live Patching : No</p> |

Criterion 3 Secure Development Practices

Arguably, vendor’s development practices have the greatest impact on the security posture of customers. This is because effective secure development practices should result in the limitation of the number of critical vulnerabilities in the software, as well as the reduction of their severity. In addition, good development practices should result in an overall improvement of the code as “lessons learned” result in changes in developers’ behavior. Finally, one cannot separate secure development practices with ongoing assurance, as the maintenance of secure code should not end with the GA release of the software. Echelon One believes that the following attributes characterize effective Secure Development Practices by commercial vendors:

1. Create security requirements as early as the design phase. The application owners need to effectively communicate their security requirements before the development work on the product is started, so as to ensure that the application has the appropriate security qualities when the application is completed. If implemented properly, this will save the organization money by minimizing the chances of serious design errors, which may result in critical flaws, and expensive reengineering of the application. Adherence to coding standards should also result in a reduction in the number of security patches that need to be issued and thus decreasing overall security maintenance costs. All things being equal, if the numbers of potential security issues are minimized, then the use of the software should not jeopardize the risk posture of the organization. Secure development practices can also be a differentiating factor in between software vendors. Consumers should be looking at vendor security assurance practices as a basic requirement for their selection criteria. By asking the right questions they can determine which vendor has a better handle on security. Such questions may include:
 - a. Has the vendor documented secure coding standards? When were

they first published? Who is in charge of maintaining these standards?

- b. How is compliance with the security development standards enforced in the organization? Is compliance with the secure coding standards a release criterion?
 - c. Are security requirements expressed as early as the design phase of software?
 - d. What is the process for updating the secure coding standards? Are those case-based (lessons learned)?
2. Integrate security in ALL areas of the development lifecycle. It is essential that security be integrated as part of the overall software development lifecycle. In the manufacturing world, people don't generally think about adding quality or safety features (??) after an automobile comes off of the assembly line. Quality is part of the fundamental manufacturing process and can't be added-on afterwards. In the computing world, Security Assurance is more important because most production environments will be exposed to malicious threats: IT professionals should expect that many people and organizations will try to break into their systems or "probe the fences" of databases after they have been placed into production. Specifically, databases often store the crown jewels of an organization: its data and intellectual property. If the right level of security isn't embedded into the fundamental database model, organizations will be unable to enforce proper control over their data, thus giving an opportunity for unauthorized personnel and criminals to access information they shouldn't have access to. The integration of security in all areas of the development lifecycle by the vendors should result in delivering products that provide effective security controls that meet "real world" requirements of customers. Customers should ask the following questions of their vendors"
- a. Does the vendor have formal secure development guidelines?
 - b. How does security impact the development of the vendor's products?
 - c. Does the vendor use threat modeling or other means to assess the effectiveness of its products security in a real world scenario?
 - d. Does the vendor maintain internal expertise for "ethical hacking" and other security penetration tests against its products?
3. Ensure that developers maintain knowledge of security fundamentals. There are very few university programs that teach developers how to write applications securely. Because of this, most application developers enter the workforce without an understanding of what constitutes good security hygiene. This is a significant issue because the majority of system compromises occur by leveraging application vulnerabilities. It is interesting to see that both Microsoft and Oracle executives have voiced concern about this issue, trying to prompt for changes in the IT grad curriculum to include secure coding training. However, until this problem

- is solved, it is the responsibility of the employers to teach developers these security fundamentals. Most software providers either develop training programs or send their developers to a series of external training programs to ensure that their developers understand how to write applications that have the appropriate level of security integrated into their applications. The training programs needs to be structured so that security is taught to all relevant staff (development, support, product management, etc). In addition such training must re-occur periodically to ensure that the knowledge is retained. Customers should ask the following questions:
- a. Does the vendor have formal secure coding training for its developers?
 - b. Is secure coding training mandatory for developers? What about product managers? Architects? Etc.
 - c. Are the organization's secure coding standards included in the training material provided to its employees?
4. Promote the ongoing use of automated tools to detect vulnerabilities in code. Part of the development process should include the use of automated tools that detect potential security vulnerabilities. This will help the organization detect and remediate security issues before the application is released (and put into production by customers). Automated tools can also play a very significant educational role by providing timely feedback to developers about the security quality of the code they develop. In addition, the use of automated tools provide for ongoing testing of code thus ensuring that fixes are released quickly throughout the development phase (as opposed to periodic human reviews). While the use of automated tools is very important, human review is also necessary as expert review often allows for the detection of more complex or sophisticated issues that may be ignored by automated scanning tools, particularly in critical sections of code. The right combination of both human and automated review helps minimize issues throughout the time an application is in production. Customers should ask:
- a. Does the vendor use automated testing tools? Are these tools used throughout development or only at the time of the pre-release (QA)?
 - b. Does the vendor use human review as well as automated tools? What is the process for fixing issues discovered throughout development?
 - c. How are coding changes managed so as to avoid the introduction of new security vulnerabilities?
5. Seek external validation of the security of the product through third-party penetration testing, code reviews, or certification. Organizations can do a good job of minimizing security issues in an application but an objective third party has a different perspective and can detect issues the organization can't see, mostly because third party organizations will not

feel bound by the “use case” provided by developers, and will not hesitate to use obscure methods in an attempt to crack the system open. This can also be because the organization has become too close to the application or that they may be using different tools to review the application. Utilizing a third party to validate the security on a periodic basis will help ensure that security issues are not overlooked. In the case of certification, using a model the Common Criteria can provide a common baseline for comparison. Customers should ask:

- a. Has the vendor submitted its products to any independent security validations such as Common Criteria? Are these certifications obsolete?
- b. Does the vendor maintain an ethical hacking capability? Does the vendor use an external vendor for penetration testing, white hat hacking, and security design review?
- c. Does the vendor have a mechanism to integrate the lessons learned from white hat hacking exercises into its secure development standards?

Both Microsoft and Oracle have integrated security into their development processes and have also published extensive information about how their processes work on their websites. In particular, both also require extensive documentation of security requirements very early in the process. There are some differences however.

Organizational structure is somewhat a point of divergence between Oracle and Microsoft. At Oracle, a central group (Global Product Security) is responsible for enforcing the application of the standards, training, etc. Development retains the ownership of its code (and its fixes). Each development group has assigned Security Points Of Contact (SPOCs). The SPOCs are their group’s main interface with Global Product Security. They are the gatekeepers in charge of promoting security within their respective organization and ensuring compliance with Oracle Software Security Assurance.

Microsoft has a central group, Security Engineering and Community (SEC), which includes Security Development Lifecycle (SDL) and Microsoft Security Response Center (MSRC). Product groups are responsible for security, and many large ones, including MS-SQL have their own security teams who partner with SEC. Additionally, SEC does not check in changes to other people's code.

Microsoft and Oracle, driving what is becoming an industry standard, have regular mandatory security training for all employees who are part of the development process.

Section 3 concluded by highlighting the value of automated tools as well as third party penetration tests and certifications. Unsurprisingly, both vendors use third party consultants and code review tools extensively.

While the Common Criteria is not perfect, it is of particular importance for many customers due to its mandated use by the federal government and the requirement it creates for commercial vendors to have strong and documented security practices across development. As we mentioned in Section 2, Oracle has been a long time proponent of the Common Criteria Certification (see <http://www.oracle.com/technology/deploy/security/seceval/security-evaluations.html>). Oracle has certified all the major releases of its database as being at least CC Level 4 compliant starting with Oracle 7.2. Microsoft has not embraced it to nearly the same degree which is evidenced by only having certified one release of SQL Server. For more information see http://www.commoncriteriaportal.org/products_DB.html#DB

| | Oracle | Microsoft |
|---|---|------------------------------|
| Security Spec'd As Part of the Product Design Process | Yes | Yes |
| Specified Coding Guidelines | Yes | Yes |
| Centralized Security Group | Yes | Yes |
| Mandatory Security Training for Developers | Yes | Yes |
| Use Automated Code Review Tools | Yes | Yes |
| Third Party Consultants and Code Review | Yes | Yes |
| Common Criteria | Oracle Database 10g (10.2.0.3) EE, SE and SE1 Editions at EAL4+ Oracle Label Security 10g (10.2.0.3) at EAL4+ Oracle9i Database (9.2.0.1.0) at EAL4+ Oracle9i Label Security (9.2.0.1.0) at EAL4+ Oracle8i Label Security (8.1.7) at EAL4 Oracle8i Database (8.1.7) at EAL4 Oracle8 Database Server (8.0.5) at EAL4 Oracle7 Server (7.2.2.4.1.3) at EAL4 In addition, the following have been submitted for evaluations: Oracle Database Vault 11g (11.1.0.7) at EAL4+ Oracle Database 11g | SQL Server 2005 SP2 at EAL4+ |

| | | |
|--|---|--|
| | Enterprise Edition (11.1.0.7) at EAL4+ Oracle Database 11g Standard Edition (SE) & Standard Edition 1 (SE1) (11.1.0.7) at EAL4+ Oracle Label Security 11g (11.1.0.7) at EAL4+ | |
|--|---|--|

Criterion 4 Ongoing Software Assurance

One of the keys to long-term success is for the vendor to have a sustainable software assurance program. As with any business process, it needs to be a sustained initiative and, in this case, integrated into the software development lifecycle. Because most customers have applications that remain in production over extended periods of time, this will provide them with a level of assurance that they will be able to depend on secure database environments in the long term.

Both Oracle and Microsoft provide long-term support for their products as stated in the policies published on their websites on a product-by-product basis. Both companies have also shown themselves to be flexible with (End Of Life) EOL dates when pushed by customers. Beyond that, some of the questions that customers should be asking with respect to long-term product support include:

1. Does the vendor have an effective process for keeping customers informed about vulnerability alerts?

Customers of commercial software rely on their vendors to inform them of any significant issues in software they have purchased. This is because customers typically have very little insight in the source code of the products they are running. If the customers are not aware of an otherwise known and exploitable vulnerability, they are exposed to risk and potential financial losses. A consistent process to inform customers about vulnerabilities, the availability of fixes or workarounds (if applicable), and how to assess the risks in their environment needs to be well-defined, and properly managed and executed by the vendor.

Customers should ask:

- a. Does the vendor have a clear security remediation policy?
- b. Does the vendor provide for a predictable patching schedule? Does the vendor provide emergency patches if required?
- c. Has the vendor an effective infrastructure to communicate security issues with its customers?
- d. Does the vendor engage in disclosing security vulnerabilities when patches are released?
- e. Does the vendor treat all its customers and partners equally as it relates to disclosing security issues in its products?

2. Does the vendor have a long-term plan for supporting the product? Every so often a customer identifies flaws in older versions of its products. If the database is supported, the proper fix should be available to customers from the vendor. In order to optimize the customer experience, there needs to be a relevant, long-term plan to support customers and inform them when flaws are identified in their databases. Excessively short support periods typically require customers to engage in constant upgrades, and this can result in excessively high cost of ownership. Customers should ask:

- a. What are the vendor's published policies for support? How does this policy affect the availability of security fixes?
- b. Does the vendor provide incentives to upgrade to more recent (and hopefully more secure) versions of its products?

3. How comprehensive and thorough is software testing prior to product release?

To get its maximum value, security testing must be integrated in all phases of the application development process. Many companies take the approach that security should be tested before the database is released to its customers, as a function on "normal" QA practices. However, if security is not embedded deeply in the product life-cycle, the right security controls may not exist in the database product. In order to achieve an optimal level of protection, testing should occur throughout the product life cycle. This includes but is not limited to rechecking previously developed code as new vulnerabilities and attack techniques are discovered. This is important because otherwise you run the risk of developing "code rot" where formerly solid code appears to become unstable, where in reality, the existing vulnerabilities have now become apparent.

Similarly how a company reacts and deals with vulnerabilities is as important as how they develop their code to begin with. This brings us into Criterion 5, Vulnerability Remediation.

Criterion 5 Vulnerability Remediation

Vendor vulnerability remediation practices are the most visible aspect of a vendor's security assurance activities to customers, and an area of prime concern to them. However, in our opinion, vulnerability remediation practices are as important as secure development practices. Vendors need to prevent vulnerabilities as much as possible. In addition, secure development practices must extend to the creation of security patches themselves to ensure that these patches are defect free or do not introduce new vulnerabilities or regression issues. A vendor vulnerability remediation practices must result in providing appropriate fixes in a timely fashion and in a cost-effective manner. In previous sections, we reviewed various security assurance activities that are directly relevant to assessing the effectiveness of a vendor's vulnerability remediation strategy. In an environment that has become particularly hostile, with the most

significant threats ranging from organized crime groups such as the Russian Business Network to governments that are intent on stealing intellectual property or military secrets, vendors' vulnerability remediation practices play an important role in maintaining a collective security posture.

Customers should ask:

- a. How mature are the vendor remediation policies?
- b. Does the vendor provide a predictable patching schedule? Does the vendor provide for out of cycle patch releases in case of critical threat?
- c. Is the frequency of the patch releases appropriate for the customer's production environment?
- d. Does the vendor provide appropriate documentation with its patches

There are notable differences between Oracle and Microsoft as it relates to their respective vulnerability remediation practices. While both vendors provide customers with patch information via email and the web, Oracle releases its patches on a quarterly basis, while Microsoft releases them monthly. Both vendors have processes for releasing emergency patches out of cycle when necessary.

One of the biggest points of contention between vendors, researchers, and users is whether vulnerabilities are released in a timely enough fashion. This is perhaps even more subjective an issue than the relevance of security features as mentioned in Section 1. Oracle's patching frequency (quarterly) has been defined with the input of customers. While in theory a more frequent patching frequency should result in a smaller "window of opportunity" for hackers, in reality organizations have little room for the downtime associated with frequent patching efforts. In addition, patching complex production environments typically requires extensive testing. In a recent joint IOUG-Oracle survey customers reported that "patching take hours and testing before applying the patches takes months".

To a large extent, the focus of customers on testing patches is due to concerns by customers associated with the reliability of patches (i.e., how likely are the patches to be reissued and are they likely to create unintended downtime in production systems), determination of patch priority (i.e., how easy is it to determine if a patch needs to be installed), and ease of patch administration (i.e., how hard is it to install a patch if needed). This has a direct impact on total operating costs. If the process is too cumbersome, it will not only increase the distribution costs but it may not be executed consistently by the customer.

Customers should ask:

- a. How are patches prioritized?
- b. How easy is it to deploy patches?
- c. How are patches tested by the vendor before their releases?
- d. Are the frequency and timing of patch releases from the vendor compatible with production requirements?

Patching can have a substantial impact on uptime, performance, etc. and frequency will probably be different on servers than on desktops. For example, quarterly or semi-annually may be a reasonable interval for most servers. Business factors heavily drive how often customers are comfortable patching. As a comparison, Cisco is moving to releasing patches every 180 days. Critical patches should be evaluated on a case by case basis and reviewed to determine if patching outside the usual change control windows is appropriate.

Criterion 6 Independent Vulnerability Assessment

An organization can perform its own vulnerability assessment, but, for a variety of reasons, this may not provide an objective view of the security posture provided by the product in a real-world production environment. Biased assessments could lead to unwanted vulnerabilities in the products the vendor delivers to its customers. A third party has no political issues to deal with and can deliver the unvarnished truth with few repercussions. Independent testers will not feel bound by deployment scenarios provided to them by the developers of the product. They will feel free to attempt to exploit any packages or interfaces in ways that were not imagined by development in order to “break into” the software. In order to have an objective understanding of a database’s security posture, it is necessary to periodically engage a disinterested third party.

From a database customer standpoint, the use of independent security assessments is also valuable for the same reasons such assessments are valuable for vendors. Through a proper assessment, customers can determine whether their deployments provide an acceptable risk posture. In addition, customers should ask whether the vendor offers tools to verify configurations and missing security patches. Through proper tools, automated configuration checks against a corporate security baseline can be done more often than an exhaustive security review by an independent security team. Third party tools can also be another method to independently evaluate the security posture provided by the product. Given the complexity of large applications such as database software, it is very difficult for people to perform a comprehensive code review without overlooking any issues. As a result, code evaluation and fuzzing tools are great for identifying common issues which then free up employees to focus on discovery more complex issues.

In recent years, the Common Criteria certification has gained popularity. This is largely due to US government mandate that all computing products carry a certification. Even outside the government space, use of Common Criteria is slowly but surely increasing as is currently the only standard available that is internationally accepted. This is an area where Oracle has a substantial advantage over Microsoft in terms of the number of database products that are common criteria certified (see section 4 for product listing).

Criterion 7 Product Administration Usability

Closing out this checklist is a discussion about Product Administration Usability. The questions we asked were:

1. Are product guidelines, templates, case studies, or reference frameworks available that ease the task of securely configuring the product?
2. Does product documentation include warnings about potential technical difficulties?
3. Does the vendor support user group discussions about changes in features and different use case scenarios?
4. Does the vendor charge customers for security patches, product documentation, and best practices guidelines?

Oracle and Microsoft both provide extensive documentation, case studies, and discussion groups to provide support for security configuration and management of the databases. Oracle provides patches and documentation access to all customers who are under active maintenance. Microsoft provides all documentation and patches for free.

Conclusion and next steps

Both Microsoft and Oracle have been working to improve the security of their database products. Both companies have very strong security assurance practices with very little differences and are leading the software industry as a whole with their efforts. Similarly, both Oracle and Microsoft are heavily engaged in various efforts to drive the importance of security assurance with the software industry as a whole. As discussed at the beginning of this paper, despite marketing claims to the contrary, there is more to measuring the security of a product than comparing the number of patches released over a given period of time. It is our hope that this research note will help readers to be much more informed and empowered to make an educated and intelligent decision about which product best meets their needs from a security perspective.
