

BUILDING UNBREAKABLE ORACLE9i AND ORACLE9iAS

Duncan Harris and Rajiv Sinha, Oracle Corporation

UNBREAKABLE?

For more than a year, Oracle has run a marketing campaign: Unbreakable. And it's still running today. The security half of Unbreakable is "Can't Break In", concerned with Oracle's efforts to build, deliver and support the most secure software in the business, concerned with providing assurance, or reassurance - that warm fuzzy feeling - to our customers that Oracle knows what it's doing when it comes to security. But Unbreakable isn't just about feelings; there's strong technical proof behind the headlines.

This paper shows how Oracle's product development practices have been revolutionised with security assurance measures and how the most rigorous pre-release security checklists in Oracle's history, those that Oracle's latest server products, Oracle9i Database Release 2 and Oracle9iAS Release 2, have just been through, came about. These technical assurance measures and checklists weren't just created overnight; they're based on 14 years' experience of formal government security evaluations, of lessons learned through Oracle's internal ethical hacking group, and of classic secure coding pitfalls.

Unbreakable is not just a marketing campaign or a fad; it's an ongoing process and a commitment to our customers.

ELEMENTS OF UNBREAKABLE

There are three key components of Unbreakable: security evaluations, security assessments and a secure product lifecycle.

Oracle proves the strength of its products' security through an independent measure of assurance; that is, third party attestation through a formal security evaluation of the validity of its product security claims. These independent measures of assurance are a key element in Unbreakable because *how* you build products, from a security perspective, is ultimately more important than *what* you build. What you built can only be truly validated as secure if you know how it was built.

Backing up security evaluations is Oracle's internal security assessment group of "ethical hackers". This group mimics real world hackers by trying to disprove Unbreakable, by breaking into Oracle's own products before the real world hackers do. In effect, they are auditing that products have been designed and built according to Oracle's secure coding standards.

The third key element in Unbreakable is the commitment to a secure product lifecycle, and to continuously improve upon it. Assurance is a critical part of creating and maintaining that lifecycle. Indeed, to establish a proof in the correctness of security functions, you need a secure development process that is demonstrably repeatable and need to show you do not break old security functions when adding new ones. Security cannot be an afterthought, it cannot be tested into existence; it must be designed in from the start.

SECURITY EVALUATIONS

Information assurance is the formal proof that a security mechanism is correct and well formed. The main vehicle for substantiating a software vendor's security claims is an independent, third party, formal, security evaluation conducted against internationally recognised, security evaluation criteria. Therefore an evaluation, by these definitions, provides information assurance.

Oracle is the undisputed market leader in formal security evaluations, with 15 independent security evaluations against every major worldwide criterion over the past 14 years, including the Common Criteria (ISO 15408), the de facto worldwide evaluation standard. No other database vendors, in fact no other software or hardware vendor of any description, has conducted so many security evaluations. Quite pointedly, our Unbreakable campaign highlights the fact that our two main database competitors, Microsoft SQL Server and IBM DB2, have only conducted 1 and 0

security evaluations of their database servers respectively. This is no mere boast; it is a reflection of how seriously each company takes its claims to have secure products. Any vendor can claim they are Unbreakable, that they have a secure product, but only Oracle puts its money where its mouth is and security evaluates every major release of its core database server product.

In the past each nation had its own standards for national security and vendors who wanted to sell their products to governments had to conduct a formal security evaluation against those standards. For example, the U.S. used to have the Trusted Computer Security Evaluation Criteria (TCSEC, or “Orange Book”) as its standard and the European Union has the Information Technology Security Evaluation Criteria (ITSEC). Even the Russians had their own Federal Security Evaluation Criteria, against which Oracle performed two evaluations. However, the recent trend has been towards globalisation of such criteria, or rather towards a mutually accepted and understood language that still allows each nation to specify its own security requirements but also allows for a product evaluated in one country against a specific, mutually agreed, assurance level to be recognised in another country. The international Common Criteria does just that, and 15 countries have now signed the official Common Criteria Recognition Arrangement (CCRA).

Oracle’s recent evaluations against the Common Criteria are to the EAL4 assurance level, the highest level that the CCRA accepts, and the highest level that commercial software can generally be considered possible to achieve. EAL4 is not easy though, being the toughest security standard available for commercial software and taking a great deal of time and money to complete. After years of practice, Oracle can now complete an evaluation in approximately 5 months: our first ever security evaluation, completed in 1994, took 4 years!

Independent measures of information assurance are required to sell into the U.S. Federal government. A Federal policy directive, National Security Telecommunications Information Systems Security Policy (NSTISSP) Number 11, requires information systems involved in national security to have independent measures of assurance, such as a Common Criteria evaluation or FIPS-140 evaluation. The U.S. Department of Defence is also backing NSTISSP #11 with its own policy directive #8500.

Although receiving a certificate at the successful conclusion of such an evaluation is a direct benefit to specific government and military customers who have a requirement to buy only products that had been evaluated, the reasons Oracle has conducted evaluations for so long have changed over the years. Originally Oracle started doing evaluations for the following reasons, in this order:

1. For the benefits of the few customers who were required only to buy evaluated products
2. For the competitive marketing benefit, and
3. Because the evaluation process might find some bugs in the products.

Today, however, the order is completely reversed. We continue to do evaluations:

1. Because the process found and finds not just product security bugs, but also drives development process improvements, and
2. To continue our competitive advantage in security, with Unbreakable the strongest proof of its success.

However, on top of all these reasons are the key assurance benefits from formal independent security evaluations, which accrue to our customers:

- *A more secure product.* Security evaluators find security vulnerabilities during the evaluation, which must be remedied as a condition of completing the evaluation.
- *A demonstrably secure development process.* A formal security evaluation includes a review of the development processes, including the product security architecture, functional specifications, design specifications, test specifications, and the actual testing processes. Security must be integrated with these processes, and repeatedly so, in order to obtain and maintain security evaluations.
- *A culture of security.* It is ultimately a “culture of security” that is the most valuable result of Oracle’s commitment to security evaluations. Security is not an add-on; it is ingrained in our products from inception and has been so

for the ten years we have been doing formal security evaluations.

Formal evaluations are part of our secure product lifecycle; each of Oracle's 15 security evaluations represents an additional \$1,000,000 investment in security by the company just in assuring that the security mechanisms are correct. This cost is exclusive of the additional features and functions we build as we enhance our product over time.

However, on their own security evaluations are not enough. While formal security evaluations have clear benefits, they are not well suited to all types of products, for reasons of:

- *Technology* — many of the new, web-facing products incorporate technologies that are not yet well understood by the evaluation bodies or laboratories. Security researchers or “hackers” often have more cutting-edge skills in this area.
- *Time-to-market* — evaluations typically take about 18 months per server product, which is about two or three product release cycles for web-facing products. Literally by the time the evaluation is done, the product would be obsolete.
- *Expense* — evaluations cost about \$1,000,000 apiece. This would not be an issue were it not for time-to-market considerations. There is no point to spending \$1,000,000 to evaluate a product that is obsolete by the time the evaluation is completed.

Evaluations are typically performed on a production release of a product and thus have little chance to influence security design decisions made during development. Oracle has been working evaluation after evaluation to pull back the date on which the product successfully completes evaluation and is certified. But the nature of security evaluations means it is virtually impossible to get a certificate awarded on or before the product is actually released for sale to customers. In addition, over the past 2 to 3 years, Oracle has seen a shift in the focus of real world hackers away from operating systems and towards OS applications, such as a middle tier application or web servers. Most hacker activity is against these web-facing components and very little against core backend database servers.

SECURITY ASSESSMENTS

To augment our commitment to formal evaluations, Oracle has expanded its security assurance group's activities to include security assessments on products for which formal evaluations are not currently feasible, and various “ethical hacking” activities. Security assessments can include everything from security architecture review to “black box” testing (install the product and try to break in) to “white box” testing in which we scan the source code of the product. While security assessments do not result in formal assurance levels (e.g. EAL4), as there is no formal methodology involved and we do not always use third parties for these, we believe that they increase the security assurance in our products, as well as building our expertise in-house of “thinking like hackers.” Ultimately, it is far better to break into your own products than to wait for someone else to do it.

This role of mimicking real world hackers in a controlled, ethical environment inside Oracle has reaped dividends. The group has had a 100% break-in “success” rate, meaning that every Oracle product presented for security assessment has been “broken into” in a laboratory environment, exactly as it could be by real world hackers. Having broken in, the group raises product security bugs that are treated as seriously as if a real world hacker had reported the same vulnerabilities. We're looking forward to the day when the security of an Oracle product is so well designed and implemented that we cannot break in and defeat those security mechanisms. As security awareness of Oracle's product developers improves, you might think that the security assessment team should have less and less “success”. But they won't do themselves out of a job because emerging new technologies always result in new techniques for hackers to defeat security; this role of auditing product development's security implementations can never go away. Knowledge we gain from security assessments is incorporated into our coding standards and hacking techniques as part of continuous security process improvement.

Oracle expects that, as web-facing products mature, as their security technologies are better understood and as security evaluation criteria for them are better defined, they will be subjected to formal evaluations. Security assessments now conducted on the main security components of our products will likely be supplanted by formal evaluations in time.

As with security evaluations, security assessments are performed after a product is released. So what Oracle doing to address the complete security assurance picture with pre-product release security measures? Improvement of

Oracle's general development process has been brought about through the rigour of security evaluations and the success of the security assurance group's ethical hackers.

SECURE PRODUCT LIFECYCLE

Beyond the assurance measures of security evaluations and security assessments, Unbreakable includes an Oracle-wide commitment to a secure product lifecycle. Security cannot be "bolted on" after a product has been completed; it must be embedded within every stage of the product development and delivery process. Security must be part of the corporate DNA, wired into the fabric of the organisation at every stage in product development and delivery.

In addition to the formality of security evaluations and the hacking success of security assessments, Oracle's secure development process now includes all of the following elements:

- Secure coding standards
- Security templates for functional, design, and test specifications
- Security regression tests
- Centralised security functions
- Vulnerability handling
- Security design reviews
- Product security release checklists

SECURE CODING STANDARDS

The only way to build and deliver secure products is for each developer to assume personal responsibility for delivering secure code through knowledge of and commitment to secure coding standards. Secure coding standards thus form a baseline of security with which every developer in Oracle must comply.

Secure coding standards follow other Oracle-standard development practices. For example, Oracle has long had 'C' coding standards that developers are trained to follow and a development tool, OLINT, Oracle's enhanced version of the standard Unix tool lint, to check code for compliance to the coding standards. Just as you cannot feasibly have a single group that reviews every line of code for 'C' coding compliance but must engender responsibility among all developers for compliance, you can never hire enough "security police" to make your code secure. Each developer must have personal knowledge of and commitment to basic secure coding practice.

Oracle secure coding standards also direct development groups to use centralised security functions where appropriate. For example, developers do not need to be cryptography experts (and most are not) to use encryption. In fact, cryptography is typically easy to get wrong and hard to get right. Rather, developers need to know how to use standard (approved) encryption libraries correctly.

Secure coding standards "raise the bar" on security in several ways:

- Developers who are educated about basic secure coding practice will avoid common security pitfalls more often, and earlier in the release cycle
- Security becomes ingrained in developers' skills over time with repetition
- Developers are more likely to consult security experts as needed since the coding standards also include identifying pointers to the core security development group

Oracle conducts periodic training covering its secure coding standards that are subject to continuous improvement. Oracle incorporates "lessons learned" from ethical hacking efforts, reported security vulnerabilities, and from information gleaned by participation in industry information sharing forums, e.g. the Information Technology-Information Sharing Analysis Center (IT-ISAC), into our secure coding standards.

Coding standards also form a security baseline with which all developers in Oracle are expected to comply; for example, the ethical hacking team will not conduct product security assessments until requesting teams have "self-assessed" by reviewing the coding standards and other security checklists.

Another reason we enforce coding standards is the economics of “pay now or pay later.” Oracle products run on multiple operating systems, and support multiple product releases at any given time. Some of our competitors only run on one or two operating systems; it is cheaper for them (but not for their customers) to use their customers as their quality control organization. If they have a vulnerability, these vendors merely issue a patch for two releases on two operating systems. In contrast, Oracle has issued as many as 78 patches for one security vulnerability, to cover all affected releases and operating systems.

Oracle’s cost avoidance, by building security correctly the first time, is also our customers’ cost avoidance. It is extremely expensive for customers to download, test, and apply security patches to their systems. One of our competitors issues a security alert every 5 days (in contrast, Oracle’s average is one security alert every 19 days) so their customers are constantly patching their systems for security flaws. Ultimately, our competitors’ customers cannot keep up and are then vulnerable to the latest exploits of unpatched security holes.

SECURITY TEMPLATES FOR FUNCTIONAL, DESIGN, AND TEST SPECIFICATIONS

Oracle has standard templates for functional, design and test specifications. These all include sections for security, and the core security team typically reviews any specification that includes security functionality.

Test specifications include details that facilitate the development of appropriate tests to validate security mechanisms. For example, buffer overflows are very common security vulnerabilities; approximately 80% of published security vulnerabilities are buffer overflows. Rather than just tell developers to test boundary conditions, the test specification templates include specific examples of how to check boundary conditions. Note that buffer overflows are particularly difficult to stamp out even though they have been well understood since the 1960’s. We are exploring code-scanning tools to better detect buffer overflows and other kinds of common security mistakes in addition to refining our standards and templates.

We put detailed security requirements into testing templates to make it as easy as possible for developers to avoid common security mistakes, by testing for potential error conditions. Hackers only have to find *one* vulnerability to obtain notoriety; developers need to close *all* vulnerabilities that come to Oracle’s attention. Having detailed test specifications makes it easier for developers to build Unbreakable code.

SECURITY REGRESSION TESTS

Regression tests — which are required for security evaluations — not only validate that security mechanisms work properly, but also validate that new features do not break current security functionality.

Oracle has a specific suite of security tests included in our regression tests that are run every day for the core database server. Oracle’s development environment is expanding so that up to twenty sets of regression tests can be run every day. This will make it easier to find security issues more quickly.

Oracle also expands regression tests when we identify and fix new security vulnerabilities. For example, Oracle incorporated 13,000 new regression tests into the development environment as a result of a buffer overflow found in the code base that constitutes most commercial Lightweight Directory Access Protocol (LDAP) implementations. If you cannot avoid all security mistakes, you need to ensure that you don’t make the same ones twice.

CENTRALISED SECURITY FUNCTIONS

Oracle has a centralised security group who has the responsibility and authority to:

- Provide core security routines used by multiple development groups within Oracle
- Drive security directions across the Oracle product stack

The core security group resides in Server Technologies, the development group that has the product responsibility for the Oracle9i Database Server and the Oracle9i Application Server that together form the core security platform used by all products in Oracle.

For example, Oracle has common libraries used for encryption, including the algorithms themselves and secure key generation, and a common Secure Sockets Layer implementation.

The reasons for security centralisation are multiple. First of all, security is not easy to do well; it is best to have a core group of security experts, rather than require every developer to be an expert on *all* aspects of security, especially the

aspects of security (like encryption) that are easy to get wrong and hard to get right. A second reason is economies of scale, i.e. there is no reason for every development group requiring Secure Sockets Layer to build their own SSL libraries. It is better to have a core set of well-tested and optimised security routines than large numbers of routines that may not work together and may be of varying degrees of quality and assurance.

Security centralisation also benefits security evaluations, since Oracle is able to validate (e.g. through a FIPS-140 evaluation) core encryption routines used by multiple products. This provides a level of assurance to *all* products using the libraries.

VULNERABILITY HANDLING

Unfortunately, not even the most stringent secure development process ever results in bug-free software or even security bug-free software. Oracle is no exception.

Oracle's commitment to Unbreakable software includes appropriate handling of significant security vulnerabilities, in order to protect our customers' systems. Our response to these vulnerabilities is twofold:

- Aggressive and responsible handling of significant security vulnerabilities, to include patching of the vulnerability as quickly as possible on all affected releases and platforms, as well as customer notification by issuance of *security alerts*
- Review of the vulnerability against our development processes to determine how we can avoid similar vulnerabilities in the future

Oracle notifies our customer base of significant security vulnerabilities through *security alerts*: a short write-up describing the vulnerability, with workarounds and patch information, that we post to Metalink (<http://metalink.oracle.com>) and to OTN (<http://otn.oracle.com/deploy/security/alerts.htm>). We may also distribute to the larger security-aware community through channels such as the Information Technology Information Sharing and Analysis Center (IT-ISAC) or the Carnegie Mellon Computer Emergency Response Team (CERT).

Typically, security alerts are issued for vulnerabilities that have most of the following characteristics:

- The vulnerability exposes a serious security hole (e.g. an unprivileged user can assume privileges he is not entitled to, or a regular user can become SYS or SYSDBA)
- The vulnerability is widespread, encompassing multiple releases, and/or multiple operating systems
- The vulnerability is relatively easily exploited. In the past, this meant that a not-very-knowledgeable user could exploit it; with the advent of the Internet, almost anyone can exploit security holes because of the increase in hacking skills, hacker forums, and the widespread proliferation of hacker tools
- There is no defence or only limited defence against it
- The vulnerability is discovered in supported Oracle products

Oracle has a unique set of challenges in dealing with security vulnerabilities due to the number of platforms we support and the number of releases of products we also support. Ideally, we patch significant vulnerabilities on *all* affected platforms prior to notifying customers through an alert. This allows all customers — regardless of release or platform — to protect their systems. Occasionally, we will post alerts in advance of all patch completion, for example, if the vulnerability has already been made public on the Internet. In this case, we expedite completion of patches as much as is possible.

Oracle believes that all customers deserve the same high level of security protection. Accordingly, we do not provide advance notice or insider information on security vulnerabilities to selected or favoured groups of customers. All customers have sensitive information that is as worthy of protection as any other customer's sensitive information.

Even Oracle's own IT infrastructure department is notified a mere day or two before patches are posted publicly so that we have time to patch our own systems before the alert is posted. There are other reasons we do not share advance information with even our most security-aware customers, such as the U.S. government, the primary reason being the Freedom of Information Act (FOIA). Any information we release to the government (in advance of the information being made public) could be released to a hacker via a Freedom of Information Act request. FOIA exemptions as currently written for sharing proprietary information may not adequately protect against disclosure of

such information; therefore, Oracle cannot make such information available, even to government entities, in advance of the general public.

Ultimately, we subscribe to the “Security Golden Rule”: do unto the security of customers’ systems as you would have done to the security of your own systems. We treat customers’ systems as if they were our own, because they are our own.

SECURITY DESIGN REVIEWS

As developers have become more aware of security requirements on them in the form of secure coding guidelines, security sections to complete in their functional, design and test specifications, or the availability of centralised security functions, they have turned to Oracle’s security assurance group for assistance to understand what is required of them. From such requests, and from issues that can arise in security checklists that they have to complete before product release, a security design review can arise or be suggested. This is usually an informal meeting in which a product’s entire architecture is explained for the security reviewer who will then focus in on security relevant aspects of the design, probing the developers for their secure design intentions. When issues arise, the reviewer is able to give advice, specify the security requirements that the product has to meet, and explain what vulnerabilities the product might be exposed to if the design is not changed to address those potential vulnerabilities. This key activity has now avoided countless security vulnerabilities from even seeing the light of day, and doubles as an education exercise in security awareness to developers who learn to avoid general secure design and coding errors in the future.

PRODUCT SECURITY RELEASE CHECKLISTS

The culmination of the experience in introducing all the above security assurance measure is the development of security checklists that are now part of our release process. Every line item owner on the product bill of materials must complete a security checklist designed to ascertain whether the product complies with secure coding standards (as well as avoiding the top fifteen or so common security mistakes). The checklists also include default configuration requirements; for example, making the file permissions on installation enforce “least privilege” considerations rather than being wide-open (e.g. 777 on UNIX systems) on installation.

In several cases, security checklists have raised issues about an underlying security implementation that has resulted in development changes before the product shipped and, in some cases, prompted a security design review.

The final release criteria is to ask whether Oracle is willing to introduce a delay in order to correct security vulnerabilities prior to public release of a product? Significant security issues *by definition* are “showstoppers”; we will stop product release to correct them. As stated earlier, it is “pay now” by delaying a release or “pay later” by patching the issue across multiple platforms with the resulting inconvenience to our customers.

As with other parts of the secure development process, Oracle continues to refine the security checklists to be more useful to developers and to ensure greater security. With each release, the requirements for security release get more stringent and incorporate latest “lessons learned” from security assessments and reported vulnerabilities.

For example, Oracle learned from our “ethical hacking” that our own administrators did not always change default passwords on database installations. While our documentation advised customers to change default passwords, we realised that we needed to make it easier to be secure. (Database administrators, after all, habitually have too much to do.) Accordingly we changed the default installation of the Oracle9i Database to lock and expire passwords on almost all default accounts.

Oracle’s Unbreakable commitment means making products progressively more secure by default, so that products are acceptably secure out-of-the-box, with minimal additional action by administrators. Even reported “vulnerabilities” which are actually configuration issues are candidates for generating a development change. The more Oracle does automatically to secure a product, the less the administrator, who seldom gets to read all the security advice in the documentation, has to do.

HOW TO BE UNBREAKABLE

Some of the experience of 14 years of security evaluations, of our security assessors, our ethical hackers, and the resulting secure product lifecycle, some of these assurance measures, those which can be released considering the “need to know” ethic, have been spelled out in two Oracle white papers. These are *A Security Checklist for Oracle9i*

available at http://otn.oracle.com/deploy/security/oracle9i/pdf/9i_checklist.pdf, and *Secure Configuration Guide for Oracle9iR2* available at http://otn.oracle.com/deploy/security/oracle9i/pdf/9iR2_checklist.pdf. These contain guidelines on how to secure your Oracle9i and Oracle9i Release 2 Database Server installations, revealing a collection of typical insecure configurations that Oracle's ethical hacking group regularly exploits to break into our own products and our own web sites. If Oracle and its own DBAs suffer from occasional insecure set-ups of Oracle products, you can be sure that your own DBAs have made the same mistakes on at least a few of your critical servers. A similar guide for Oracle9iAS Release 2 will be available soon.

Of course Oracle is not satisfied with just documenting the most common insecure configurations typically observed. This information is fed back to our product developers to ensure that future releases are secure "out of the box" so that a DBA need do nothing to be confident of having a secure configuration. Each release of the Oracle Database Server will become more secure by default to the point where a secure configuration guide is almost unnecessary, though we will probably continue to produce something that is more of an advisory document, warning about the dangers of particular configuration changes away from the secure default.

STAYING UNBREAKABLE

From time to time, Oracle issues a security alert that advises customers to apply a security patch. These need to be taken seriously because no one can be sure when a malicious real world hacker will write some code to exploit vulnerabilities in unpatched installations. Oracle emails all Metalink customers whenever a security alert is issued or modified. These security alerts are published on Oracle Technology Network at <http://otn.oracle.com/deploy/security/alerts.htm>, and even non-customers can sign up on this page to receive emails about new security alerts.

ASSURING UNBREAKABLE ORACLE9I AND ORACLE9IAS

This paper has demonstrated the significant changes that have come about in Oracle's development practices as a result of security assurance measures over many years. The secure product lifecycle fills the gap which security evaluations and security assessments, typically post release assurance measures, leave. Together these three measures provide a complete picture of security assurance, the technical proof behind Oracle's marketing campaign's headline – Unbreakable. The promise of Unbreakable has ensured that Oracle9i and Oracle9iAS have been built to be the most secure database and application server products that Oracle has ever developed, and will ensure security grows stronger in every product release to come.