Unbreakable: Oracle's Commitment to Security

An Oracle White Paper
February 2002

ORACLE

Why Unbreakable?

Beginning in November 2001, Oracle began a marketing campaign: Unbreakable. The security portions of the campaign reference Oracle’s 14 independent security evaluations (described below *in What is Information Assurance?*).

Such a bold statement raises a number of questions:

- *How can anyone claim to be Unbreakable?* Security professionals often say that security is a process, not a result. Also, every software product has bugs, and some of those are security bugs.
- *Why would anyone claim to be Unbreakable?* Security professionals don’t like to be hacker targets, and some hackers wonder if this campaign is a ploy by Oracle to get free “security research.”
- *What does Unbreakable really mean?* How can vendors and customers know how secure a product is, and whether the security is sustained from release to release? Building secure software is so hard, why even bother, much less try for Unbreakable?

What is Unbreakable?

Unbreakable is Oracle’s commitment to our customer base to build, deliver and support the most secure mission-critical software in the business:

- Unbreakable is our security commitment to our customers yesterday, today and tomorrow.
- Unbreakable is not a three week or even six-month marketing campaign; Unbreakable builds upon ten years of building provably secure databases.
- Unbreakable extends the secure development process of our core products across the entire Oracle product stack..

What’s the Opposite of Unbreakable?

At first blush, “Unbreakable” seems to be the act of a marketing department run amok. After all, what company would wish to be the target of hackers, many of whom are extremely clever and persistent? And, after all, no product is perfect, is it?

Cynics and naysayers should consider the alternative to Unbreakable, which is the degree of comfort far too many vendors have with the insecurity of the software they foist on their customer base:

- "Kind of secure" is not good enough.
- "As secure as we can make it and release product every six months" is inadequate.
- "Secure once you have applied the last twelve security patches" is a disgrace.

Most vendors would never dream of claiming they are Unbreakable, because they put minimal effort into securing their software and — by extension — their customer's systems. They don't care, and it shows. They don't care, and it costs their customers in multiple ways:

- increased "hacker insurance" premiums from running unsecurable software
- billions of dollars in damages from viruses caused by neglect of basic security mechanisms
- increased costs from applying patch after patch in a vain attempt to secure products for which security was an afterthought

Unbreakable is a commitment that Oracle gladly makes for multiple reasons:

- *importance of security*. September 11 was a wake-up call for information security as much as physical security. The ultimate terrorist attack is one in which our critical infrastructure is brought down via a cyberattack by an unknown person from an unknown device in an unknown place.

- *business reputation.* Oracle's very first customers were among the most security- aware in the world. Twenty-five years after our founding, our core customer constituency still includes the most security-aware customers in the world. Our good name in security is our stock in trade, and ours to lose if we are not Unbreakable.

- *cost avoidance*. The saying "pay now, or pay later" is especially applicable to security. It is far cheaper for our customers — and for us — that Oracle builds security correctly the first time, than to try to patch it after the fact.

Those who deride Unbreakable as a marketing gimmick should ask the question: why doesn't *every* vendor commit to make their security Unbreakable?

Unbreakable sets a standard for all vendors of information technology to follow: even if Oracle does not do everything perfectly today, but builds security better than our competition, and customers purchase our products on that basis, security will improve in the industry. Unbreakable is not only our commitment to our customer base; it is a commitment that, by so doing, we will improve security for the entire industry.

What are the Elements of Unbreakable?

A key element in Unbreakable software is independent measures of assurance — that is, third-party attestation through a formal security evaluation that our product security claims are valid. Independent measures of assurance are a key element in Unbreakable because how you build your products — from a security standpoint — is ultimately more important than what you build, and the "what" can *only* be validated if you know "how."

A second element in Unbreakable is a commitment to a secure product lifecycle. Assurance is an important part of creating and maintaining that lifecycle; indeed, to establish the correctness of your security, you need a secure product development process that must be repeatable, to ensure that you do not break old security mechanisms by adding new features. As John Pescatore of Gartner Group has said: "Security cannot be 'tested' into software; it must be a high priority from the start — during requirements analysis and planning."

The following sections elaborate on Oracle's information assurance measures and secure product lifecycle.

What is Information Assurance?

The growth of the Internet has increased the importance of information security. "Webification" of the enterprise increases the amount of information stored, managed, and accessed in databases, and the number of products and tools accessing those databases. At the same time, the Internet has increased the *rate* of software product development as customers move to "web-ify" their enterprises.

In this environment, many vendors add lots of features and functionality in products as quickly as possible, with little regard for the security of those features. The vendors may conduct a cursory inspection of security before the product ships, but the main factor in the product release is "How much new functionality can I stuff in before the release ships?" not "How can I ensure that my customers' systems are secure with what I build?" Paradoxically, the requirement for strong security correctly implemented is increasing precisely as more time-to-market forces conspire against it. Web-ifying the enterprise creates more security risk, both because the newest (and probably least secure) products are protecting the more mature back-end data store, and because the time-to-market pressures are greatest for web-facing products.

In this environment, the requirement for information assurance — that is, proof that the security mechanisms of a product are correct and well-formed — is paramount. Otherwise, customers are at the mercy of vendors' marketing departments: *all* vendors claim their products are secure, even those who issue security alerts every 2 ½ days.

The main vehicle for substantiating vendors' security claims are independent (that is, third-party), formal security evaluations against international criteria. These criteria can be thought of as definitions of "what do you mean when you say you are secure." Only an independent security evaluation can attest to the correctness of a vendor's security claims, since all vendors claim their products are secure. Putting it differently, formal security evaluations are a way of vendors "putting their money where their mouth is" regarding security claims.

Just as many consumers will not purchase a product or device without an Underwriters' Laboratory™ rating or without consulting Consumer Reports, customers should not purchase mission-critical software without an independent security evaluation.

While there have historically been a number of country-specific evaluation criteria, the trend in the past few years has been towards internationalization. For example, the Common Criteria is an International Standards Organization (ISO) standard (15408); multiple countries not only subscribe to the Common Criteria but, through mutual recognition, will accept Common Criteria evaluations performed in other countries. Today, Oracle only performs Common Criteria evaluations for our server products, and Federal Information Processing Standard (FIPS)-140 evaluations, which validate cryptographic modules.

Independent measures of information assurance are also required to sell into the US 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. In a post-September 11 world, there will be few, if any, waivers granted by the National Security Agency (NSA) from NSTISSP #11 requirements.

Oracle is the undisputed market leader in formal security evaluations, with fourteen independent security evaluations against every major worldwide criteria over the past ten years. The security claims of the Unbreakable *specifically* rest upon independent measures of assurance provided through 14 security evaluations of the Oracle data servers against every major worldwide security evaluation criteria, including the Common Criteria (ISO-15408), the de facto worldwide evaluation standard.

There are multiple benefits to Oracle 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 provably 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 repeatably 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 fourteen 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.

Oracle also does less formal product “risk assessments’ which are described later in this paper. Over time, we expect that component “risk assessments” will migrate to more formal evaluations as security-relevant components in products mature to a longer product lifecycle.

The Secure Product Lifecycle

Beyond the measures of assurance outlined above, 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 one’s corporate DNA, wired into the fabric of the organization at every stage in product development and delivery.

Oracle’s secure development process includes all of the following elements:

- Secure coding standards
- Security templates for functional, design, and test specifications
- Security regression tests
- Centralized security functions
- Product risk assessments and formal security evaluations
- Product release criteria for security

Secure Coding Standards

The only way to build and deliver secure products is for each developer to assume personal responsibility for delivering secure product 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 practice. For example, Oracle has long had ‘C’ coding standards that developers are trained to follow; code reviews by development managers review code against the ‘C’ 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 cannot 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 centralized 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 developer 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 on security coding standards and the coding standards are subject to continuous process improvement. Oracle incorporates "lessons learned" from our ethical hacking efforts, reported security vulnerabilities, and from information gleaned by participation in industry information sharing forums (e.g. IT ISAC) into our 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 risk assessments until requesting teams have "self-assessed" by reviewing the coding standards and other security checklists.

Another reasons we enforce coding standards is the economics of "pay now or pay later." Oracle runs on multiple operating systems, and supports 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. To 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 customer's 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 2 ½ days; customers are constantly patching their systems for security flaws. Ultimately, they cannot keep up, and they are then victimized by the latest virus that exploits non-patched security holes.

Security Templates for Functional, Design, and Test Specifications

Oracle has standardized templates for functional, design and test specifications. Standardized templates include sections for security, and functional specifications which include security typically are reviewed by the core security team.

Test specifications include details that facilitate developing appropriate tests to validate security mechanisms. For example, one common security vulnerability is buffer overflows (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: 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 *all* potential error conditions. Hackers only have to find one vulnerability to obtain notoriety, developers need to close *all* vulnerabilities. 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. Oracle runs a full suite of regression tests every day for the core database server, and is expanding our development environment so that we can run up to twenty sets of regressions per 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, as part of continuous process improvement. 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.

Centralized Security Functions

Oracle has a centralized 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, which has the product responsibility for the Oracle9*i* Data Server and Oracle9*i* Application Server, which 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 as well as secure key generation, and a common Secure Sockets Layer implementation.

The reasons for security centralization 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 optimized security routines than large numbers of routines that may not work together and may be of varying degrees of quality and assurance.

Security centralization also benefits security evaluations, since you are 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.

Product Risk Assessments and Formal Security Evaluations

As described earlier in this paper, Unbreakable security is all about information assurance: ensuring that security mechanisms are correctly implemented and well-formed across the product stack.

Oracle remains committed to independent security evaluations of our core server products, building upon our 14 evaluations to-date:

- Oracle Label Security is in evaluation at EAL4 against the Common Criteria, expected to complete in the first half of 2002.
- Oracle9*i* Database Server and Oracle9*i* Label Security release 2 will be submitted for evaluation against the Common Criteria at EAL4.
- Oracle9*i* Application Server (version 2) will be submitted for a FIPS-140 evaluation.

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 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 a year per server product, which is about two product release cycles of 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 no point to spending $1,000,000 to evaluate a product that is obsolete by the time the evaluation is completed.

To augment our commitment to formal evaluations, Oracle has expanded our assurance group’s activities to include risk assessments on products for which formal evaluations are not currently feasible, and various “ethical hacking” activities. Risk 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 risk 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 of 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.

Knowledge we gain from risk 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 and their security technologies are better understood, that they will be subjected to formal evaluations. Risk assessments now conducted on the main security components of our products will likely be supplanted by formal evaluations in time.

Product Release Criteria for Security

Oracle has developed security checklists which are 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 has complied 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 some cases, security checklists have raised issues about an underlying security implementation that has resulted in development changes before the product shipped.

The final release criteria is; is Oracle willing to delay product shipments in order to correct security vulnerabilities prior to shipment? Significant security issues *by definition* are “showstoppers;” we will stop product shipment to correct them. As stated earlier, it is “pay now” (in delaying a release) or “pay later” in patching the issue across multiple platforms and in 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 releasability get more stringent and incorporate latest “lessons learned” from risk 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 realized that 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 Oracle9*i* 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 one can do automatically to secure a product, the less the administrator (who seldom gets to read all the security advice in the documentation) has to do.

Security Vulnerabilities

An important aspect of a secure product lifecycle is vulnerability handling. Unfortunately, even the most stringent secure development process may not result in bug-free software or even security bug-free software.

Oracle's commitment to Unbreakable software includes appropriate handling of significant security vulnerabilities, in order to protect our customer's 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 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 TechNet (http://technet/deploy/security/alerts.htm), and which we may also distribute to the larger security-aware community through channels such as Internet Security Systems' (ISS) X-force bulletins or through the Information Technology Information Sharing and Analysis Center (IT-ISAC).
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 Internet, almost anyone can exploit security holes because of the increase in hacking skills, hacker forums, and hacker tools
- there is no defense or only limited defense against it
- the vulnerability is discovered in currently-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 work with all speed to complete patch sets as soon as 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 favored groups of customers. All customers have sensitive information which is as worthy of protection as any other customer's sensitive information.

Even Oracle's own IT department is notified a mere day or two before patches are posted publicly (so that we have time to patch our own systems by the time the alert is posted). There are other reasons we do not share advance information with even our most security-aware customers, such as the US 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 John Q. Hacker via his making a Freedom of Information Act request. In the absence of a FOIA exemption for sharing information about security threats, Oracle cannot make this information available even to government entities in advance of the general public.

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

Conclusion

There are no security magic bullets, but Oracle stands behind the security of our products, and the security of our customers' systems. As we run our own company on Oracle, we have a vested interest in delivering Unbreakable software. Unbreakable builds on the strength of some 20 years of our building systems for the most security-conscious customers in the world — including intelligence agencies and the Department of Defense — and the assurance afforded by 14 independent security evaluations. (Our nearest competitors have 0 and 1 evaluations, respectively. Why are *they* not investing in a secure product lifecycle and provably secure software?)

Unbreakable software is a long term commitment, already in-process, to extend the same development methodology and assurance measures to every Oracle product. Today, no customer who is serious about security runs their database on anything but Oracle. Our commitment is to extend our database best-of-security breed to every product Oracle develops and delivers: Unbreakable everywhere.

Marketing campaigns may come and go, but Unbreakable is our security commitment to our customers yesterday, today and tomorrow.

For More Information

NSTISSP #11:

http://www.nstissc.gov/Assets/pdf/nstissp_11.pdf

Common Criteria homepage:

http://www.commoncriteria.org

NIAP homepage:

http://niap.nist.gov
http://niap.nist.gov/niap/library/index.html

The UK scheme homepage:

http://www.cesg.gov.uk/assurance/iacs/itsec/index.htm

UK certified products list:

http://www.cesg.gov.uk/assurance/iacs/itsec/cpl/index.asp

The TCSEC evaluated products list:

http://www.radium.ncsc.mil/tpep/epl/

FIPS 140-1 and 140-2 Cryptographic Module Validation List:

http://csrc.nist.gov/cryptval/140-1/1401val.htm

Unbreakable
February 2002
Author: Mary Ann Davidson

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle Corporation provides the software
that powers the internet.