Global Information Assurance Certification Paper

"Deploying a website built using Oracle9*i*AS Portal"

GIAC Security Essentials Certification – Version 1.4b Option 2
Steve Coates

Abstract

This paper is a case study of the deployment of a website built using the Portal component of Oracle9*i* Application Server (Oracle9*i*AS) in 2001. It has been submitted as the practical assignment for GSEC certification (Version 1.4b, Option 2).

The paper describes the scenario and the product, Oracle9*i* AS (Standard Edition) Release 1 for Windows NT 4.0, before performing a high-level risk analysis of the website. The architecture implemented is discussed in terms of risk. The paper also identifies the security vulnerabilities discovered with Oracle9*i* AS during the six-month development period and the steps taken to harden an 'out-of-the-box' version.

Since the case study concerns a production system, certain details have been omitted or sanitised.

1. Before Snapshot

1.1 Background

At the end of 2000, my company decided to bring its hosted website in-house. To do this it formed a new department, E-Commerce, sitting between Sales & Marketing (as the major providers of content) and Computer Services (who would take over running of the operational website). As one of the staff recruited from the Computer Services Department, my role was to lead former colleagues in the Technical Development Team to the implementation of a suitable web infrastructure.

Following a successful trial, we chose Oracle9*i* AS Portal as the key tool for us to build and content manage the new website. Since the Computer Services Department had certain preferred manufacturers and resellers, many other building blocks of the website were already identifiable; e.g., operating system (Windows NT 4.0), server hardware (Compaq), firewall software (Check Point), database software (Oracle), network provider, etc.

1.2 Description of Oracle9iAS Portal

"Oracle9*i* Application Server" (Oracle9*i*AS) [1] is an application server supporting both Java 2 Enterprise Edition (J2EE) and PL/SQL, of which one major component is "Oracle9*i*AS Portal".

Oracle9*i*AS Portal is intended as a common, integrated starting point for accessing data stored in an Oracle database (including files, images and applications) for both intranets and public websites. The majority of development is done via its wizard-based system, but it includes Java and PL/SQL development kits for more complex work. It comes with its own built-in caching.

1.3 Business benefits of using Oracle9iAS Portal

Although (as WebDB) the product had started out as a way of leveraging database content to provide a simple intranet, it encouraged us to be one of the first companies to use it to host our own dynamic website.

Oracle9*i*AS Portal offered benefits to all areas of the business. For example, it meant that the Sales & Marketing Director could divide up the website and give different content responsibilities to different managers on different teams. Those content managers, without additional tools or knowledge of HTML, could use wizards to deploy simple applications in response to changing requests from the business. Hundreds of items (such as forms, reports, charts and lists of values) could simply be picked,

configured and used. Additionally, support for over 20 languages plus Portal Single Sign-On (SSO) with groups and permissions promised a website with one-to-one personalisation for our customers.

It offered the Computer Services Department a scaleable deployment architecture, starting from just a single Windows NT server and with portability to other operating systems. It also meant software support from a single vendor.

The E-Commerce department gained a database-driven website with ease of management and control for the overall environment and with support for Java, XML, PL/SQL and Perl.

1.4 Editions of Oracle9iAS

The product was available for download from the Oracle web site in versions for Windows NT/2000, Linux (Intel) and Solaris (SPARC). The two main editions were Standard and Enterprise. [2]

Standard Edition comprised the following components:

- Oracle9*i*AS Portal
- Oracle HTTP Server *Powered by Apache* (OHS)
  (Apache 1.3 and Apache JServ with mod_jserv, mod_plsql, mod_ssl, mod_perl, mod_ose, Perl Interpreter and JDK)
- Oracle Enterprise Java Engine
  (Support for Servlets, Java Server Pages, Enterprise Java Beans, Entity Beans and Business Components for Java (BC4J))
- Oracle9*i* File System
  (An Internet file system presenting documents and media as files and folders for users to access through Windows, HTTP, FTP and e-mail)
- Oracle9*i*AS Developer Kits
  (Developer kits for XML, database client and LDAP client)
- Oracle Enterprise Manager (OEM)
  (The base version of the browser-based management environment)

Enterprise Edition extended Standard Edition with the following components:

- Oracle9*i*AS Web Cache
  (Web and database caching for increased performance)
- Oracle9*i*AS Discoverer
  (Browser-based access to Oracle Discoverer for ad-hoc database queries)
- Oracle9*i*AS Forms Services
  (Browser-based access to Oracle Forms applications)
- Oracle9*i*AS Reports Services
  (Browser-based access to Oracle Reports)

Standard Edition contained the components sufficient to build our public website.

1.5 Product Versions

The table below illustrates how Oracle9*i*AS itself was a comparatively new product, first released in August 2000. Oracle9*i*AS Portal had a longer pedigree, as the successor to "WebDB 2.2". Although version 3 of WebDB had been anticipated, it was actually the Early Adopter edition of Oracle9*i*AS Portal (3.0.6) that appeared in September 2000. The first production version (3.0.7) was released at the end of 2000.

	Server	WebDB	Portal	iAS
Jul 1998	Oracle8 (8.0.5)			
Oct 1998				
Feb 1999		WebDB 2.0		
Mar 1999	Oracle8*i* (8.1.5)			
Apr 1999				
Jul 1999		WebDB 2.1		
Sep 1999	Oracle8 (8.0.6)			
Oct 1999				
Dec 1999		WebDB 2.2		
Jan 2000	Oracle8*i* (8.1.6)			
May 2000				
Aug 2000				IAS 8*i* 1.0.1
Sep 2000			9*i*AS Portal 3.0.6	
Nov 2000	Oracle8*i* (8.1.7)			9*i*AS 1.0.2
Dec 2000			9*i*AS Portal 3.0.7	
Mar 2001				9*i*AS 1.0.2.1
Apr 2001			9*i*AS Portal 3.0.8	
May 2001			9*i*AS Portal 3.0.9	9*i*AS 1.0.2.2

Table: Oracle Product Release Dates for Windows NT

Our installation (at the start of 2001) therefore baselined with these product versions:

- Oracle8*i* Server 8.1.7.0.0
- Oracle9*i*AS Portal 3.0.7
- Oracle9*i*AS 1.0.2

1.6 High-Level Risk Analysis

1.6.1 Business View

The business view was that, once the website was live, the three risks were reputation, revenue and legal. Hackers or rivals must not compromise the privacy of customers' personal data (confidentiality). Outsiders must not be allowed to take over content management of the site (integrity). The website must not be down during TV and radio advertising campaigns (availability). Single loss expectancy could easily have exceeded the total cost of building the website.

We also had the business issues (carrying 100% risk) of lack of experience and budget. We did not have sufficient security experience within the company. We did not have sufficient Java experience within the company to complete the custom coding and integration work by ourselves. Since the product was so new, no-one could have had any real-world experience of deploying a website built using Oracle9*i* AS Portal. At the start of the project we did not have sufficient budget for all the components on our wishlist.

1.6.2 Risks and vulnerabilities

The primary threat vectors [3] were:

- Outsider attack from the Internet
- Attack from malicious code
- Insider attack from our LAN
- Insider attack on the physical system

Apart from the business risks, the other risks were separated into network, host and application risks [4] (i.e. the Oracle software), as outlined below.

1.6.2.1 Network

- Attacks on our DNS.
- Denial of service (DOS) attacks.
- Eavesdropping of customers' personal and transaction data sent over the Internet.

1.6.2.2 Host

- The vulnerabilities of unhardened Windows NT servers.
- Hardware availability (requiring at least one working database server and one working webserver).

1.6.2.3 Application

- The vulnerabilities of the Oracle software, plus additional vulnerabilities introduced by patches or upgrades.
- Access via extraneous files (such as examples or feature demonstrations).
- Privileged access to Portal or to the database using default passwords (leading to exposure of customer data).
- Defacement of the website through external content management.
- Loss of the authoritative copy of the website.
- Software availability (requiring at least one active database server and at least one active webserver).

2. During Snapshot

2.1 Business Issues

2.1.1 Experience

We transferred risk for security oversight to a well-known security consultancy, which have remained our advisers since the build. I also attended the SANS Kick Start and Security Essentials tracks (New Orleans 2001).

Since we had little experience of Java web programming at the time, we worked with a local software house. The software house had used Java and JSPs to deliver a graphical system to a major public utility. In order to qualify them for our project, all three departments (E-Commerce, Computer Services and Sales & Marketing) made site visits to interview them and inspect their offices and development procedures. Since they were a small firm and a new supplier, we insisted on regular code drops including sources and documentation.

As the product was so new, Oracle pre-sales consultants acted as mentors and advisers throughout the project and we paid for Oracle consultants to work with the software house. We also made use of Oracle Support.

2.1.2 Budget

Since it was clear that the entire wishlist was not going to fit within budget, we costed both the ideal solution and a scaled down version. This “minimum compliant architecture” included the development and test systems, met the business needs and could be extended gradually toward the ideal solution.

For example, a major budget item was Oracle licences. Since Oracle9*i* AS was initially licensed by CPU power, we used a number of smaller webservers (with the option of a CPU upgrade and with room for a second CPU and more memory). This move also increased overall availability of the webservers.

2.2 Risks

2.2.1 Network

We transferred the risk for DNS hosting to our ISP and their DNS Support Team.

We chose to use a number of smaller firewalls to isolate the E-Commerce systems from both the LAN and the Internet. As well as reducing the costs of hardware and licensing, this gave us the opportunity to lock down a short, well tested security policy for the Internet-facing firewalls, whilst being able to customise and adapt a second

policy for the LAN-facing firewalls.

To help protect against DOS attacks, on the Internet-facing firewalls we disabled ICMP and HTTP PUT and we configured Check Point's protection systems against address spoofing and SYN flooding attacks. (On the webservers we used BlackICE to drop hostile HTTP connections.)

To protect sensitive customer data transmitted over the Internet, we installed 40-bit (rather than 128-bit) SSL certificates [5] on our Oracle9*i* AS servers.

2.2.2 Hosts

The Technical Development Team hardened all the Windows NT servers and installed BlackICE. They configured limited accounts for the Service Desk to run tape backups. The only web access permitted from the servers was for each webserver to surf itself.

Our design made extensive use of hardware RAID. We tried to avoid single points of failure and, indeed, the main Oracle database server was actually a hardware cluster with shared disks. All hardware and software items were put on maintenance and we established a small spares holding of common items (disks, fans, etc.). To ensure that the Service Desk would be alerted in case of device failure, Compaq Insight Manager was configured on the servers. Thus a console within the website could collect and log SNMP health data and send SMTP alerts to the email server on the LAN.

Load balancers did not make it into the "minimum compliant architecture". Instead we started off with round-robin DNS on our ISP's DNS servers: once a customer landed on a webserver (www), OHS would redirect them to that webserver by its own name (www1, www2, etc.). The business accepted the risk that if the first webserver was down the customer might not try a second time!

When we later had the budget for load balancers, we had two main choices: licence the functionality of Check Point's ConnectControl on the internet-facing firewalls or install "black-box" load balancers. Although ConnectControl could have done the job, we chose to install F5 servers running BIG/IP because they would be able monitor a range of services on each webserver to determine responsiveness (for example, both JServ on port 8007 and OHS on port 80).

To address disaster recovery (DR) for our database-driven website, we installed a half-system at the DR site (a pair of webservers with a server for an Oracle standby database) and a private leased line. In normal operation the webservers at the DR site ran from the database at the main site whilst the archive logs were trickle-fed to the Oracle standby database. This meant that we could survive the failure of all the webservers at one site or the other (i.e. either www1/3/5/7 or www2/4) and that during a DR event, we could open the standby database to run a service at reduced capacity. Had we decided to encrypt the Oracle traffic on the leased line to the DR site, we had

the choice of using either the feature set of our Cisco routers or, at extra cost, the functionality of the firewalls.

Only the Technical Development Team and the Service Desk had (unaccompanied) physical access to the computer rooms at the main site and the DR site.

2.2.3 Application

2.2.3.1 Installation and hardening

We took the opportunity to rehearse installing the application, not only on our own PCs but also with our development and test systems on the LAN. I had been noting relevant vulnerabilities to Oracle8*i* and Oracle9*i* AS: by January 2001 there were already eight (numbered V1-V8 at Appendix 1).

On the production kit, we performed our tailored Oracle8*i* and Oracle9*i* AS installations and hardened the application software, as described in the checklist at Appendix 2. Later on, all further patches and upgrades were similarly rehearsed and documented.

2.2.3.2 Access via extraneous files

As part of the hardening process we removed the demonstration files, help files and release documents from the web root of OHS, to prevent them being crawled and appearing in search engine listings. These demonstration files could have been targets for reconnaissance and later vulnerabilities, for example the Perl CGI, printenv, in the directory, oracle\iSuites\Apache\Apache\cgi-bin. However, during rehearsals we discovered that certain upgrades (for example, from Portal 3.0.7 to 3.0.9) restored these files.

2.2.3.3 Privileged access through default passwords

Users in WebDB had been Oracle database users: for example, logging into WebDB as SCOTT/TIGER meant that you had just logged into the production database as SCOTT. A major change with Oracle9*i* AS Portal meant that you logged into Portal as a lightweight SSO user, and then Portal itself logged into the database (as one of a number of controlled database users).

For the default portal name of PORTAL30, there were five database users whose passwords needed to be changed (both in the database and on the portal configuration page):

PORTAL30/PORTAL30
PORTAL30_PUBLIC/ PORTAL30_PUBLIC

PORTAL30_SSO/PORTAL30_SSO
PORTAL30_SSO_PUBLIC/ PORTAL30_SSO_PUBLIC
PORTAL30_DEMO/ PORTAL30_DEMO

We used the ISS Database Scanner for Oracle to review other weak and well-known database passwords. We also changed the default passwords for the two matching lightweight SSO superusers:

PORTAL30/PORTAL30
PORTAL30_SSO/PORTAL30_SSO

2.2.3.4 Passwords for content managers and customers

It would have been very simple to configure the SSO password security settings to demand long, alphanumeric password that expired frequently. Whilst it appeared ideal to impose this level of security on the content managers, the same password requirements would have applied to the thousands of customers registering with our website. To get around this, what we did was to have the customers log into a Java application (which handled their personal account details within the database) and have the Java application automatically log them into a group in Portal. Since each group login only had a slightly different read-only view of the same Portal content, exposure of the group passwords was not considered vital. However, we also configured OHS to deny access to certain Portal URLs from outside the LAN, ensuring that only content managers on the LAN could reach the pages for manual login.

2.2.3.5 Authoritative copy of the website

The beauty of the webservers was that they took their Portal content by logging into the Oracle database and their Java content from a single Java archive file (JAR or EAR). Backing up the webservers to CDROM meant a two-CD restore (one for Windows and Oracle9*i* AS, the other for the latest JAR or EAR file).

Having a number of Portal content managers on the LAN, the authoritative copy of [the Portal portion of] the website was whatever was currently stored in the production Oracle database. [6] What we did was to configure version control in Portal so that we could revert individual content items without resort to the database backup tapes.

2.2.3.6 Software availability

Having had problems with the reliability of external URL monitoring services for the webservers, we gave the Service Desk a PC that just showed a view of the monitoring screen for BIG/IP (to show which components on which webservers were responding).

We decided to configure the LAN-facing firewalls to permit the main OEM console on the LAN to communicate with the agents on the three database servers for database monitoring and alerting.

Had we wanted the OEM console to run batch jobs on any of the servers, then we should have had to assess the risk of creating special Windows NT accounts for their OEM intelligent agents. These accounts would have required additional privileges ("Logon Locally" or Administrator; "Logon as a batch job"; "Logon as a service"; member of the ORA_DBA group) and a password that did not expire.

During the development process, we were able to perform extensive load testing on the production systems to help us tune the Portal caching and to determine the maximum performance of the “minimum compliant architecture”.

2.3 Vulnerability assessment

Before going live, we inspected our security footprint. The domain name registrations gave little away, and nmap showed that we were only exposing ports 80 and 443 on the Internet-facing firewalls. We used various pieces of software to try URLs on the website, to inspect the cookies and session ids and to demonstrate that the IDS was functioning.

As part of this testing, I observed vulnerability V11 (Appendix 1). Simply by changing the flag at the end of a Portal URL (from “_mode=3” to “_mode=2”), a public user gained access to the navigator screen to browse all objects on the website, to view user log details and to export modules. I raised a fault with Oracle and made sure that the patch was backported to our version of Portal.

As a final confidence check, a team from our security consultancy ran a vulnerability scan of the website (uncovering nothing harmful).

3. After Snapshot

3.1 Live Website

When the website went live in the middle of 2001, the Computer Services Department adopted it. The Technical Development Team took on the responsibility for researching vulnerabilities and testing and applying further patches (including the upgrade to Portal 3.0.9).

Since launch the “minimum compliant architecture” has built up enough to include most items on the wishlist, including a strong, commercial load-testing tool fully compatible with Portal and regularly scheduled external vulnerability scans.

3.2 Additional Complications

We started out with a single firewall management console (with RAID and its own tape backup unit). Had we switched over to run from the DR site for an extended period back then, we would have been operational only until one of the firewalls needed its security policy changing.

The business has accepted that it takes time to load Java releases or reconfigure OHS because, the way we have locked down the infrastructure, staff have to visit each webserver at the main site and the DR site. The business has also accepted that content management is only possible from the LAN.

Oracle later changed the licensing structure for Oracle9*i* AS, basing it on the number of CPUs instead of their size or power. This meant that for us to improve performance, it became cheaper to replace a single-CPU webserver with a faster model than to licence the software for a second (slower) CPU or add an additional webserver.

3.3 Lessons Learned During Development

3.3.1 Working inside the envelope

One of the plus points of Oracle9*i*AS had been a one-click installation of the infrastructure of our website. However, any components, configurations or version upgrades outside this envelope were not supported by Oracle. Two such examples were the version of Apache (in OHS) and the version of J2EE.

Firstly, whilst the current version of Apache in February 2001 was 1.3.19, Oracle HTTP Server *Powered by Apache* (OHS) was still powered by Apache 1.3.12.0.0.a from

February 2000. A secondary issue was that we were unable to use third-party tools such as Comanche to help configure such an old version.

The second example was that we wanted to run some of our Java code inside Oracle9*i*AS but outside Portal and its Java development kit. We struggled to configure the Oracle Servlet Engine (OSE) for our needs and found that Oracle's JServ was too old to support the 2.2 Servlet API. Instead we had to supplement JServ (required by Portal) with Jakarta Tomcat (to run our Java code). Oracle caught up in a later release, introducing a new J2EE engine, Oracle Containers for Java (OC4J).

Conversely, if there were unwanted components inside the envelope we either had to configure them out or keep them patched and up-to-date (for example, the XSQL Servlet – V6 at Appendix 1). If we decided to configure them out, we had to be sure of the dependencies of other sub-components.

3.3.2 Support

As customers with paid support, we could use Oracle's Metalink website to search for patches and bugs and to raise electronic fault reports (iTARs). The first problem was that every single sub-component of Oracle9*i*AS was listed separately, so it meant regular trawls though Metalink to keep up-to-date. The second problem was that, once a bug was diagnosed, it would be fixed at a subsequent release. Only if a good business case could be made might Oracle backport the fix for an older release.

3.3.3 Transition to Live

If using a traditional development process, i.e. transferring the site from a development system through a test system to the live system, then each transfer required a multi-stage export/import sequence for the Portal content. We reported an iTAR for the process of exporting/importing folder portlets and sub-pages and Patch #1727108 was back-ported to Portal 3.0.7 for us in due course.

However, this import also had some unwanted side effects: firstly, ownership of all pages and sub-pages changed to the portal owner, default PORTAL30. Secondly, all lightweight SSO users had their personal details purged and their passwords reset to their username in uppercase. Had we been using SSO for our customers this would have been a major problem.

Since it was only possible to export/import between systems running exactly the same version of Oracle9*i*AS Portal, the development, test and live systems had to be kept in step.

An alternative process would have been to backup the whole database and move it across or to develop the site in-place, on the live system, using version control and

manual content release.

3.3.4 Resources for Web Developers

A Portal website is not like a standard website: content and structure are embedded in a proprietary environment. Web developers who are already familiar with tools like Dreamweaver and FrontPage are faced with foreign concepts (such as portlets, folders, regions, pages, categories and perspectives). So many books that promise "Oracle and the Internet" still refer to obsolete products, such as OAS and WebDB. The best source of information for Oracle9*i*AS Portal is the Oracle web site, with its manuals and whitepapers, discussion forums and online presentations.
After we had gone live, ideal reference was published

3.4 Reflection on SANS Material

Had I not attended the SANS training, it would have been much more difficult for me to have raised awareness of the security risks of hosting our own database-driven website. Particular aspects included an understanding of the threat, the practicalities of hardening Windows NT and the need for an IDS (even though we had firewalls).

Without this knowledge, backed up by concrete examples, it would have been impossible to justify the time and budget required for the security activities. We could then have ended up exposing servers to the Internet running, if not a standard build of Windows NT, then certainly an unmodified copy of Oracle9iAS.

References

[1] Oracle. “Oracle9*i* AS v1.0.2.2 Technical Info”. 2002. URL: http://otn.oracle.com/products/ias/techlisting.html (20 May 2003)

[2] Oracle. “Oracle9*i* Application Server (Oracle9*i* AS) Summary of Standard, Enterprise and Wireless Edition Features (v1.0.2.1)”. April 2001. URL: http://otn.oracle.com/products/ias/pdf/9iaspackaging.pdf (20 May 2003)

[3] The SANS Institute. Intrusion Detection: The Big Picture. New Orleans: The SANS Institute, 2001. 6-35.

[4] The SANS Institute. Information Security: The Big Picture. New Orleans: The SANS Institute, 2001. 1-26.

[5] Lowenthal, Bruce. “Best Practices in HTTP Security.” 05 June 2001. 12. URL: http://otn.oracle.com/product/ias/pdf/best_practices/security_best_practices.PDF. (20 May 2003)

[6] Allen, Julia H. The CERT Guide to System and Network Security Practices. Boston: Addison-Wesley, 2001. 114.

Appendix 1: Relevant Vulnerabilities (October 2000 - May 2001)

Vulnerability V1	Listener allows overwrite or append to any file.
Published	October 2000, Internet Security Systems (ISS) X-Force.
Exploit	http://xforce.iss.net/alerts/advise66.php
Solutions	Although fixed at Oracle8*i* 8.1.7.0.0, a flag in the "listener.ora" file ("ADMIN_RESTRICTIONS_<listener_name>") must be set to ON. Its default value is OFF. Oracle also recommend establishing the listener password in this mode of operation, where the default listener password is blank. http://technet.oracle.com/deploy/security/pdf/listener_alert.pdf http://xforce.iss.net/static/5380.php
Fixed at version	Oracle8*i* 8.1.7.0.0, but flag needs to be set.

Vulnerability V2	Configuration web page not password protected.
Published	**December 2000, Michal Zalewski**
Exploit	http://archives.neohapsis.com/archives/bugtraq/2000-12/0339.html http://archives.neohapsis.com/archives/bugtraq/2000-12/0372.html
Solutions	Oracle recommends restricting this to legitimate users using the "administrators=" setting in the wdbsvr.app file, so preventing public users from altering or deleting the Database Access Descriptor (DAD) from a browser as a denial of service. http://archives.neohapsis.com/archives/bugtraq/2000-12/0463.html http://technet.oracle.com/deploy/security/pdf/webdb_bugpost.pdf SAINT checks for this vulnerability (http://www.wwdsi.com/cgi-bin/doc.pl?document=vulnerability/Oracle_vulnerabilities).
Fixed at version	Oracle9*i*AS Portal 3.0.8

Vulnerability V3	Passwords visible in configuration file (wdbsvr.app).
Published	**(consequential)**
Exploit	The Database Access Descriptor (DAD), i.e. the database username and password of the portal owner and single sign-on owner, is stored in clear in the file, wdbsvr.app. Local access or directory traversal vulnerability may expose these details.
Solutions	Password is encrypted at 3.0.8
Fixed at version	Oracle9*i*AS Portal 3.0.8

Vulnerability V4	Injection of PL/SQL code or procedure calls into URL.
Published	**December 2000, Michal Zalewski**
Exploit	http://archives.neohapsis.com/archives/bugtraq/2000-12/0339.html http://archives.neohapsis.com/archives/bugtraq/2000-12/0372.html http://archives.neohapsis.com/archives/bugtraq/2000-12/0373.html http://archives.neohapsis.com/archives/bugtraq/2000-12/0395.html http://archives.neohapsis.com/archives/bugtraq/2000-12/0431.html http://archives.neohapsis.com/archives/bugtraq/2000-12/0439.html http://archives.neohapsis.com/archives/bugtraq/2000-12/0449.html

Solutions	One of: A. Revoke public access to procedures (such as OWA, SYS and DBMS) which can potentially execute user-specified SQL statements. B. Disable access to URLs which match certain criteria by adding rules to the plsql.conf file: <Location /pls/*/ABCD.*> SetHandler pls_handler **Order deny,allow** *Deny from all* </Location> where ABCD is the name of the procedure or synonym, and the rule must be repeated for all permutations of lower, upper and mixed case. http://technet.oracle.com/deploy/security/pdf/webdb_bugpost.pdf http://archives.neohapsis.com/archives/bugtraq/2000-12/0463.html SAINT checks for this vulnerability (http://www.wwdsi.com/cgi-bin/doc.pl?document=vulnerability/Oracle_vulnerabilities).
	Oracle Patch #1554571 for mod_plsql version 3.0.0.8.3 introduces a new configuration parameter in mod_plsql called “exclusion_list”. This parameter can be used to disallow URLs with specific formats from being passed to mod_plsql; by default it excludes URLs with special characters such as space, tab, newline, carriage return, single quote, and backslash. http://metalink.oracle.com/ http://archives.neohapsis.com/archives/bugtraq/2001-01/0206.html
Fixed at version	Oracle9*i*AS Portal 3.0.8

Vulnerability V5	mod_plsql error message cannot be configured.
Published	**December 2000, Michal Zalewski**
Exploit	Error messages from mod_plsql (rather than 404 pages) have been crawled, such that a simple search engine query locates the signature of WebDB, OAS and Portal installations: http://www.google.com/search?q=procedure+dad+environment+%22ora-06550%22+url http://archives.neohapsis.com/archives/bugtraq/2000-12/0339.html

Solutions	None
Fixed at version	

Vulnerability V6	**Oracle XSQL Servlet accepts external XML style sheets.**
Published	**January 2001, Georgi Guninski**
Exploit	http://www.guninski.com/oraxsql.html http://archives.neohapsis.com/archives/win2ksecadvice/2001-q1/0018.html
Solutions	One of: The new release of XSQL Servlet (1.0.4.1) can be obtained from Oracle Technology Network at http://technet.oracle.com/software/tech/xml/xsql_servlet/software_index.htm http://technet.oracle.com/deploy/security/pdf/xsql_alert.pdf http://archives.neohapsis.com/archives/bugtraq/2001-01/0364.html B. Patch the database up to 8.1.7.1.1 http://metalink.oracle.com/
Fixed at version	XSQL Servlet 1.0.4.1 (included in Oracle8*i* 8.1.7.1.1)

Vulnerability V7	**Directory traversal vulnerability in JSP/SQLJSP.**
Published	**January 2001, Georgi Guninski.**
Exploit	http://www.guninski.com/orajsp.html http://archives.neohapsis.com/archives/win2ksecadvice/2001-q1/0028.html
Solutions	An updated version of OJSP (1.1.2.2.0) is available on the Oracle Technology Network at http://technet.oracle.com/tech/java/servlets/index.htm http://technet.oracle.com/deploy/security/pdf/jspdocroot_alert.pdf http://archives.neohapsis.com/archives/bugtraq/2001-02/0239.html
Fixed at version	OJSP 1.1.2.0.0

Vulnerability V8	**JSP/SQLJSP allows unintended execution of .jsp in a similar directory path outside the web root.**
Published	**January 2001, Georgi Guninski.**
Exploit	http://www.guninski.com/orajsp.html http://archives.neohapsis.com/archives/win2ksecadvice/2001-q1/0028.html
Solutions	Ensure that the virtual path in a URL is different from the actual directory path when using Oracle Apache/JServ. Do not use the <servletzonepath> directory in "ApJServMount <servletzonepath> <servletzone>" to store data or files. http://technet.oracle.com/deploy/security/pdf/jspexecute_alert.pdf http://archives.neohapsis.com/archives/bugtraq/2001-02/0240.html
Fixed at version	Oracle9*i*AS 1.0.2.1

Vulnerability V9	Access is possible outside the web root if “FilePermission” is granted to <<ALL FILES>>.
Published	**February 2001.**
Exploit	http://technet.oracle.com/deploy/security/pdf/ojvm_alert.pdf http://archives.neohapsis.com/archives/bugtraq/2001-02/0255.html
Solutions	Oracle's recommended fix is to grant permission specifically to the document root path and not to the '<<ALL FILES>>' wildcard.
Fixed at version	

Vulnerability V10	Apache Listener stops responding after prolonged use of mod_plsql.
Published	March 2001, Oracle9*i*AS 1.0.2.1 Release Notes (NT/2000).
Exploit	Intermittently the Apache listener stops responding within two to three days after prolonged use of mod_plsql.
Solutions	Restart the Apache listener
Fixed at version	

Vulnerability V11	Portal user gains access to site structure.
Published	May 2001.
Exploit	Changing the "mode" parameter of a Portal URL, from http://www.portal307.com/servlet/page?_pageid=250&_dad=portal30&_schema=PORTAL30&_mode=3 to _mode=2 allows access to browse all objects on a site, view user log details and export modules.
Solutions	Patch backported to Oracle9*i*AS Portal 3.0.7 as patch #1764569 http://metalink.oracle.com/
Fixed at version	Oracle9*i*AS Portal 3.0.9

Compiled 18 May 2001.

Appendix 2: Oracle9*i*AS Portal 3.0.7 Configuration Checklist

1. Configure Oracle8*i* database
   - Set ADMIN_RESTRICTIONS_<listener_name>=ON in listener.ora file
   - Upgrade Oracle8*i* to 8.1.7.1.1 (including XSQL Servlet 1.0.4.1)

2. Configure Oracle9*i*AS
   - Install OJSP 1.1.2.2.0
   - Set administrators= flag in wdbsvr.app file
   - Edit plsql.conf file to deny external access to OWA, SYS and DBMS procedures and synonyms
   - Apply Oracle patch #1554571 (mod_plsql exclusion_list)
   - Apply Oracle patch #1727108 (export/import)
   - Apply Oracle patch #1764569 (mode=2 access to site structure)

3. Configure passwords
   - Change database passwords and settings
     - Alter well-known passwords of privileged users
       SYS/CHANGE_ON_INSTALL
       SYSTEM/MANAGER
       AURORAORBUNAUTHENTICATE/INVALID
       CTXSYS/CTXSYS
       DBSNMP/DBSNMP
       MDSYS/MDSYS
       ORDPLUGINS/ORDPLUGINS
       ORDSYS/ORDSYS
       OUTLN/OUTLN
     - Delete demo users
       ADAMS/WOOD
       BLAKE/PAPER
       CLARK/CLOTH
       JONES/STEEL
       SCOTT/TIGER
     - Alter default passwords of Portal users
       PORTAL30/PORTAL30
       PORTAL30_PUBLIC/ PORTAL30_PUBLIC
       PORTAL30_SSO/PORTAL30_SSO
       PORTAL30_SSO_PUBLIC/ PORTAL30_SSO_PUBLIC
       PORTAL30_DEMO/ PORTAL30_DEMO
     - Enable auditing of failed login attempts

- Update Portal Database Access Descriptor (DAD)
- Change Portal SSO passwords and settings
    - Change SSO passwords of Portal & SSO owners
      PORTAL30/PORTAL30
      PORTAL30_SSO/PORTAL30_SSO
    - Configure SSO password policy

4. Configure OHS (Apache)
    - Perl printenv
    - Demo files
    - Help
    - Documentation