Database Security Assessment

The first step toward complete database security and regulatory compliance

As the primary repository for the enterprise's most valuable information, the database is perhaps the most sensitive segment of the IT landscape. Many organizations are learning that database assets are vulnerable to both external attackers via Web applications and internal employees who take advantage of more direct privileges. Customer records, financial reports, and patient data are all at risk. In addition, compliance with regulatory requirements such as Sarbanes-Oxley (SOX), the Payment Card Industry Data Security Standard (PCI DSS), and others require organizations to perform database security assessments.

The purpose of this paper is to help organizations take the first step toward securing their databases through best practice security assessment. This paper outlines the elements of a best practice database security assessment process and where it fits within a complete database security lifecycle. It also includes brief introductions to Imperva's solutions in this area: the Scuba by Imperva Database Assessment Scanner and the SecureSphere Database Security and Monitoring Gateways.

What is Database Security Assessment?

Database security assessment is fundamentally a process that measures database risk at a point in time. The first element of risk is measured by evaluating a database's susceptibility to a series of known vulnerabilities and attack scenarios.

A vulnerability might be a best practice system configuration error such as a lack of a database password policy; a software coding error such as a buffer overflow in a procedure; or a privilege management error such as public access to a sensitive table. Each vulnerability identified is then rated by severity – low, medium, high, critical, etc. Finally, a report is generated that summarizes the results. A typical assessment summary, for example, charts the total number of vulnerabilities by severity (Figure 1). This summary is essentially a snapshot of overall risk that management can use to prioritize the steps required to improve database security. It tells security managers and database administrators which databases and which specific vulnerabilities need their attention first.

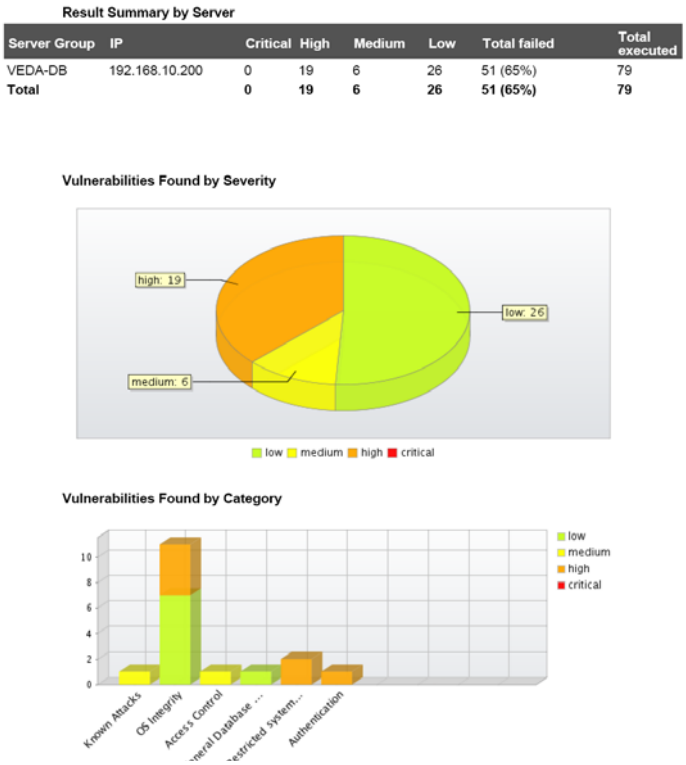

Figure 1: A Typical Database Assessment Summary

Best Practice Database Security Assessment

The fundamental design of a best practice database security assessment process includes consideration of the following four attributes:

- Impact on production systems
- Accuracy
- Efficiency
- Breadth of analysis

Impact on Production Systems

Many assessment processes attempt to identify vulnerabilities by mimicking the activities of an attacker. For example, an assessment may try to exploit a known buffer overflow vulnerability or use brute force to obtain valid access credentials. Such exploit techniques are common amongst automated Web server and network assessment tools – especially open source tools such as Nikto, Nessus, and Whisker. The problem with these methodologies is that they can cause downtime or damage the database if any exploit is successful. A buffer overflow simulation, for example, can crash a database.

Any chance of downtime or damage is obviously unacceptable in production environments. This fact makes exploit mechanisms appropriate for lab testing only. On the other hand, lab test results are not applicable to production databases. Vulnerabilities found, or more importantly, those not found in the lab may or may not exist in production. The database security assessment solution, therefore, should function without using actual exploits. Production databases cannot be put at risk.

Accuracy

Many assessments do not dig deep enough into available database information to validate the status of a given vulnerability. Consider the xp_sprintf buffer overflow vulnerability in Microsoft-SQL server (BID1204). xp_sprintf is an extended stored procedure that may be exploited by an attacker to crash the server or gain administrative privilege. There are two coarse approaches to assessment with respect to this vulnerability that lead to inaccurate results.

Exploit Approaches

Exploit-oriented approaches attempt to send data to xp_sprintf (despite the risk). Depending upon the response to the exploit, they report whether or not the server is vulnerable. However, the accuracy of this approach depends upon whether or not the database user account used by the assessment tool has EXECUTE privileges over xp_sprintf. A PUBLIC user may not have privileges to this account. Therefore, an exploit from a PUBLIC account fails and the assessment reports a not vulnerable result. However, a Web application may have privileges to xp_sprintf making the database server quite vulnerable – despite the original not vulnerable result.

Version-Only Approaches

Other assessment approaches simply check software version to evaluate vulnerability. In this case, SQL Server versions 6.5 SP4 and earlier suffer from this vulnerability. Such version-only assessment concludes that a server with version 6.5 SP1 is vulnerable. However, this approach ignores whether or not the stored procedure actually exists on the server. If the stored procedure has been removed, as best practices recommend, the assessment result should be not vulnerable.

Efficiency

Database security assessment should do more than provide a flat vulnerability listing. Such a listing alone is not actionable. Administrators could attempt to sequentially remediate each vulnerability on the list, but such a effort would be extremely inefficient. Some vulnerabilities require immediate attention, while others can wait or even be ignored. A more efficient assessment process prioritizes each vulnerability according to risk. With a prioritized risk analysis, a security manager or database administrator can develop an effective plan for remediation.

Another problem with flat listing database vulnerabilities is that the risks cannot be evaluated and prioritized by management. Database administrators, security managers, and internal auditors need assessment reports that communicate database security posture at an executive level. Among other uses, these reports are used to justify security budgets, report on the results of a new security process, and measure regulatory compliance. Therefore, database assessment solutions should integrate reporting capabilities that translate flat vulnerability data into executive-level risk analysis.

Breadth of Testing

Best practice database security assessment should include a breadth of tests that address each of the following areas.

- Known Platform Vulnerabilities
- System Configuration
- Privilege Management
- External Objects
- Regulatory Compliance

Known Vulnerabilities

Public vulnerability databases (Bugtrac, NVD, etc.) track thousands of known software vulnerabilities – including those that exist within databases and their underlying operating systems. Database security assessment should evaluate susceptibility to all such known vulnerabilities that are relevant to the target database software and underlying operating system. To understand how such vulnerability testing and prioritization may take place without actually simulating an attack, consider the following example.

BID 2041: Microsoft SQL Server / Data Engine xp_printstatements Buffer Overflow Vulnerability

Bugtrac ID number 2041 identifies a Microsoft SQL Server 2000 / Data Engine vulnerability in which an extended stored procedure is susceptible to a buffer overflow that may crash the system or enable execution of arbitrary code (i.e. a worm). To test for this vulnerability without actually sending data to the input buffer in question, the security assessment may apply the following process.

Test for Vulnerability

- Check the software version against those known to be vulnerable. Microsoft has issued a patch for this vulnerability.
    - If the patch has been applied, then the server is not vulnerable.
    - If the version is found to match known vulnerable versions, check for the existence of the xp_printstatements stored procedure. Database security best practices recommend the removal of all unnecessary extended stored procedures.
        - If xp_printstatements has been removed, then the database is not vulnerable.
        - If xp_printstatements exists, then the database is vulnerable.

Prioritize by Risk

- To prioritize the vulnerability by risk, list of users that have "EXECUTE" privileges on the xp_printstatements stored procedure is extracted.
    - If privileges are granted to a large number of users or PUBLIC (i.e. everyone), then the risk associated with this vulnerability is relatively high.
    - On the other hand, if privileges are granted only to SA (system administrators) then the risk associated with this vulnerability is low.

System Configuration

Database security is highly sensitive to system configuration issues, many of which are covered by best practices. Hundreds of configuration items should be assessed based upon the type and intended usage of the database (e.g. a database supporting an external Web application may require different configuration than one supporting a large group of internal users). One simple example is password change frequency. Best practices recommend that user account passwords change frequently. The database assessment should allow the administrator to configure a change frequency (i.e. 30 days) policy and then automatically compare that policy to the age of all actual passwords. The assessment solution may then list all accounts with passwords that are in violation of policy to help prioritize the threat and remediate the problem. A good assessment test checks whether the administrator has used built-in database mechanisms to correctly enforce the password policy.

Privilege Management

The extent to which database privileges are managed should also be assessed. In general, database owners should adhere to a "least privilege" policy. For example, best practices mandate that user privileges should be granted to accounts only through roles. Privileges granted directly may yield excessive access rights when users shift positions within the organization. Direct grants also imply an undocumented authorization process – a common barrier to compliance with regulations such as Sarbanes-Oxley. Therefore, the

assessment should examine the data dictionary for records in which the grantee is an account and not a role. Any accounts that match this condition may then be listed along with the privileges directly granted to help to measure the threat and help remediate the problem.

External Objects

Certain objects that are external to the database may be leveraged (if not properly configured) to attack a database. These external objects are primarily operating system objects such as files, services, registry keys, etc. For example, DB2 includes an executable file called db2job that can be exploited, by default, to allow local users to execute code with administrative privileges. Susceptibility to this vulnerability can be determined by checking operating system permissions on the db2job file.

Regulatory Compliance

Database assessment should pay special attention to security requirements that are specifically relevant to regulatory compliance. For example, SOX requires tracking of all new user accounts within databases that store financial reporting information. Therefore, the assessment should check the data dictionary for accounts whose creation date is within a configurable time period. Any new accounts may then be listed in the assessment results reports.

Barriers to Assessment

Despite the benefits to database assessment, many organizations face one of several implementation barriers: budget, resources, and staff expertise.

Budget

Many organizations require written justification prior to approving budget for investments in IT. Database security assessment measures the security posture of their databases and can help justify investment in additional database security measures, but what can IT use to justify investment in assessment tools?

Unfortunately, commercial assessment tools require investment before IT can completely quantify the need. Free security assessment tools have been available for the network, Web servers, and other infrastructure segments, but until recently there were no free database assessment tools.

Resources

Database administrators are usually overworked and their responsibilities are limited to daily operations. They have limited bandwidth to allocate toward security tasks. Since the consequences of lax security, although inevitable, may not be immediate, database assessment tends to be pushed to the back burner.

Expertise

Organizations with staff dedicated to security and resources to assess database infrastructure rarely have the level of database-specific expertise necessary to conduct a best practices assessment that includes analyzing database configurations, privileges, etc.

Overcoming the Barriers to Database Security Assessment – Scuba by Imperva

Scuba by Imperva is a free database vulnerability scanner that can be used to conduct an initial database assessment. Scuba by Imperva is a lightweight Java program available as a Windows executable for download at www.imperva.com/scuba. It contains over 350 assessment tests for Oracle, Microsoft SQL Server, IBM

DB2, and Sybase databases. This tool is used by organizations to overcome the barriers to database assessment described above.

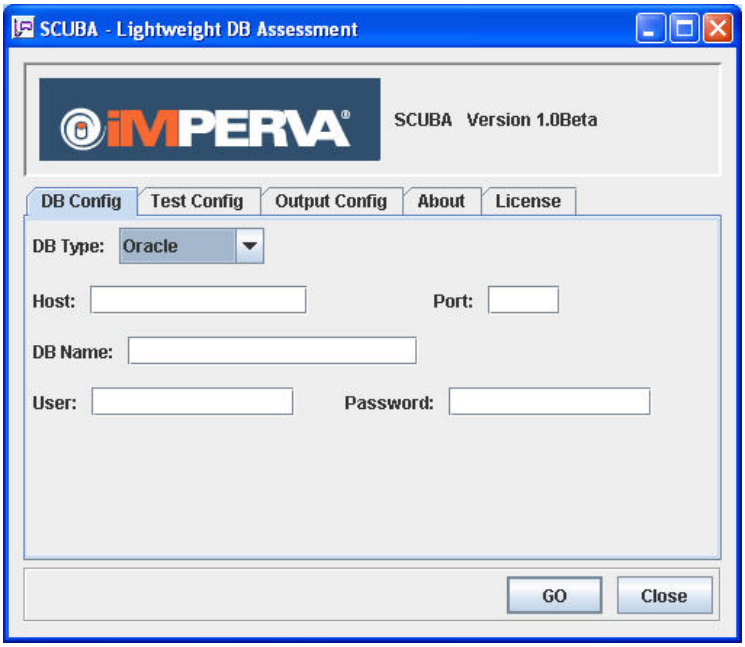

Figure 2: SCUBA requires minimal configuration. Within minutes, reports are generated to meet security, compliance, and executive needs.

- **Budget** – As a free scanner, Scuba by Imperva enables organizations to evaluate database security posture without extensive budget allocation. The reports Scuba generates may be used to evaluate and/or justify budget for additional database security.
- **Resource** – Scuba by Imperva automates over 350 assessment tests. Vulnerability identification, prioritization, and reporting require no administrator intervention.
- **Expertise** – Scuba by Imperva requires a minimum configuration process that includes only a handful of database parameters, including type, admin account name, password, host name/IP, database name, and port. Default assessment policy parameters (i.e. password change frequency) provide reliable results although policy can be easily modified as necessary. Graphical reports in HTML format apply universal terminology and straightforward classifications that make it easy to evaluate results.

Scuba by Imperva also meets the four imperative criteria for best practice assessment.

- **Impact on Production Systems** – Assessment tests check for conditions that prove the existence of vulnerability but do not exploit any vulnerability. Therefore, Scuba can be applied to production databases without risk of downtime or damage.
- **Accuracy** – Test procedures incorporate the best practice knowledge of the Imperva Application Defense Center (ADC), an internationally recognized research organization dedicated to the advancement of database security. Test procedures go beyond basic software version checking and other coarse assessment mechanisms. For each test, enough detailed information is collected to definitively validate the existence or non-existence of vulnerability. Tests are regularly updated with the latest ADC research. It is important to use the latest version of the Scuba by Imperva database assessment tool.
- **Efficiency** – Reports are analyzed and prioritized to enable efficient remediation. There is no need to sift through pages of technical documentation to identify the most serious vulnerabilities . In addition, executive-level reports eliminate the burden of translating results into a format that may be interpreted by management.
- **Breadth of Testing** – Testing covers each key assessment area described above: known vulnerabilities, system configuration, privilege management, external objects, and regulatory compliance.

Beyond the First Step - Enterprise Assessment and Security

The Scuba by Imperva scanner enables organizations to take the first step toward improving database security. However, sustainable database security across an enterprise requires that organizations address more advanced requirements that include scalability, behavioral assessment, and implementation of a complete database security life cycle program.

Scalability

Lightweight database assessment as provided by Scuba by Imperva enables immediate assessment of an individual database. The simplicity of this model makes it extremely useful for organizations that are just starting a database assessment program. However, as the assessment process matures across large database environments, more sophisticated management features and automation are required.

- **Multiple Servers** – Large-scale database assessment requires the ability to automatically assess multiple servers or server groups.
- **Multiple Poli**cies – Multi-server assessment often entails the need to assess unique policies for each server or server group.
- **Scheduling** – To remain current, the assessment process should be scheduled and repeated periodically across all servers.

Continuous Assessment

To this point our discussion of assessment has been limited to "point-in-time" assessments that evaluate security posture by extracting security information that is available at a given point in time. However, there are aspects of best practice security assessment that require continuous user behavior assessment. The identification of shared accounts provides a good example of this requirement.

An Example of Continuous Assessment - Shared Accounts

The sharing of a single database account by many users violates the fundamental security requirement for user accountability. There is no way to definitively link any single user to a potentially malicious event whenever user accounts are shared. Therefore, virtually every best practice IT security controls framework (COBIT, etc.) emphasizes the need to prevent and/or detect account sharing. As a result, this issue is another common barrier to compliance with regulations such as Sarbanes-Oxley. Unfortunately, a point-in-time assessment of database account information does not reveal shared accounts. Shared account discovery requires behavior monitoring to continuously track database sessions and identify any accounts that are used by concurrent sessions from different IP addresses.

The Database Security Life Cycle

Database assessment represents the first step within a complete database security life cycle. Three subsequent steps build upon the results of the assessment.

- **Set Policies / Controls** – With a complete assessment of database risk and user profiles available as guidance, the administrator is in a position to define audit and security policies or "controls". These policies specify allowed behavior, disallowed behavior, and specifically regulated transactions.
- **Monitor and Enforce** – Database activity is monitored and enforced according to policy. Audit data is collected and violations optionally trigger real-time response.
- **Measure** – Reporting systems measure the results of the previous three phases of the cycle. Reports serve as a feedback loop for adjusting the lifecycle and help communicate results to auditors and management.

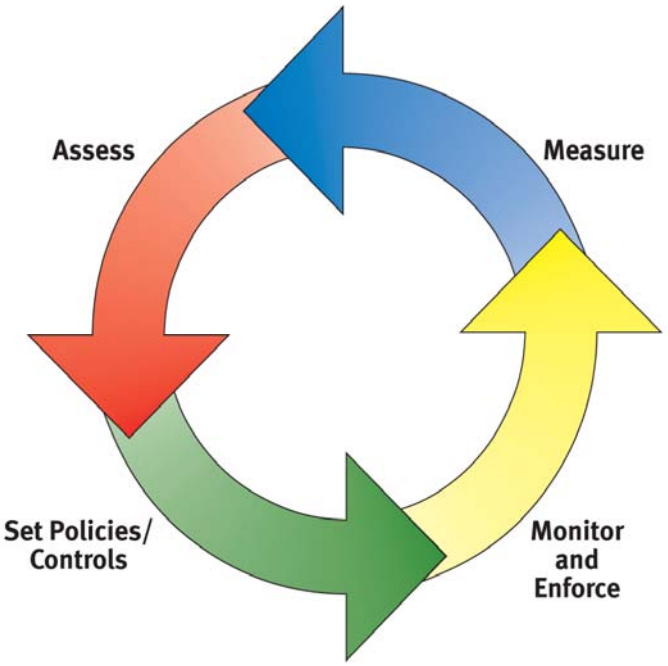

Figure 3: The Database Security Lifecycle

Imperva SecureSphere – A Complete Database Security Lifecycle Solution

The Imperva SecureSphere database security and monitoring appliances (Figure 4) enable organizations to implement all four phases of a complete database security life cycle: assess, set policies / controls, monitor and enforce, and measure.

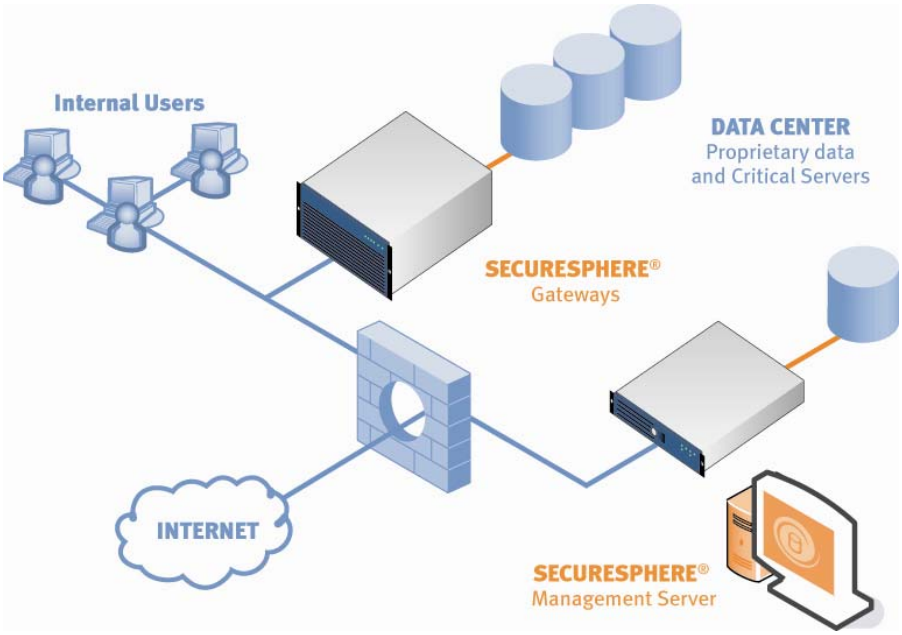

Figure 4: SecureSphere consists of three main components - SecureSphere Gateways, the SecureSphere Management Server and the DB Monitor Agents. Together, they enable a complete database security lifecycle.

SecureSphere Assessment

SecureSphere incorporates all of the capabilities in Scuba by Imperva outlined above while adding a series of advanced management features that scale assessment processes across the largest enterprise data centers.

- **Multiple Servers** – Multiple servers or server groups are automatically assessed without administrator intervention.
- **Multiple Policies** – Multiple policies are stored and applied to different servers or server groups.
- **Hierarchical and Role-based Management** – IT Assets and Users/Roles can be arbitrarily defined, with hierarchies if needed, to aid management in a delegated fashion. This vastly reduces the operational overhead of an activity monitoring, audit and security solution.
- **Task Oriented Workflow** – SecureSphere implements a workflow engine that can be used to manage audit and security processes as well as verify that appropriate steps are followed in case of monitoring or security alerts.
- **Scheduling** – Assessment processes are automatically run according to a predefined schedule.

SecureSphere significantly exceeds the point-in-time capabilities of Scuba by Imperva by incorporating automatic data discovery, and continuous user behavior assessment. For example, SecureSphere includes the following capabilities:

1. **Server and Sensitive Data Discovery** – SecureSphere can scan an IP range to identify application, web and database servers, and scan these data stores for sensitive information stored in these servers.

2. **Dynamic Profiling** – By observing user activity over time, SecureSphere automatically builds and maintains a complete model of each user's normal behavior. The profiles enable administrators to assess the location of sensitive data, know who is accessing it, and how they are using it. These profiles also serve as a baseline for establishing policies that govern acceptable database usage. [See Set Policies / Controls below.]
3. **Best Practice Behavior Assessment** – By analyzing user profiles and monitoring traffic for specific patterns, SecureSphere reveals best practice and regulatory behavioral violations such as shared accounts, dormant accounts, etc. These behavior oriented violations cannot be identified by point-in-time assessment alone.

Set Policies / Controls

Imperva SecureSphere policies enable administrators to define rules that specify acceptable behavior, unacceptable behavior, and specifically regulated transactions. SecureSphere policies are highly granular with attributes that extend to specific database columns, SQL operations (SELECT, etc.), query response capture, time-of-day restrictions, source IP, source hostnames, source OS, and more.

Monitor and Enforce

SecureSphere may be flexibly configured to monitor and enforce policies. Audit data is highly granular enabling comprehensive forensics. Audit data storage leverages a distributed architecture that scales across the largest data centers while maintaining a unified view. Audit archival supports automated scheduling, compression, encryption, and digital signing. Security violations may trigger response actions ranging from simple event logs, to real-time alerts, to user blocking. Finally, advanced event correlation enables accurate detection of the most sophisticated attacks.

Measure

The graphical reporting capabilities of SecureSphere enable measurement of the results of the previous three steps of the database security lifecycle. Preconfigured regulatory compliance and security templates are available for immediate use. Custom reports are also easily created.

SecureSphere Deployment and Operational Efficiency

SecureSphere can be deployed either inline or non-inline with zero impact upon surrounding data center infrastructure. It requires no changes to databases, applications or the network. In addition, automated learning mechanisms eliminate the need for extensive configuration or ongoing tuning. As a result, SecureSphere deployments can be accomplished in less than one day.

Summary

Database security assessment is an important first step in a good overall database security strategy. The four best practice design criteria must include impact on product systems, accuracy, efficiency, and breadth of testing. The free Scuba by Imperva database vulnerability assessment tool addresses each of these criteria while overcoming budget, resource, and expertise barriers. Beyond an exploratory first step offered by Scuba and toward a more secure database infrastructure, an organization should consider deployment of a complete database security lifecycle solution, as offered by the Imperva SecureSphere database security and monitoring gateway products. Imperva SecureSphere products provide a complete database security lifecycle that not only extends assessment across the enterprise, but also integrates granular policy definition, monitoring, enforcement, and measurement capabilities.

For more information on Scuba by Imperva and to download the free database assessment tool, please visit http://www.imperva.com/application_defense_center/scuba/.

For more information on how Imperva SecureSphere products meet your database security and compliance requirements, please visit http://www.imperva.com/solutions/ or contact Imperva Sales at sales@imperva.com.

US Headquarters
950 Tower Lane
Suite 1550
Foster City, CA 94404
Tel: +1-650-345-9000
Fax: +1-650-345-9004

International Headquarters
125 Menachem Begin Street
Tel-Aviv 67010
Israel
Tel: + 972-3-6840100
Fax: + 972-3-6840200

© 2007 Imperva, Inc. All rights reserved. Imperva and SecureSphere are registered trademarks of Imperva, Inc.
All other brand or product names are trademarks or registered trademarks of their respective holders. WP_DBSA0807.01