Ironsphere SQL Proxy

Stronger, Simpler and Increased Security Accessing Database Networks

Organizations rely heavily on data security. Regardless of the industry (Financial, Insurance, Telecom, Pharma, etc.) or data type (product, employee, customer, financial, medical), organizations recognize the need to secure and effectively monitor their data.

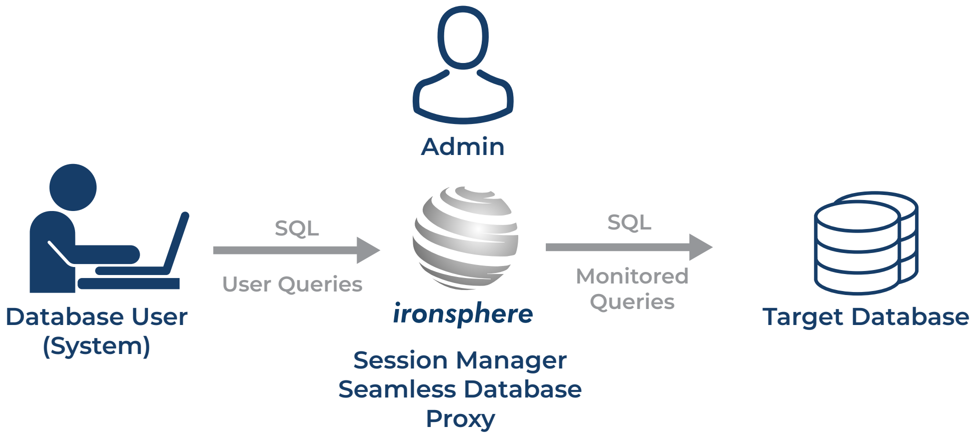
In this solution brief, you will learn about the SQL Proxy capabilities provided by Ironsphere, including the following:

- Ironsphere comprehensively logs all privileged session database connections and activities that may generate data breaches and impact business continuity.
- Ironsphere efficiently and centrally secures and controls privileged access to databases.

Full Visibility:	Logging entire session (SQL) messages indisputably.
Single Sign On:	Integration with Directory (LDAP/AD) services. Auto lock inactive accounts. Auto lock user account on employee termination/leave.
Least Privilege:	User Groups and Database Groups can be defined separately. Time & Date based Access Management. Multi-tenancy.

Logging

Ironsphere provides session logging functionality to database admins. All database queries can be logged for security and compliance purposes. The user experience is unchanged as they continue to use their own database client. The client can communicate with Ironsphere's SQL Proxy through the standard protocol used by the database.



Database admins can control logging as well as user permissions to execute SQL commands, by employing policy enforcement and database masking as preventive actions.

Policy Enforcement

Database admins can control user permissions to run SQL queries by establishing rules or policies that comply with the established security requirements. Ironsphere's policy enforcement can be accomplished in four ways:

1) Blacklist and whitelist
2) Time-based restrictions
3) Maintenance mode restrictions
4) Context based restrictions

Dynamic Data Masking

Data masking technology is aimed at preventing the abuse of sensitive/confidential data by giving users fictitious (yet realistic) or hidden data, instead of real and sensitive data.

Data masking targets the misuse of data at rest, typically in nonproduction databases (static data masking), and data in transition, typically in production databases (dynamic data masking).

Dynamic Data Masking is necessary, especially for application testing use cases that require representative and coherent data. Dynamic data masking (DDM) limits sensitive data exposure by masking it to non-privileged users.

ORIGINAL DATA

Name	Phone	Birth Date
John Doe	511-336-4455	11.4.1986
Adam Smith	511-472-1314	2.2.1967

MASKED DATA

Name	Phone	Birth Date
John Doe	511-111-1111	1.2.1987
Adam Smith	511-123-4567	10.11.1966

DDM can also be configured to hide sensitive data in the database query result sets over designated database fields, while the data in the database remains unchanged.

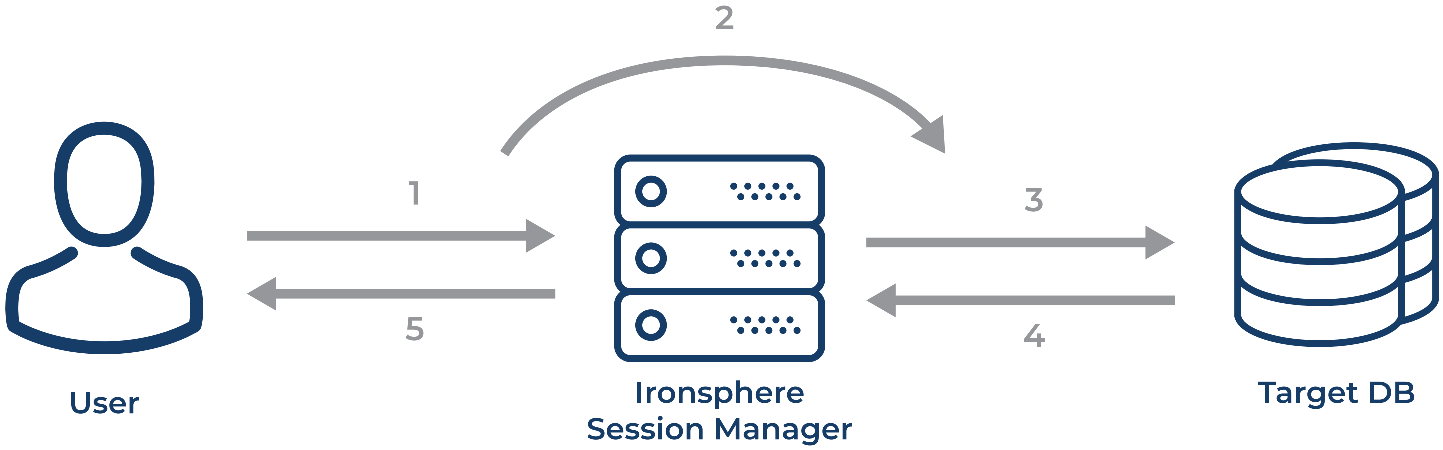
ORIGINAL DATA

Name	Phone	Birth Date	Credit Card
John Doe	511-336-4455	11.4.1986	1111 2222 3333 4444
Adam Smith	511-472-1314	2.2.1967	5555 6666 7777 8888

MASKED DATA

Name	Phone	Birth Date	Credit Card
John Doe	511-111-1111	1.2.1987	1111 2222 3333 XXXX
Adam Smith	511-123-4567	10.11.1966	5555 6666 7777 XXXX

How it Works?



Step 1: User runs query.

Step 2: Query is logged and then re-written based on the masking rule.

Step 3: Manipulated query is forwarded to target DB.

Step 4: Target DB returns the result of query to Ironsphere.

Step 5: Ironsphere forwards the results to the user.

Masking Rules

Redaction/Nulling

Masking original value with a fixed and blacked-out value. E.g., all phone numbers are nulled: 511-567-1918 -> xxx-xxx-xxxx.

Shuffling

Shuffling the original values. E.g., postal code 34670 is returned as 43706.

Blurring

Adding or subtracting a random value (within limits) to/from the original value. E.g., original birth date 11.04.2001(MM.DD.YYYY) is changed to 07.29.2001.

Tokenization

Masking a section of the original value. E.g., original credit card number 1111-2222-3333-4444 is changed to 1111-1234-5678-4444.

Substitution

Substituting the original value with another random value, selected from a pre-defined set. E.g., original name John is changed to Adam which is randomly selected from a pre-defined name list.

Advanced

Custom rules defined by regular expressions.

Benefits

- » All queries are logged indisputably. Users authenticate with their own credentials even if there is no such DB user, so the real user running a query is known and logged.
- » Sensitive data can be manipulated and delivered to applications or users in such a manner that it is no longer sensitive, but still coherent and usable.
- » Policies (DB masking rules) can easily and instantly be assigned to users, application accounts and/or groups and roles.
- » Eliminates weak and non-expiry passwords. Disables inactive accounts.
- » Accounts can be time-limited (hours of day, day of week, etc.).
- » Has no performance degradation impact on target Databases.
- » Users are not required to use a proprietary database client and can continue using familiar tools (i.e. Toad, etc.).

For more information about how your company can benefit from an advanced, automated privileged account management system, visit www.ironsphere.com.