Identify and Protect your Sensitive Data with Seamless Interception

Jack Di Giacomo
TANDsoft, Inc.
Jack loves the outdoors

30 years of experience in the design, development and support of NonStop software solutions.

Former Tandem instructor and Specialist in intercept technology.
Let’s define Interception Technology as it applies to computer programming

Interception technology covers a range of techniques that can be used to alter or augment the behaviors of applications, operating systems, or other software components by intercepting function calls or system calls.
The code that handles intercepted function calls, system calls, events, or messages is called a “hook” or an “intercept” library.
Architecture of Interception

program

Intercept Library

Guardian  OSS  Enscribe  SQL/MP  SQL/MX  IPC  TMF  TCP/IP  DLL
Safeguard  Tape  Pathway  System Clock
The best thing about Seamless Interception...

No Source Code Required!

No Need to Recompile Programs!

Works with all Apps!
Interception Technology is used to monitor behaviors and to modify application functions.
NonStop customers use interception to extend application functionality

- Business Continuity

- Automatic TMF protection of Enscribe files
  - Insert TMF transactions BEGINTRANSACTION, ENDTRANSACTION

- Replicate Enscribe unaudited files or Enscribe file modifications to a backup site

- Replicate Enscribe, SQL/MP, and SQL/MX DDL changes to a backup site
NonStop customers use interception to extend application functionality

Virtualization and System Consolidation

- **Time-Zone simulation**
  - Allows Guardian and OSS applications to operate within any virtual time zone

- **System Clock simulation**
  - Allows Guardian and OSS applications to operate with any virtual system clock or current time value
Large North American bank used interception to consolidate applications requiring multiple time zones into one data center.

Virtualization and System Consolidation

The bank decided to consolidate its West-Coast operation into its East-Coast data center as a cost-savings measure. A challenge was determining how to run applications that needed to run in the Pacific Time Zone on a system with a clock set to the Eastern Time Zone.

Using interception technology, the bank created virtual time zones that allow one production system to service two time zones and one disaster recovery system to service two time zones.
NonStop customers use Seamless Interception to Identify and Protect Sensitive Data

**Security and Compliance**

- Identifies and logs all sensitive database access
- Protects sensitive data at rest via
  - encryption or tokenization
  - data masking
- Supports Enscribe, SQL/MP, SQL/MX, TNS/R,E,X
- All without the need for any application modifications
- Helps organizations comply with government and industry regulations (PCI, GDPR)
Seamless Intercept Technology Example for Security and Compliance

Security

Intercepts HPE NonStop database access calls, then works with HPE, third-party, or in-house security solutions to protect sensitive data by encrypting / tokenizing data written to disk and decrypting / de-tokenizing data read from disk.

Intercepts and Masks sensitive data.

Compliance

Intercepts and Logs sensitive database access and statements.
Identify and protect Enscribe, SQL/MP and SQL/MX Sensitive Data

We all agree, it is critical to protect sensitive data

**Protect - NonStop sensitive data**

Using Seamless interception technology +
- comForte SecureDPS
- Micro Focus (Voltage, Xypro) SecureData
- Protegrity and others
- Data masking

**Identify - NonStop sensitive data**

Using Seamless interception technology to
- Log all access to NonStop DB
Protect (with no source code changes) two columns in EMPLOYEE SQL/MP table

mxci (version with intercept library, IL) will be used to access the EMPLOYEE SQL/MP table.

Customer can use any Encryption or Tokenization engine.
- NonStop Partner solutions
- Micro Focus/HPE/Voltage SecureData
- Protegrity or other

For this example, a simple encryption technique was used;
- Character substitution: 0 = 9, 1 = 8, 2 = 7, ... $ = %
Seamless Data Protection - Example

SQL/MP Table EMPLOYEE

( EMP_ID SMALLINT NO DEFAULT NOT NULL,
  NAME CHAR(10) NO DEFAULT NOT NULL,
  SSN CHAR(11) DEFAULT NULL,
  HIRED_DAY DATETIME YEAR TO DAY DEFAULT NULL,
  END_DAY DATETIME YEAR TO DAY DEFAULT NULL,
  SALARY CHAR(10) NO DEFAULT NOT NULL )

Protected Table EMPLOYEE, Columns SSN, SALARY

Data in the Clear (Green)
Data is Encrypted / Tokenized (Red)
Data is Masked (Purple) SSN mask = (xxx-xx-x???)
Seamless Intercept Library (Transparent, NO App changes) (Dark Red)
Seamless Data Protection - Example

mxci (with IL) will be used to access the SQL/MP table.

Intercept Library name = sdSQLdll
Original mxci = /usr/tandem/sqlmx/bin/mxci
mxci program = /demo/mxci
eld -change libname ‘$sas21.sdi2.sdSQLdll’ /demo/mxci

run mxci (with IL):
osh -c "run /demo/mxci"
Seamless Data Protection - Example

mxci - INSERT into =employee values ( 2, ‘John D’, ‘222-12-3456’,
datetime '2017-08-01' YEAR TO DAY, NULL, ‘$60,000’ );

PROCESS_MX_TABLE ( Table=$SAS21.SDI2.EMPLOYEE )
REGISTER_MX (SSN, SALARY )
Registered Columns (Table=$SAS21.SDI2.EMPLOYEE: SSN,SALARY)
Process_Col.in. {SSN, InsUpd, (222-12-3456)} {SALARY, InsUpd, ($60,000)}
Process_Col.out. {SSN, InsUpd, (777-87-6543)} {SALARY, InsUpd, (%39,999)}

--- 1 row(s) inserted.
# Seamless Data Protection - Example

**Original SQLCI** - SELECT * from `employee`;

<table>
<thead>
<tr>
<th>EMP_ID</th>
<th>NAME</th>
<th>SSN</th>
<th>HIRED_DAY</th>
<th>END_DAY</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>John D</td>
<td>777-87-6543</td>
<td>2017-08-01</td>
<td>?</td>
<td>%39,999</td>
</tr>
</tbody>
</table>

**mxci** - SELECT * from `employee`;

- **Process_Col.in.**
  - `{SSN, Read}, (777-87-6543)`, `{SALARY, Read}, (%39,999)`

- **Process_Col.out.**
  - `{SSN, Read}, (xxx-xx-x456)`, `{SALARY, Read}, ($60,000)`

<table>
<thead>
<tr>
<th>EMP_ID</th>
<th>NAME</th>
<th>SSN</th>
<th>HIRED_DAY</th>
<th>END_DAY</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>John D</td>
<td>xxx-xx-x456</td>
<td>2017-08-01</td>
<td>?</td>
<td>$60,000</td>
</tr>
</tbody>
</table>
Seamless Data Protection - Example

mxci - UPDATE =employee set salary = '$65,000' where emp_id = 2;

Process_Col.in. {SALARY, InsUpd ($65,000)}
Process_Col.out. {SALARY, InsUpd (%34,999)}
--- 1 row(s) updated.

Original SQLCI - SELECT * from =employee;

<table>
<thead>
<tr>
<th>EMP_ID</th>
<th>NAME</th>
<th>SSN</th>
<th>HIRED_DAY</th>
<th>END_DAY</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>John D</td>
<td>777-87-6543</td>
<td>2017-08-01</td>
<td>?</td>
<td>%34,999</td>
</tr>
</tbody>
</table>
--- 1 row(s) selected.

mxci – DELETE from =employee where emp_id = 2;
--- 1 row(s) deleted.
Seamless Intercept Technology API - Example

mxci - INSERT into =employee values ( 2, ‘John D’, ‘222-12-3456’,
datetime '2017-08-01' YEAR TO DAY, NULL, ‘$60,000’ );

**Seamless Intercept API** + **Protection Code**

**Seamless Intercept API**
calls PROCESS_MX_TABLE (=employee)
calls PROCESS_COL

**Protection Code**
For a protected table (=employee),
calls REGISTER_MX (SSN,SALARY)

input = {SSN, (222-12-3456)},
{SALARY, ($60,000)}
output = {SSN, (777-87-6543)},
{SALARY, (%39,999)}

mxci - SELECT * from =employee;
calls PROCESS_COL

input = {SSN), (777-87-6543)}
{SALARY), (%39,999)}
output = {SSN), (xxx-xx-x456)}
{SALARY), ($60,000)}
Identify and Log NonStop DB Access - Example

Log NonStop DB access (Insert, Update, …) to EMPLOYEE table (columns SSN and SALARY)
- Program name, Process name, Login name, File/Table name, Column Name, DB Access, Time Stamp

Log DB SQL statements from mxci program
- Program name, Process name, Login name, Time Stamp, SQL Source Statement

** Data from Seamless Data Protection example used for following slides **
Identify DB Access – Example
( NonStop DB Access Log )

```sql
SELECT * from =SDIAUDE;
```

<table>
<thead>
<tr>
<th>Program</th>
<th>Process</th>
<th>Login Name</th>
<th>Table</th>
<th>Column</th>
<th>Access</th>
<th>Time Stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>=EMPLOYEE</td>
<td>SSN</td>
<td>Insert</td>
<td>2017-10-07 19:11:27.088123</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>=EMPLOYEE</td>
<td>SALARY</td>
<td>Insert</td>
<td>2017-10-07 19:11:27.097439</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>=EMPLOYEE</td>
<td>SSN</td>
<td>Read</td>
<td>2017-10-07 19:11:27.482625</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>=EMPLOYEE</td>
<td>SALARY</td>
<td>Read</td>
<td>2017-10-07 19:11:27.491312</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>=EMPLOYEE</td>
<td>SALARY</td>
<td>Update</td>
<td>2017-10-07 19:11:27.582885</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>=EMPLOYEE</td>
<td>SSN</td>
<td>Read</td>
<td>2017-10-07 19:11:27.770105</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>=EMPLOYEE</td>
<td>SALARY</td>
<td>Read</td>
<td>2017-10-07 19:11:27.779283</td>
</tr>
</tbody>
</table>
### Identify DB Access - Example (NonStop DB Statement Log)

```sql
SELECT * from SQLAUDE;
```

<table>
<thead>
<tr>
<th>Program</th>
<th>Process</th>
<th>Login Name</th>
<th>Time Stamp</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>2017-10-07</td>
<td>19:11:27.085</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>2017-10-07</td>
<td>19:11:27.086</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>2017-10-07</td>
<td>19:11:27.381</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>2017-10-07</td>
<td>19:11:27.482</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>2017-10-07</td>
<td>19:11:27.685</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>2017-10-07</td>
<td>19:11:27.770</td>
</tr>
<tr>
<td>/mxci</td>
<td>$Z123</td>
<td>TAND.JACK</td>
<td>2017-10-07</td>
<td>19:11:27.998</td>
</tr>
</tbody>
</table>
Identify - NonStop DB Access - Usage

Industry compliance (PCI, GDPR)
Auditing - Log all sensitive data access.
List all access to the Primary Account Number (PAN)
• select * from =SDIAUDE where Column = ‘PAN’;

Detect Fraudulent or unauthorized access
List all DB access to PAN column between midnight & 6 am
• select * from =SDIAUDE where Column = ‘PAN’ and cast (timestamp as datetime HOUR) between datetime ‘00' HOUR and datetime ‘06' HOUR;
Quality Control and Testing

Compare SQL source statement repository for each product release.

List all SQL statements from program remote/banking server
• Select * from =SQLAUDE where program like ‘%remote/banking%’;

List all SQL statements from mxosrvr (JDBC/ODBC) server.
• Select * from =SQLAUDE where program like ‘%mxosrvr%’;
The best thing about Seamless Interception for Sensitive Data ...

No Source Code Required!
No Need to Recompile Programs!
Works with all Apps!
In Production and available today!
Thank You for Attending

Any Questions?

Ask them now, or contact me later at

jack.digiacomo@tandsoft.com

www.tandsoft.com