Getting Started with SAP Application Server ABAP 7.4 SP5 with SAP NetWeaver AS, add-on for code vulnerability analysis [Trial Edition]

Provided as Virtual Appliance by the SAP Cloud Appliance Library

Version 1.0

August 2014

Typographic Conventions

Type Style	Description
Example Text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
`Example text`	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, F2 or ENTER.

Icons

Icon	Description
⚠	Caution
❗	Important
✎	Note
⬆	Recommendation or Tip
➔	Example

Table of Contents

1. Overview

This guide provides information on first steps for using instances created from the “SAP NetWeaver Application Server ABAP 7.4 SP05 with SAP NetWeaver AS, add-on for code vulnerability analysis [Trial Edition]” solution.

2. Solution Information

This solution comprises a backend image and a frontend image which can be instantiated using the *Create Instance* command:

You can find the installation media on which this solution is based on SAP Service Marketplace in the Software Download Center: https://service.sap.com/swdc

Backend [SUSE Linux Enterprise Server 11, patch level 2]

[Main Component]: SAP High Performance Analytic Appliance 1.0

[Stack-no. of Main Component]: 1.00.74

[Main Component]: SAP NetWeaver Application Server ABAP 7.4

[Stack-no. of Main Component]: 7.40 SP05

Frontend [Microsoft Windows Server 2008 R2, AWS AMI]

[Main Component]: SAP JVM 7

[Stack-no. of Main Component]: 7.1.013

[Main Component]: SAP GUI for Windows 7.30

[Stack-no. of Main Component]: 7.30 patch level 07

[Main Component]: SAP HANA Studio 1.0

[Stack-no. of Main Component]: 1.00.70

[including Component]: ABAP Development Tools 2.19.2

[including Component]: SAPUI5 Tools 1.16.6

[Main Component]: SAP HANA Client 1.0

[Stack-no. of Main Component]: 1.00.70

3. Using the SAP Cloud Appliance Library

3.1. Retrieving Your AWS Account Information

You will need the following information of your Amazon Web Services (AWS) account for configuring the cloud service provider account of SAP Cloud Appliance Library. This information is required to establish the communication between SAP Cloud Appliance Library and your cloud computing environment:

1. The *Access Key* and the *Secret Key* of your AWS account.
   To view your AWS access key and secret key, use the following procedure:
   1. Navigate to http://aws.amazon.com.
   2. Logon to your account.
   3. Choose *Account* → *Security Credentials*.
   4. In the *Access Credentials* section:
      - To see your access key, choose the *Access Keys* tab.
      - To see your secret key, choose the *Secret Access Key* tab and then choose the *Show* link.

2. Ensure to use the AWS location **US-East (Virginia)** for creating instances or preparing a virtual private cloud (VPC) for your instances.

3.2. Using SAP Cloud Appliance Library

Enter your CAL account using the link to the test drive center of the SAP Cloud Appliance Library:

`https://caltdc.netweaver.ondemand.com/console/tenant_`**`<tenant_name>`**

The next steps show how to configure your solution in SAP Cloud Appliance Library:

1. Create an account in SAP Cloud Appliance Library using your AWS credentials described above. As the user who has created the account, you become an account owner and can assign other users to your account.
2. Browse for your solution (“SAP NetWeaver Application Server, add-on for code vulnerability analysis [Trial Edition]”) in the *Solutions* tab page and activate it.
3. Select the activated solution and hit the *Create instance* button to start the wizard.
4. In the wizard you can choose between two important deployment options:
    a) Public: If you choose this option, we strongly recommend to **uncheck** (not check) the *Open all TCP ports* option in one of the following steps. This creates a default security group for your solution instance acting like a firewall. Thus, only port 22 (SSH) and port 3389 (RDP) are accessible from outside.
    b) Corporate Network: This is the right option for creating your instances in the subnet of a virtual private cloud (VPC). In a secure VPC environment you could also open additional ports by adding additional *Access Points* to the default entries.

    Internally, meaning for the communication between frontend and backend instances, all ports are open (valid for both options).

! Please be aware that creating your instances in the public zone of your cloud computing platform is convenient but less secure. Thus, please ensure to open only port 22 (SSH) and port 3389 (RDP) of the default security group. In addition, we also recommend to limit the access to your instances by defining a specific IP range in the *Access Points* settings using CIDR notation. The more complex but secure alternative is to set up a virtual private cloud (VPC) with VPN access, which is described in this tutorial on SCN.

The list below describes the open ports of the default security group, if you don’t check the *Open all TCP ports* option (recommended setting):

Protocol	Port	Description
SSH	22	Used for SSH connection to the backend server
RDP	3389	Used for RDP connection to the frontend server

✎ For more information about these three steps, see the official documentation of SAP Cloud Appliance Library (choose Related Links & Help → Documentation and choose + (expand all) button to see all documents in the structure). You can also use the context help in SAP Cloud Appliance Library by choosing the Help panel from the right side.

The creation of the solution instance including starting the database and the ABAP system takes initially about 45 minutes. **Please be patient and don’t interrupt the initial deployment phase.**

3.3. Working with Solution Instances

You can find the solution instances you created on the *Instances* tab page of the SAP Cloud Appliance Library. For more information, see the *Working with Solution Instances* document from the official documentation of SAP Cloud Appliance Library (choose *Related Links & Help* → *Documentation* and choose ⊞ (expand all) button to see all documents in the structure). You can also use the context help in SAP Cloud Appliance Library by choosing the *Help* panel from the right side.

✎ If you decided to go with the option "Activate or suspend manually" during instance creation please make sure to suspend the instance manually when you don't work with the instance to avoid unnecessary costs. We also strongly recommend to set up a billing alert for your AWS charges.

4. Connecting to Your Frontend Instance

4.1. Remote Desktop Client

For connecting to your frontend instance you need an RDP client for your local operating system:

Microsoft Windows: Start the *Remote Desktop Connection* using the Start Menu (All Programs > Accessories) or executing mstsc.exe.

Apple Mac OS X: Use the free Microsoft Remote Desktop app available in the Mac App Store to connect to your frontend.

Linux: You can use open source RDP clients like rdesktop or FreeRDP.

4.2. Connecting with RDP

You can find the `<IP Address>` of your frontend instance by clicking on the instance name in your SAP Cloud Appliance Library account. Take the IP of the frontend instance and use the OS user "Administrator" with your master password to log in with your remote desktop client.

4.3. Using SAP GUI for Windows

SAP GUI for Windows is already installed on your frontend instance with a pre-configured system connection for SAP Logon. If you want to create an additional entry, proceed with the following steps:

1. Start the SAP Logon.
2. Choose new entry → User defined.
3. In the *System* wizard, specify the following parameters:

Parameter ID	Parameter Value	Note
Application Server	*`abapci`*	The IP address of the instance from SAP Cloud Appliance Library
Instance Number	00	ABAP instance number used by the appliance.
System-ID	A4H	ABAP system id used by the appliance.
User Name	Client 000: SAP*, DDIC Client 001: SAP*, DDIC, DEVELOPER	Default users
Password	*`<Master Password>`* ❗ It is recommended that you change the password for all three users directly after creation of the instance!	The password of *SAP, DDIC and DEVELOPER* users are the same.

✎ For out-of-the-box ABAP development and the pre-configured demo applications we recommend to use the user DEVELOPER in client 001.

4.4. Using SAP HANA Studio

The pre-installed SAP HANA Studio on your frontend instance also contains the ABAP Development Tools, the SAPUI5 Tools, and the BW Modeling Tools. In the *Systems* view of the *HANA Development* perspective you find a pre-configured connection to your HANA system, using the following parameters:

Parameter ID	Parameter Value	Note
Hostname	*hanadb*	The IP address of the instance from the SAP Cloud Appliance Library
Instance Number	02	HANA instance number used for the appliance.
User Name	SYSTEM	For the connection to the DB use *SYSTEM* user.
Password	*<Master Password>*	The password is the same as the master password provided during instance creation in the SAP Cloud Appliance Library.

The system ID of the database is HDB. It is recognized automatically via the host name.

For detailed information about these SAP development tools for Eclipse, we recommend to use the documentation available within Eclipse by opening the *Help* menu > *Help Contents* or consult the standard documentation available at http://help.sap.com/.

5. Connecting to Your Backend Instance

5.1. Connecting to Your Backend on OS Level

In case you want to access your backend instance on OS level (not recommended unless you know what you are doing), you need an SSH client for your local environment, e.g. PuTTY for Windows.

The following steps describe how to connect to your backend instance using PuTTY:

1. Click on the instance name in your CAL account, to retrieve the IP of your backend instance and download the instance key pair (maybe you already downloaded the key pair during instance creation).
2. Extract the private key of the key pair by using a tool like puttygen.exe.
3. Open PuTTY and enter the IP of your backend instance.
4. Navigate to the SSH > Auth node and enter your private key file.
5. Navigate to the Connection > Data node and enter *root* as auto-login username.
6. Save these session settings and hit the *Open* button.

Now you can log in to your backend instance on OS level (SLES) for monitoring, troubleshooting, or accessing files on the server.

Using a standard ssh client like unix openssh or Cygwin openssh on windows, you can use the key pair retrieved from your CAL account directly by specifying the file containing the key pair on the ssh command line with the parameter '-i':

➢ ssh -i <your_key_file_name> root@<IP of your backend instance>

The following tables list all important users on OS level:

Parameter ID	Parameter Value	Note
OS User Name	root	The default OS Administrator user for Linux SUSE.
OS Password	<none>	Use the private key (downloaded during the activation of the instance in SAP Cloud Appliance Library) for login with the root user.

The administration users for HANA and ABAP on operating system level are defined as follows:

Parameter ID	Parameter Value	Note
HANA administrator name	hdbadm	Additional user for HANA lifecycle management – start/stop, administration, functions, recovery
HANA administrator password	*<Master Password>*	The password is the same as the master password provided during instance creation in the SAP Cloud Appliance Library.
ABAP administrator name	a4hadm	Additional user for ABAP lifecycle management – start/stop, administration, functions, recovery
ABAP administrator password	*<Master Password>*	The password is the same as the master password provided during instance creation in the SAP Cloud Appliance Library.

Additional users on operating system level are:

Parameter ID	Parameter Value	Note
SAP System Administrator	sapadm	
SAP System Administrator password	*<Master Password>*	The password is the same as the master password provided during instance creation in the SAP Cloud Appliance Library.
SAP System Administrator	daaadm	
SAP System Administrator password	*<Master Password>*	The password is the same as the master password provided during instance creation in the SAP Cloud Appliance Library.

5.2. Manually starting and stopping the system

The system (ABAP server and database) is automatically started when you activate an instance in CAL. The system (ABAP server and database) is automatically stopped, when you suspend the instance in CAL. There might be nevertheless situations where you want to start or stop the ABAP server or the database manually. The next sections describe how to do this.

5.2.1. ABAP System

To check the status of the ABAP system logon as root on operating system level and execute:

```
su - a4hadm
sapcontrol -nr 00 -function GetProcessList
```

For stopping the ABAP system logon as root on operating system level and execute:

```
su - a4hadm
stopsap r3
exit
```

For starting the ABAP system logon as root on operating system level and execute (database must run):

```
su - a4hadm
startsap r3
exit
```

5.2.2. SAP HANA Database

To check the status of the database logon as root on operating system level and execute:

```
su - hdbadm
sapcontrol -nr 02 -function GetProcessList
```

For stopping the database logon as root on operating system level and execute (make sure the ABAP system has been stopped before):

```
su - hdbadm
HDB stop
exit
```

For starting the database logon as root on operating system level and execute:

```
su - hdbadm
HDB start
exit
```

5.3. Transport of Copies

The system has been set up in a way that allows you to import and export ABAP objects as transport of copies. This section describes an export/import scenario.

5.3.1. Export

To export objects with a transport of copies you have to execute the following procedure:

1. In transaction SE01 choose *Create* (F6).
2. Mark *Transport of Copies* and choose *Enter*.
3. Enter a description.
4. As transport target enter **DMY** and choose *Save*.
5. Add the objects you need into the request. You may enter them either directly or via the menu *Request/Task → Object List → Include Objects…*
6. Release the request.
7. You will find your transport files in the directories:
    a. `/usr/sap/trans/data`
    b. `/usr/sap/trans/cofiles`
    c. For the file transfer you can use sFTP/SCP clients like WinSCP with user *root* and the private key file of your backend instance (see Connecting to Your Backend on OS Level) or you can directly import the existing PuTTY connection profile.

5.3.2. Import

To import transports into the system you have to execute the following procedure:

1. Copy your transport files to:
    a. `/usr/sap/trans/data`
    b. `/usr/sap/trans/cofiles`
    c. For the file transfer you can use ftp or FTP client tools like WinSCP (see above).
2. Ensure that user *a4hadm* has sufficient rights for accessing your transport files (e.g. use the *chown a4hadm:sapsys <file>* command), otherwise the import will fail.
3. In transaction *STMS* open the *Import Overview* (F5) and double click on *A4H*.
4. In the menu select *Extras → Other Requests → Add*.
5. Use the F4 help to select your transport request.
6. Choose *Enter* and answer the question if you want to attach the request to the A4H import queue with yes.
7. Mark the request in the import queue and select `Ctrl+F11` (Import Request).
8. In the popup select for Execution “Synchronous” (for smaller request) and mark all import options.
9. Choose *Enter* and *Yes* to import your request.

6. Technical Licenses

6.1. Installation of the AS ABAP License

The Application Server ABAP comes with a temporary license that allows you to logon to the system.

As a first step before using the system you need to install a 90 days Minisap license as follows:

1. Logon to ABAP via SAP GUI with user SAP* in tenant 000.
2. Start transaction SLICENSE
3. Get a “Minisap” license at http://www.sap.com/minisap .
   As system ID choose *A4H - SAP NetWeaver 7.4 AS ABAP (Linux / SAP HANA)*.
   As hardware key use the hardware key shown in transaction SLICENSE.
4. Choose *Install new License* and select the downloaded license from step 3.
5. After license installation call transaction SECSTORE and run a check for all entries using F8. This is needed to enable RFC after the change of the installation number from INITIAL to DEMOSYSTEM.

Installing the Minisap license will change the installation number from INITIAL to DEMOSYSTEM. The developer access key for user DEVELOPER and installation number DEMOSYSTEM is already in the system and you can start developing in the customer name range (Z*, Y*).

6.2. Installation of the HANA License

The SAP HANA database comes with a temporary license. Please install a 90 days Minisap license as follows:

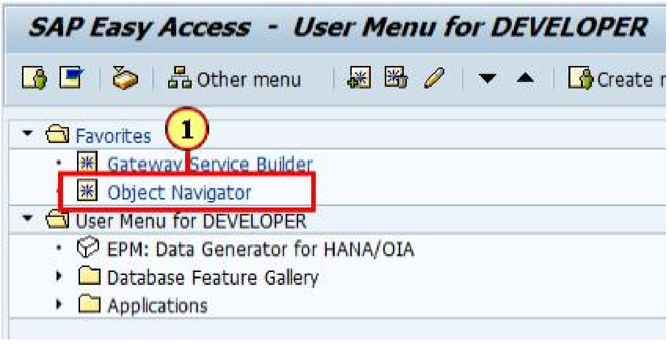
1. Start the installed SAP HANA Studio and open the HANA Development perspective.
2. From the *Systems* view right-click on your HDB (SYSTEM) connection.
3. Select *Properties.*
4. From the *Properties* dialog box, select the *License* tab.
5. Get a “Minisap” license at http://www.sap.com/minisap .
   As system ID choose *HDB - SAP HANA Platform Edition (64GB)*.
   As hardware key use the hardware key shown on the *License* tab.
6. Once you received the license key file, choose first *Delete License Key* and then *Install License Key*.

7. Tutorials and Demo Scenario

7.1. Finding the Leak – SQL Injection in user self service

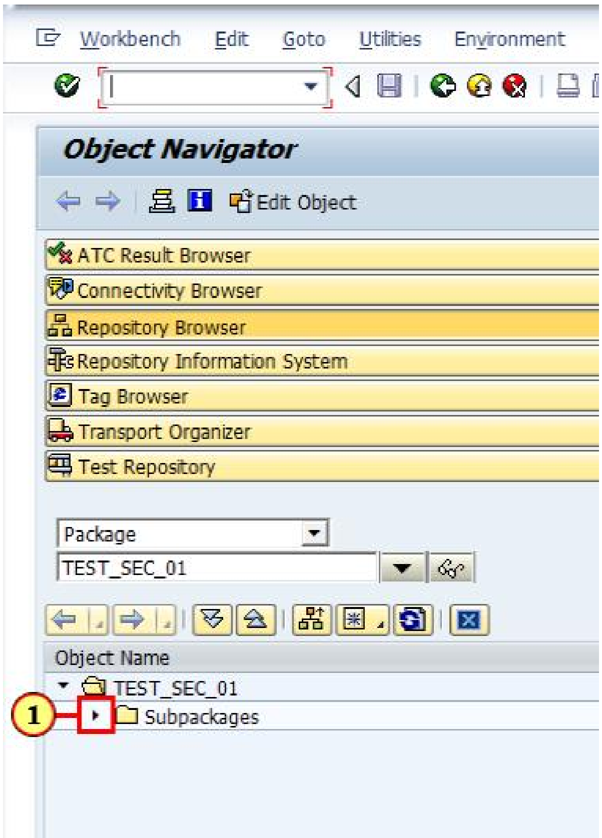
The following example assumes you are already logged in with user developer to the ABAP system A4H, client 001, with Language EN using SAP GUI either using a direct connection from your PC or using the preinstalled SAP-GUI on the windows front-end.

7.1.1. Start the ABAP development Workbench

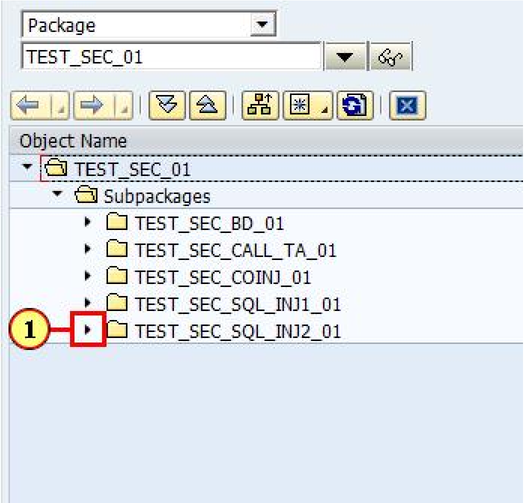


The transaction **Object Navigator** is started by a double-click on the navigation link.

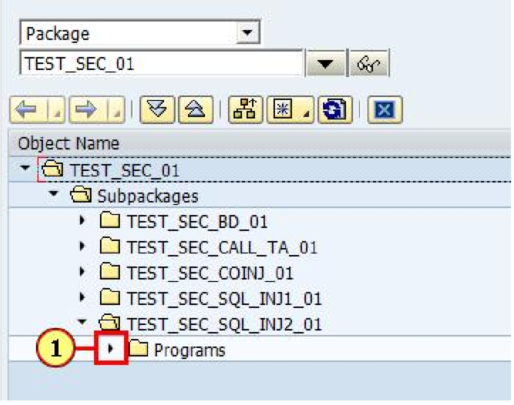
7.1.2. Navigate to the coding



(1) Click ‣ to expand Subpackages.

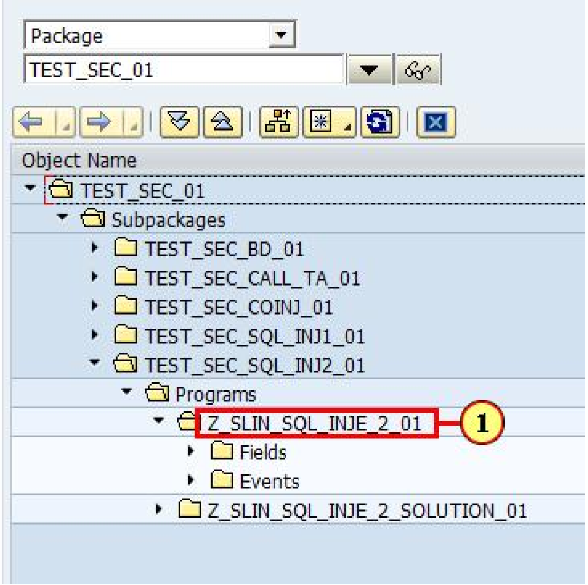


Click ▸ to expand package TEST_SEC_SQL_INJ2_01.



Click ▸ to expand Programs.

7.1.3. Select the report to run



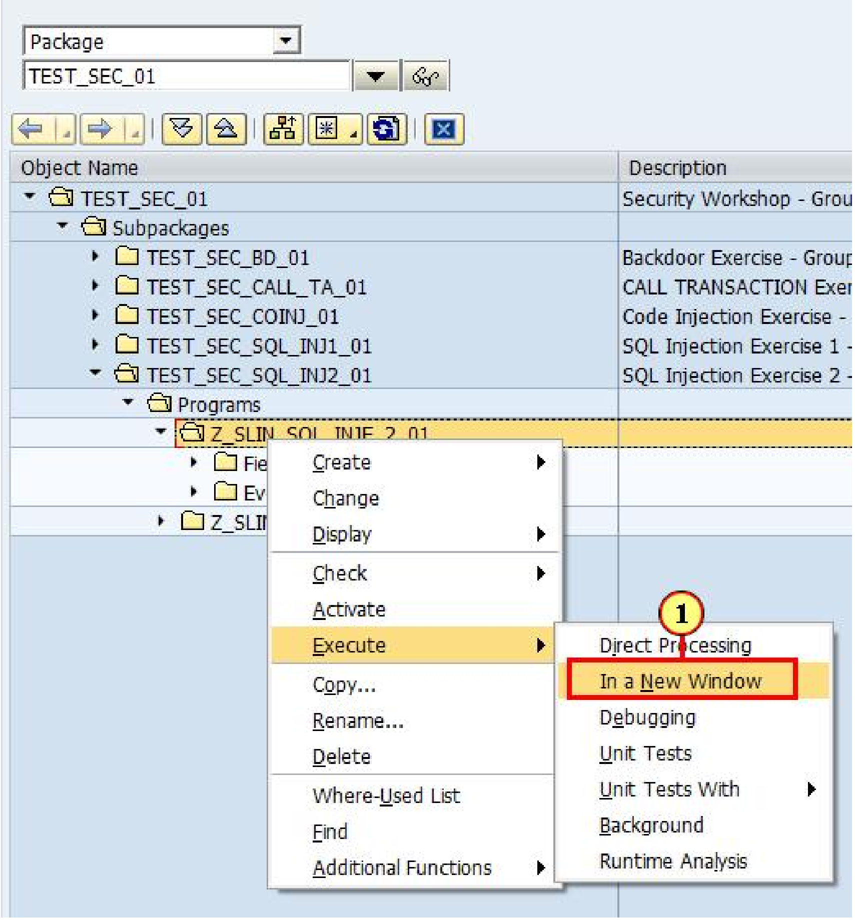
Package

TEST_SEC_01

Object Name
TEST_SEC_01
Subpackages
TEST_SEC_BD_01
TEST_SEC_CALL_TA_01
TEST_SEC_COINJ_01
TEST_SEC_SQL_INJ1_01
TEST_SEC_SQL_INJ2_01
Programs
Z_SLIN_SQL_INJE_2_01
Fields
Events
Z_SLIN_SQL_INJE_2_SOLUTION_01

(1) Right-clicking on Z SLIN SQL INJE 2 01 with the mouse opens a shortcut menu.

7.1.4. Start the report in a new window



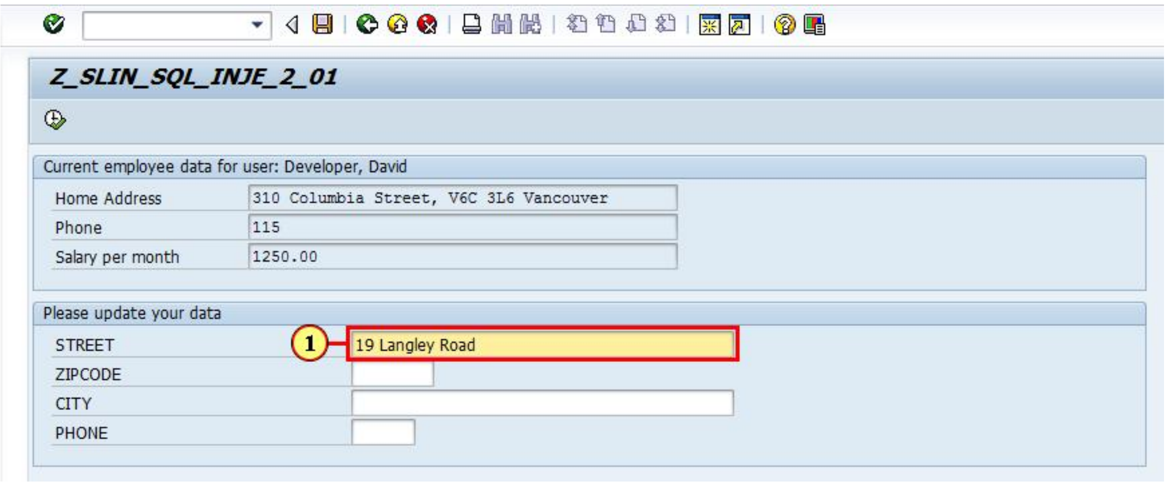
Package

TEST_SEC_01

Object Name	Description
TEST_SEC_01	Security Workshop - Grou
Subpackages	
TEST_SEC_BD_01	Backdoor Exercise - Group
TEST_SEC_CALL_TA_01	CALL TRANSACTION Exer
TEST_SEC_COINJ_01	Code Injection Exercise -
TEST_SEC_SQL_INJ1_01	SQL Injection Exercise 1 -
TEST_SEC_SQL_INJ2_01	SQL Injection Exercise 2 -
Programs	
Z_SLIN_SQL_INJE_2_01	
Fie	
Ev	
Z_SLII	

Clicking the **Execute ->** In a New Window menu item executes it.

7.1.5. Using the self-service



Z_SLIN_SQL_INJE_2_01

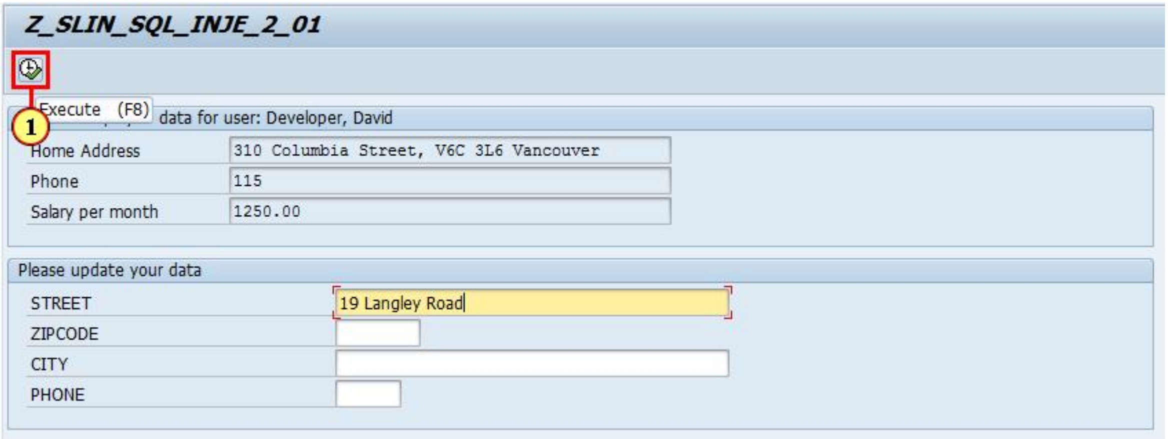
Current employee data for user: Developer, David

Home Address	310 Columbia Street, V6C 3L6 Vancouver
Phone	115
Salary per month	1250.00

Please update your data

STREET	1 19 Langley Road
ZIPCODE	
CITY	
PHONE	

Enter a new street address in the field Street, for instance 19 Langley Road.



Z_SLIN_SQL_INJE_2_01

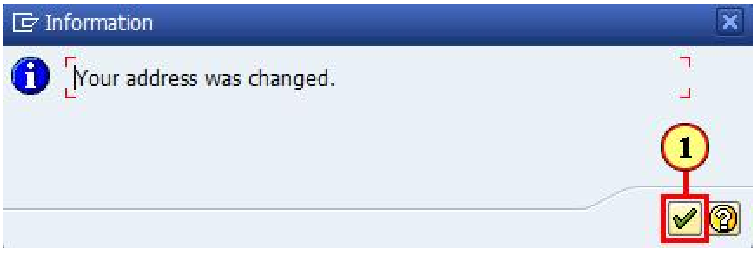
Current employee data for user: Developer, David

Home Address	310 Columbia Street, V6C 3L6 Vancouver
Phone	115
Salary per month	1250.00

Please update your data

STREET	19 Langley Road
ZIPCODE	
CITY	
PHONE	

Click **Execute** .

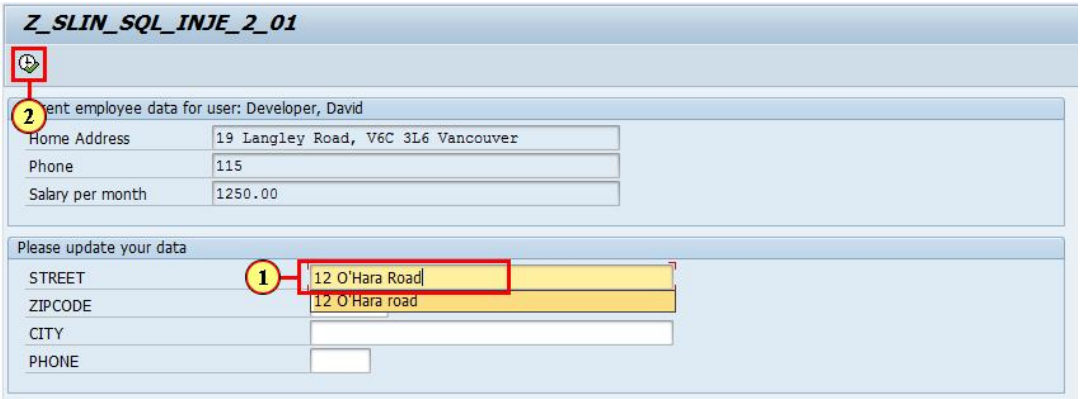


Information

Your address was changed.

The system will confirm the update. Click **Continue** .

7.1.6. Using the self-service a second time



Z_SLIN_SQL_INJE_2_01

ent employee data for user: Developer, David

Home Address	19 Langley Road, V6C 3L6 Vancouver
Phone	115
Salary per month	1250.00

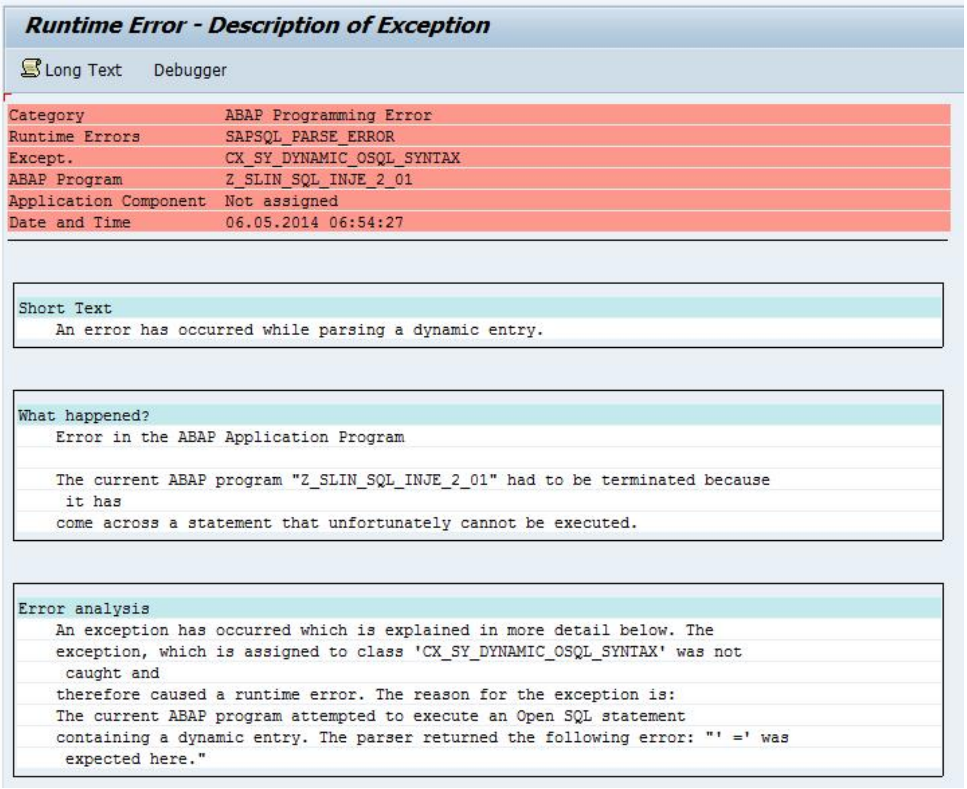
Please update your data

STREET	12 O'Hara Road
ZIPCODE	
CITY	
PHONE	

(1) Into the field STREET, enter

12 O'Hara Road

(2) Click **Execute** .



Runtime Error - Description of Exception

```
Category              ABAP Programming Error
Runtime Errors        SAPSQL_PARSE_ERROR
Except.               CX_SY_DYNAMIC_OSQL_SYNTAX
ABAP Program          Z_SLIN_SQL_INJE_2_01
Application Component  Not assigned
Date and Time         06.05.2014 06:54:27
```

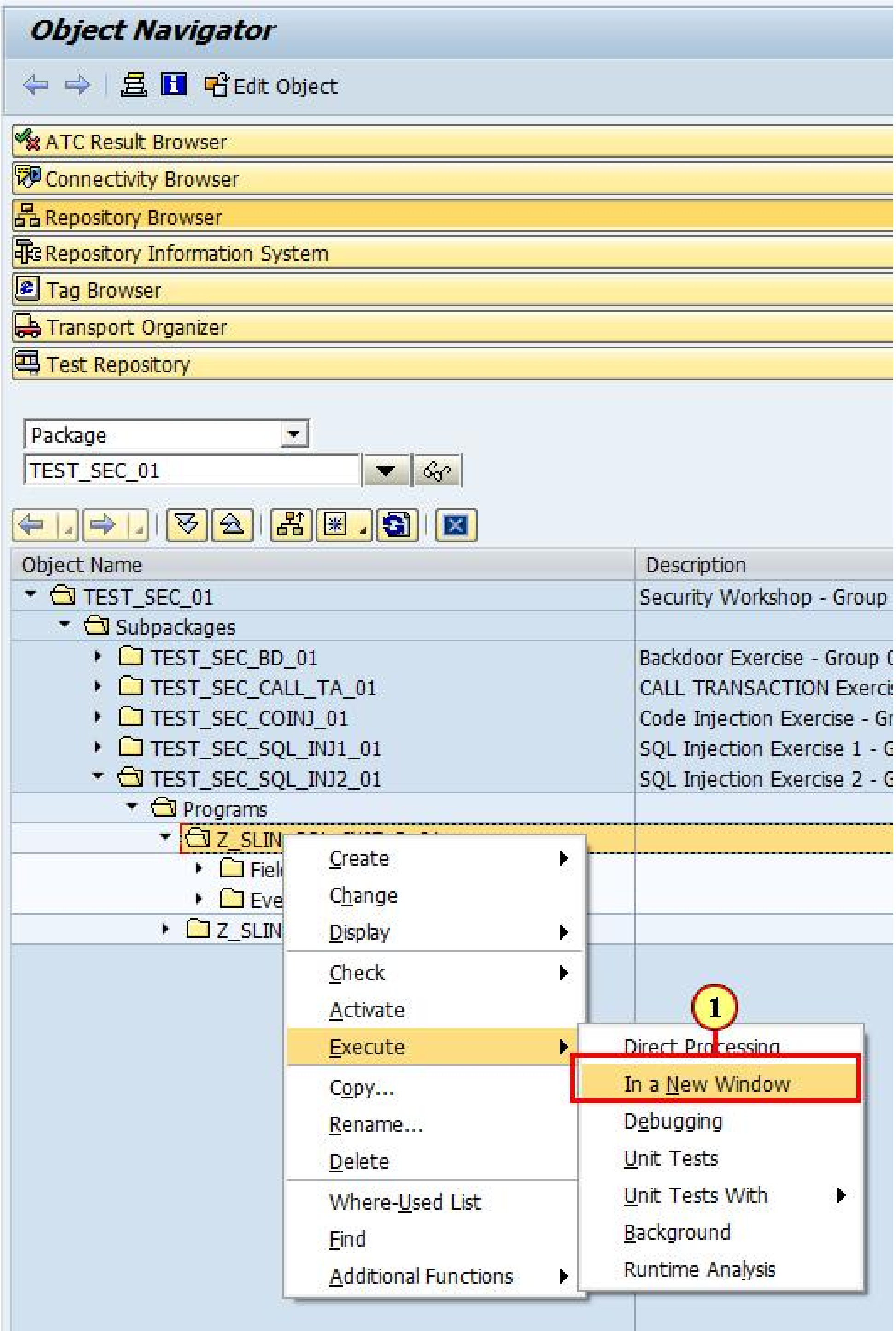
```
Short Text
    An error has occurred while parsing a dynamic entry.
```

```
What happened?
    Error in the ABAP Application Program

    The current ABAP program "Z_SLIN_SQL_INJE_2_01" had to be terminated because
     it has
    come across a statement that unfortunately cannot be executed.
```

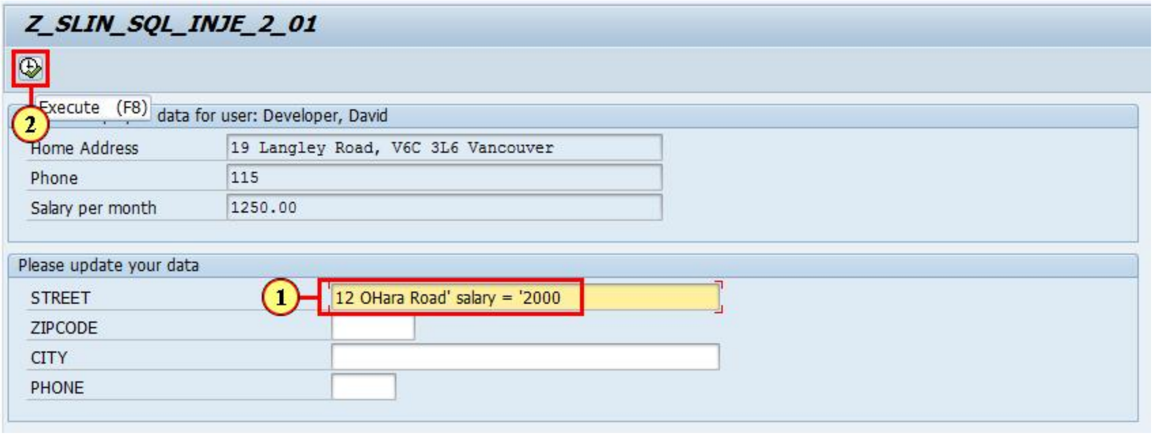
```
Error analysis
    An exception has occurred which is explained in more detail below. The
    exception, which is assigned to class 'CX_SY_DYNAMIC_OSQL_SYNTAX' was not
     caught and
    therefore caused a runtime error. The reason for the exception is:
    The current ABAP program attempted to execute an Open SQL statement
    containing a dynamic entry. The parser returned the following error: "' =' was
     expected here."
```

You will get a dump displayed, indicating, that there was an error in the program executed. Please close the window to get back to where you started the program from.

7.1.7. Exploit the security issue



Back on the initial screen from the Object Navigator (SE80), restart the application by right-click on Z_SLIN_SQL_INJE_2_01 and clicking the Execute-> In a New Window menu item.



Z_SLIN_SQL_INJE_2_01

data for user: Developer, David

Home Address	19 Langley Road, V6C 3L6 Vancouver
Phone	115
Salary per month	1250.00

Please update your data

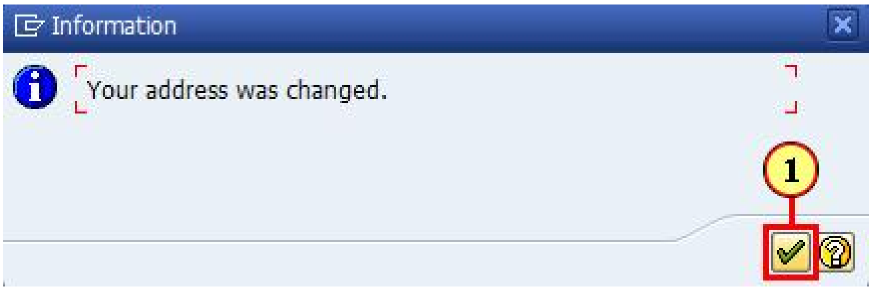
STREET	12 OHara Road' salary = '2000
ZIPCODE	
CITY	
PHONE	

(1) Enter exactly

12 OHara Road' salary = '2000

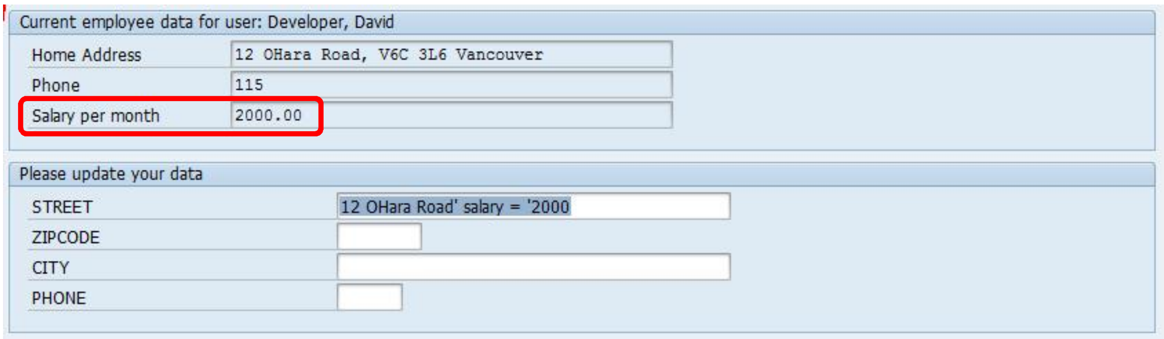
in the field street. Please pay special attention to the placement of the quotes!

(2) Click **Execute** .



Your address was changed.

The information was accepted. Click **Continue** ✓ to see the result.



Current employee data for user: Developer, David

Home Address	12 OHara Road, V6C 3L6 Vancouver
Phone	115
Salary per month	2000.00

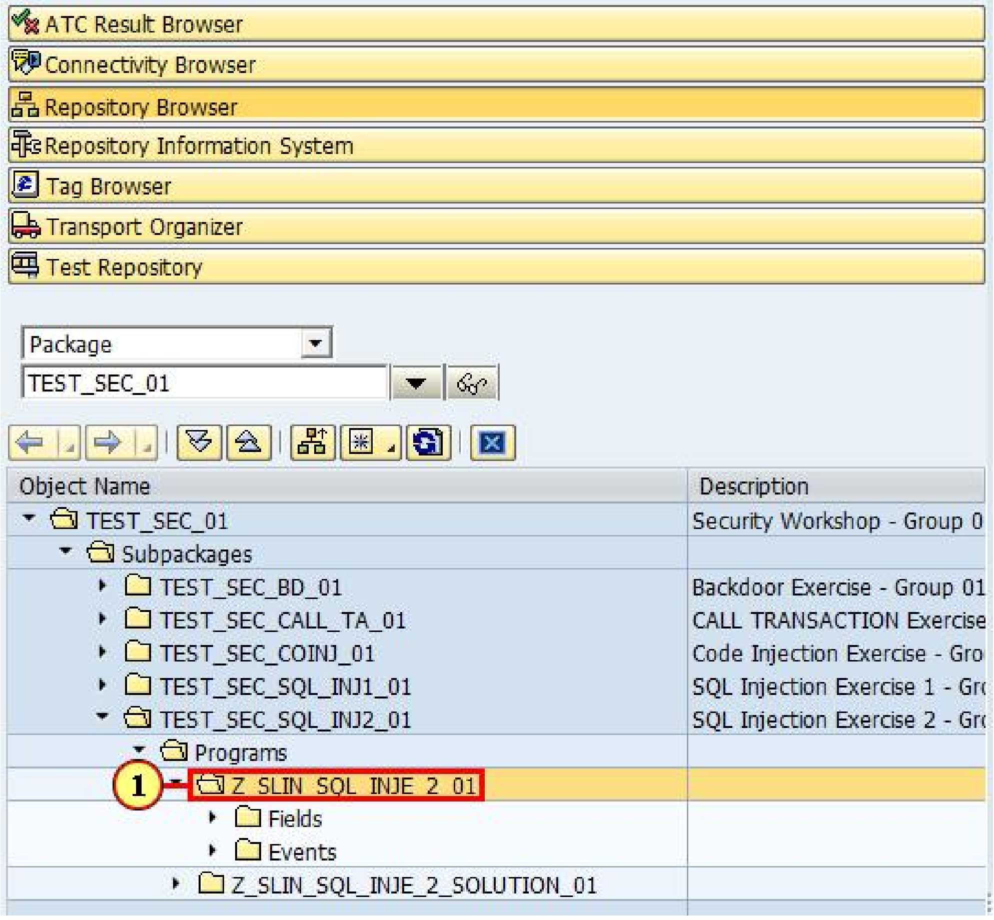
Please update your data

STREET	12 OHara Road' salary = '2000
ZIPCODE	
CITY	
PHONE	

Although only data into the field street has been entered, the salary data has changed as well! This is a typical form of an SQL Injection.

You can now close this window.

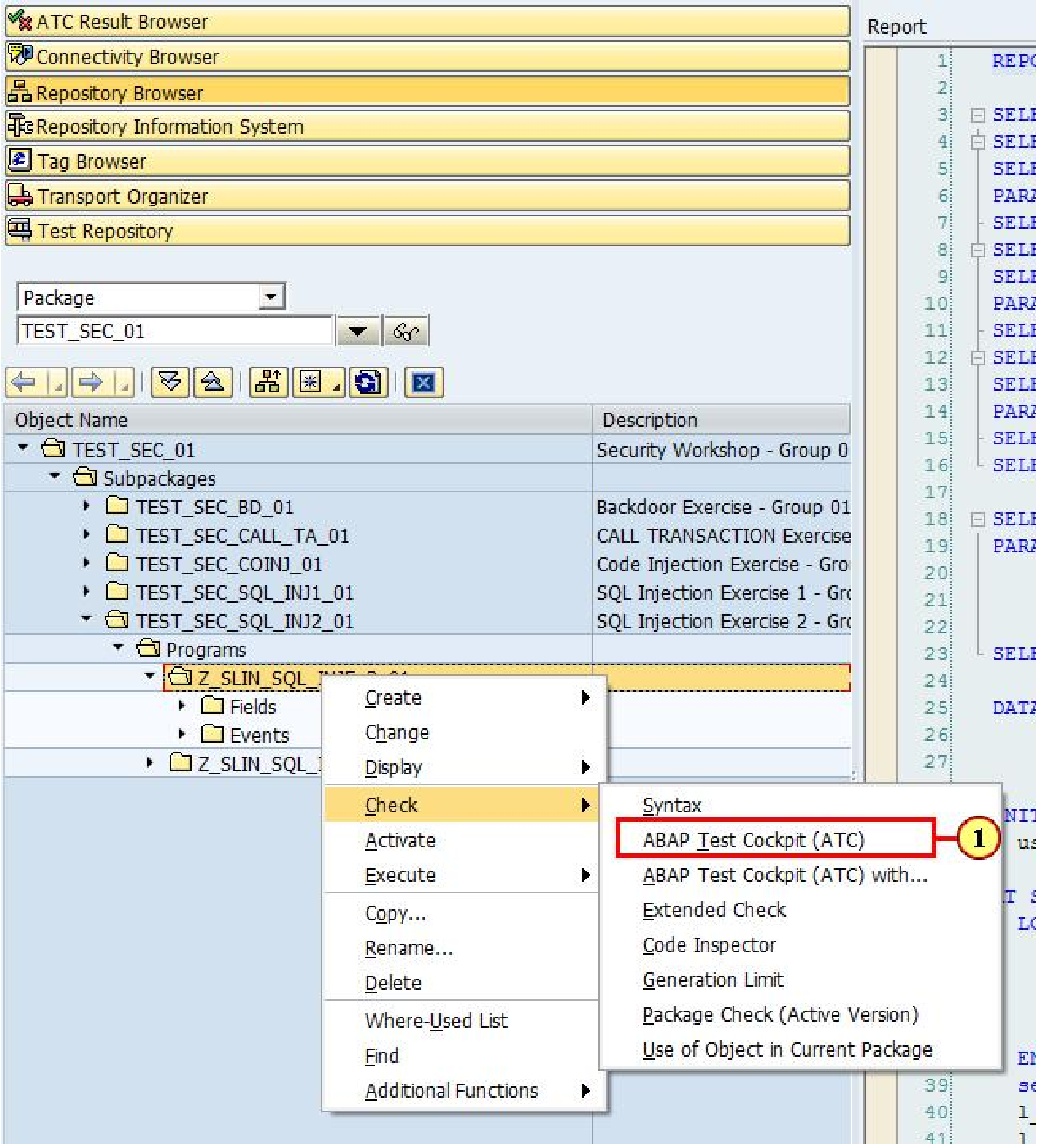
7.1.8. Finding the leak - check the sources



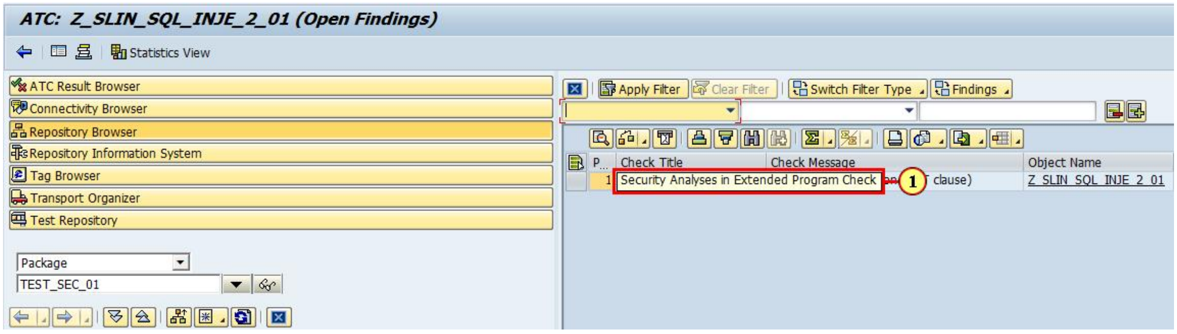
Package
TEST_SEC_01

Object Name	Description
TEST_SEC_01	Security Workshop - Group 0
Subpackages	
TEST_SEC_BD_01	Backdoor Exercise - Group 01
TEST_SEC_CALL_TA_01	CALL TRANSACTION Exercise
TEST_SEC_COINJ_01	Code Injection Exercise - Gro
TEST_SEC_SQL_INJ1_01	SQL Injection Exercise 1 - Gr
TEST_SEC_SQL_INJ2_01	SQL Injection Exercise 2 - Gr
Programs	
Z_SLIN_SQL_INJE_2_01	
Fields	
Events	
Z_SLIN_SQL_INJE_2_SOLUTION_01	

(1) Right-click on Z_SLIN_SQL_INJE_2_ to display the context menu.



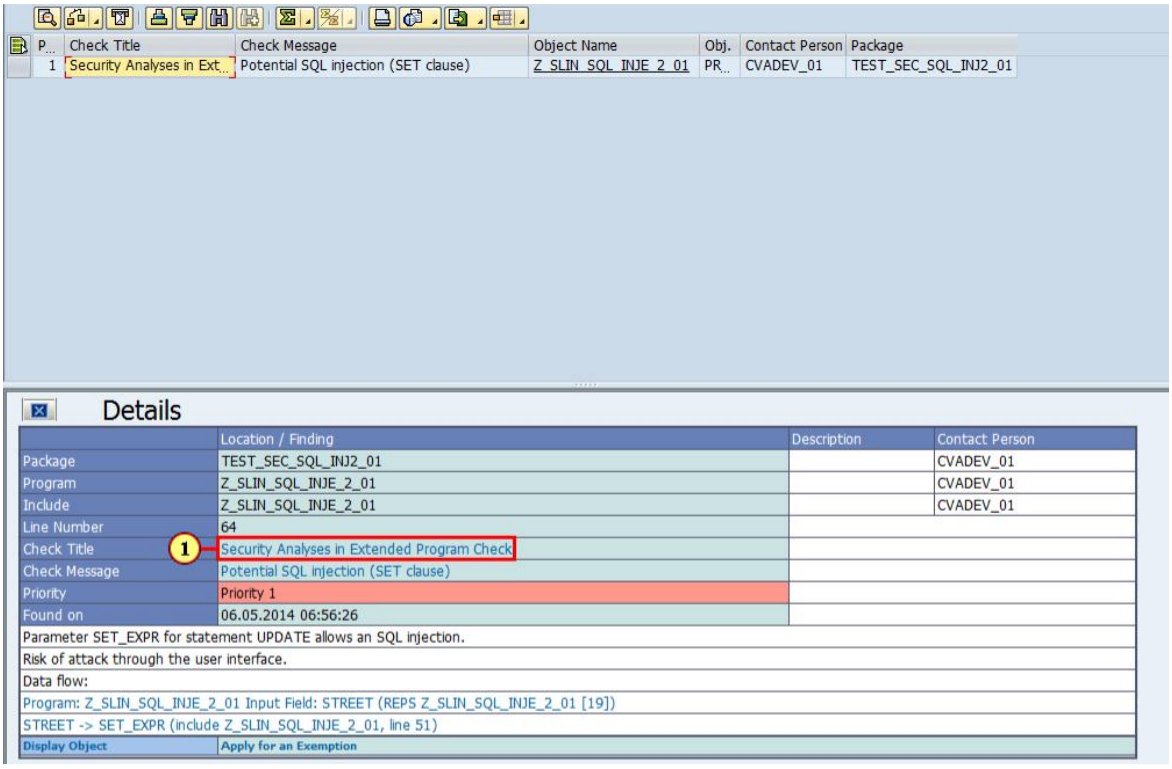
(1) Click on Check --> **ABAP Test Cockpit (ATC)** to execute the security checks. The system will now scan the program for security issues.



ATC: Z_SLIN_SQL_INJE_2_01 (Open Findings)

P...	Check Title	Check Message	Object Name
1	Security Analyses in Extended Program Check	... clause)	Z_SLIN_SQL_INJE_2_01

ATC will display the check results, after the checks have been done. There will be one finding displayed. In column **Check Title**, double-click Security Analyses in Extended Program Check to display the details for the result found.



P...	Check Title	Check Message	Object Name	Obj.	Contact Person	Package
1	Security Analyses in Ext...	Potential SQL injection (SET clause)	Z_SLIN_SQL_INJE_2_01	PR...	CVADEV_01	TEST_SEC_SQL_INJ2_01

Details

	Location / Finding	Description	Contact Person
Package	TEST_SEC_SQL_INJ2_01		CVADEV_01
Program	Z_SLIN_SQL_INJE_2_01		CVADEV_01
Include	Z_SLIN_SQL_INJE_2_01		CVADEV_01
Line Number	64		
Check Title	Security Analyses in Extended Program Check		
Check Message	Potential SQL injection (SET clause)		
Priority	Priority 1		
Found on	06.05.2014 06:56:26		

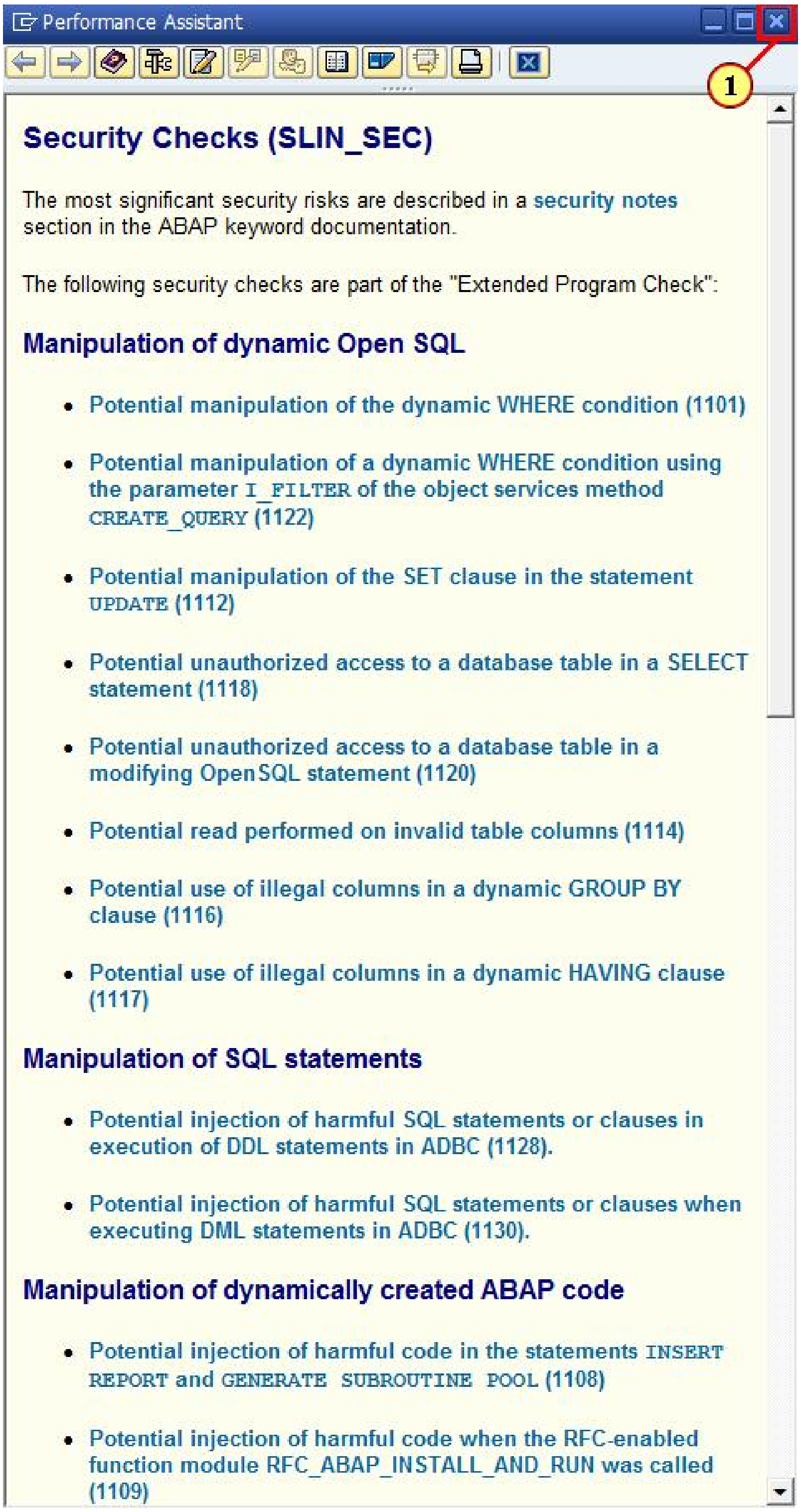
Parameter SET_EXPR for statement UPDATE allows an SQL injection.
Risk of attack through the user interface.
Data flow:
Program: Z_SLIN_SQL_INJE_2_01 Input Field: STREET (REPS Z_SLIN_SQL_INJE_2_01 [19])
STREET -> SET_EXPR (include Z_SLIN_SQL_INJE_2_01, line 51)

Display Object | Apply for an Exemption

ATC will now display the details for the issue found. On the details screen, there are various types of information available, like for instance information about the checks in general, the specific results and navigational links into the source code.

Click on the **Check Title** Security Analyses in Extended Program Check to get information on the security checks in general.

7.1.9. Display list of security checks of SLIN_SEC



Security Checks (SLIN_SEC)

The most significant security risks are described in a security notes section in the ABAP keyword documentation.

The following security checks are part of the "Extended Program Check":

Manipulation of dynamic Open SQL

- Potential manipulation of the dynamic WHERE condition (1101)
- Potential manipulation of a dynamic WHERE condition using the parameter I_FILTER of the object services method CREATE_QUERY (1122)
- Potential manipulation of the SET clause in the statement UPDATE (1112)
- Potential unauthorized access to a database table in a SELECT statement (1118)
- Potential unauthorized access to a database table in a modifying OpenSQL statement (1120)
- Potential read performed on invalid table columns (1114)
- Potential use of illegal columns in a dynamic GROUP BY clause (1116)
- Potential use of illegal columns in a dynamic HAVING clause (1117)

Manipulation of SQL statements

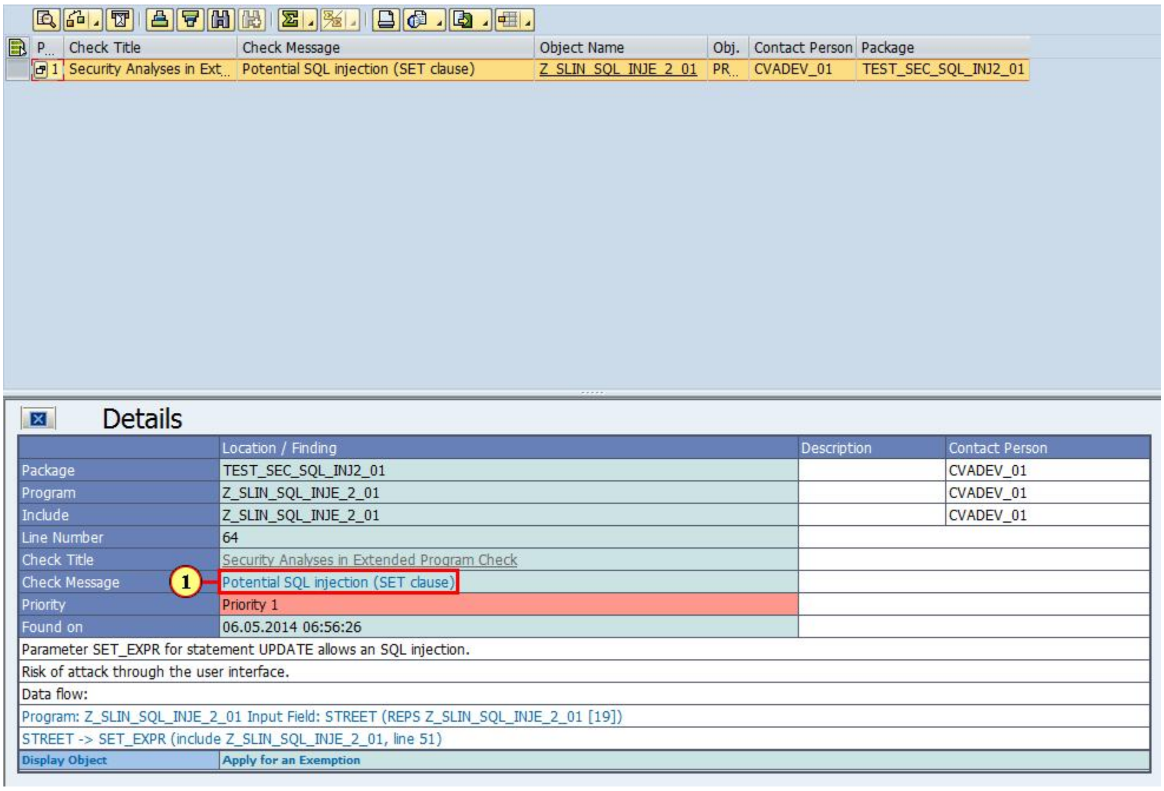
- Potential injection of harmful SQL statements or clauses in execution of DDL statements in ADBC (1128).
- Potential injection of harmful SQL statements or clauses when executing DML statements in ADBC (1130).

Manipulation of dynamically created ABAP code

- Potential injection of harmful code in the statements INSERT REPORT and GENERATE SUBROUTINE POOL (1108)
- Potential injection of harmful code when the RFC-enabled function module RFC_ABAP_INSTALL_AND_RUN was called (1109)

The system now displays a list of checks for this group of security checks. You can navigate into the details of each check by clicking any of the checks. Close the window when done.

7.1.10. Display documentation of the issue



P...	Check Title	Check Message	Object Name	Obj.	Contact Person	Package
1	Security Analyses in Ext...	Potential SQL injection (SET clause)	Z_SLIN_SQL_INJE_2_01	PR...	CVADEV_01	TEST_SEC_SQL_INJ2_01

Details

	Location / Finding	Description	Contact Person
Package	TEST_SEC_SQL_INJ2_01		CVADEV_01
Program	Z_SLIN_SQL_INJE_2_01		CVADEV_01
Include	Z_SLIN_SQL_INJE_2_01		CVADEV_01
Line Number	64		
Check Title	Security Analyses in Extended Program Check		
Check Message	Potential SQL injection (SET clause)		
Priority	Priority 1		
Found on	06.05.2014 06:56:26		

Parameter SET_EXPR for statement UPDATE allows an SQL injection.

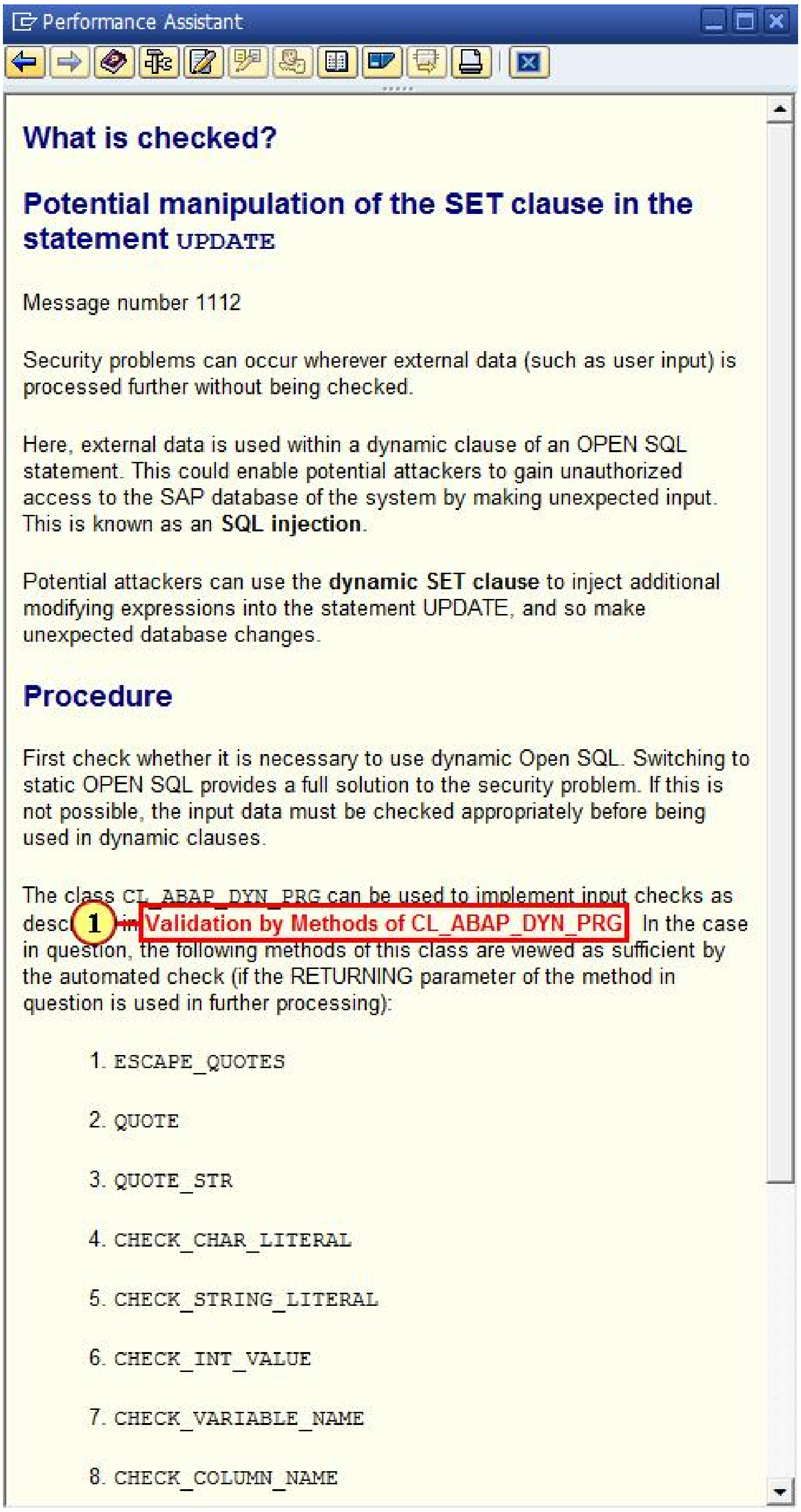
Risk of attack through the user interface.

Data flow:

Program: Z_SLIN_SQL_INJE_2_01 Input Field: STREET (REPS Z_SLIN_SQL_INJE_2_01 [19])

STREET -> SET_EXPR (include Z_SLIN_SQL_INJE_2_01, line 51)

Click on the check message Potential SQL injection (SET clause) to directly access the information relevant for this check result.



What is checked?

Potential manipulation of the SET clause in the statement UPDATE

Message number 1112

Security problems can occur wherever external data (such as user input) is processed further without being checked.

Here, external data is used within a dynamic clause of an OPEN SQL statement. This could enable potential attackers to gain unauthorized access to the SAP database of the system by making unexpected input. This is known as an **SQL injection**.

Potential attackers can use the **dynamic SET clause** to inject additional modifying expressions into the statement UPDATE, and so make unexpected database changes.

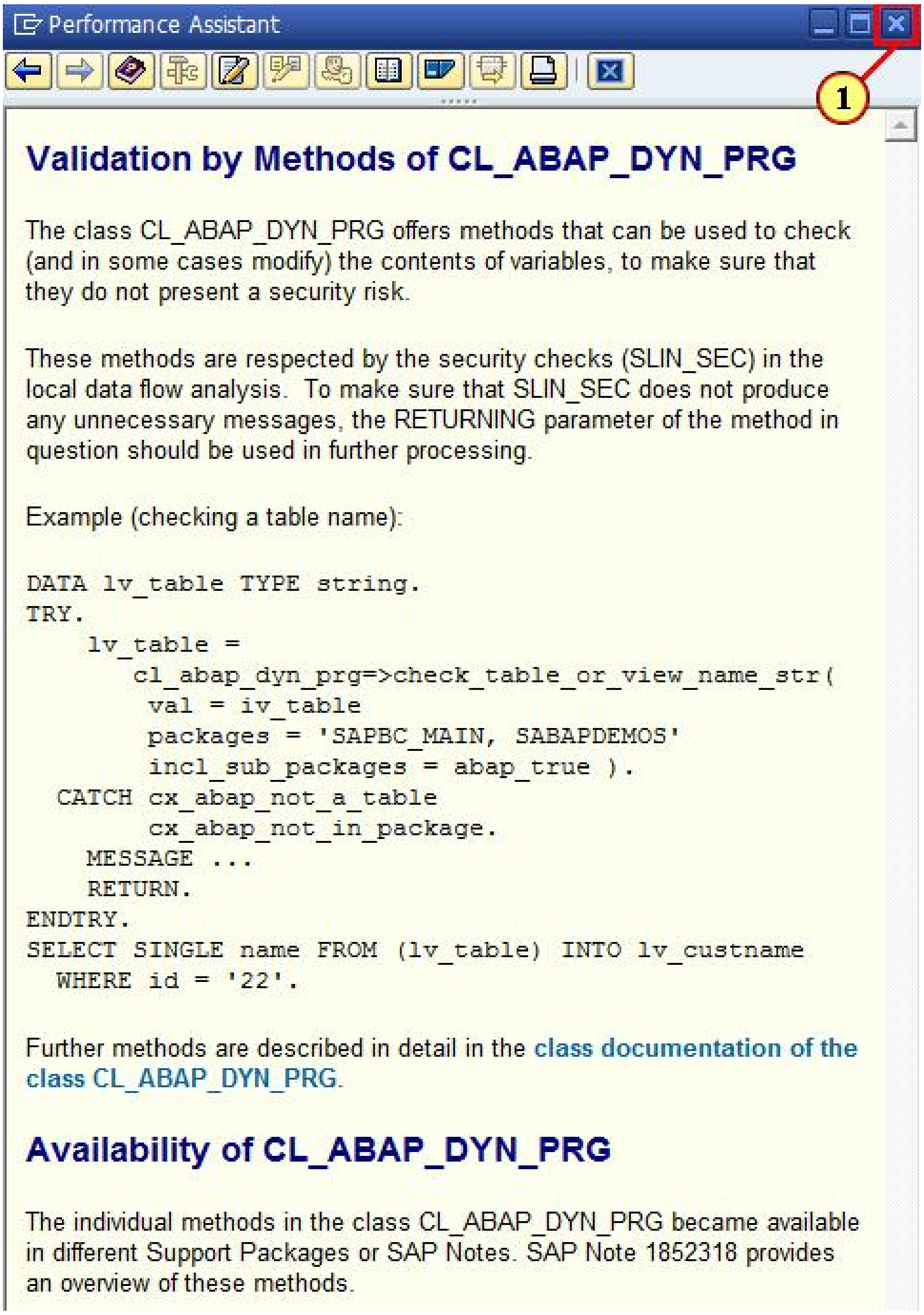
Procedure

First check whether it is necessary to use dynamic Open SQL. Switching to static OPEN SQL provides a full solution to the security problem. If this is not possible, the input data must be checked appropriately before being used in dynamic clauses.

The class CL_ABAP_DYN_PRG can be used to implement input checks as desc[1] in Validation by Methods of CL_ABAP_DYN_PRG. In the case in question, the following methods of this class are viewed as sufficient by the automated check (if the RETURNING parameter of the method in question is used in further processing):

1. ESCAPE_QUOTES
2. QUOTE
3. QUOTE_STR
4. CHECK_CHAR_LITERAL
5. CHECK_STRING_LITERAL
6. CHECK_INT_VALUE
7. CHECK_VARIABLE_NAME
8. CHECK_COLUMN_NAME

The system displays the information related to the finding. This includes the risk caused by this coding and hints on how to improve the coding. Further navigation is available, for instance to get example coding. Click on **Validation by Methods of CL_ABAP_DYN_PRG** to inspect this information.



Validation by Methods of CL_ABAP_DYN_PRG

The class CL_ABAP_DYN_PRG offers methods that can be used to check (and in some cases modify) the contents of variables, to make sure that they do not present a security risk.

These methods are respected by the security checks (SLIN_SEC) in the local data flow analysis. To make sure that SLIN_SEC does not produce any unnecessary messages, the RETURNING parameter of the method in question should be used in further processing.

Example (checking a table name):

```
DATA lv_table TYPE string.
TRY.
    lv_table =
       cl_abap_dyn_prg=>check_table_or_view_name_str(
        val = iv_table
        packages = 'SAPBC_MAIN, SABAPDEMOS'
        incl_sub_packages = abap_true ).
  CATCH cx_abap_not_a_table
        cx_abap_not_in_package.
    MESSAGE ...
    RETURN.
ENDTRY.
SELECT SINGLE name FROM (lv_table) INTO lv_custname
  WHERE id = '22'.
```

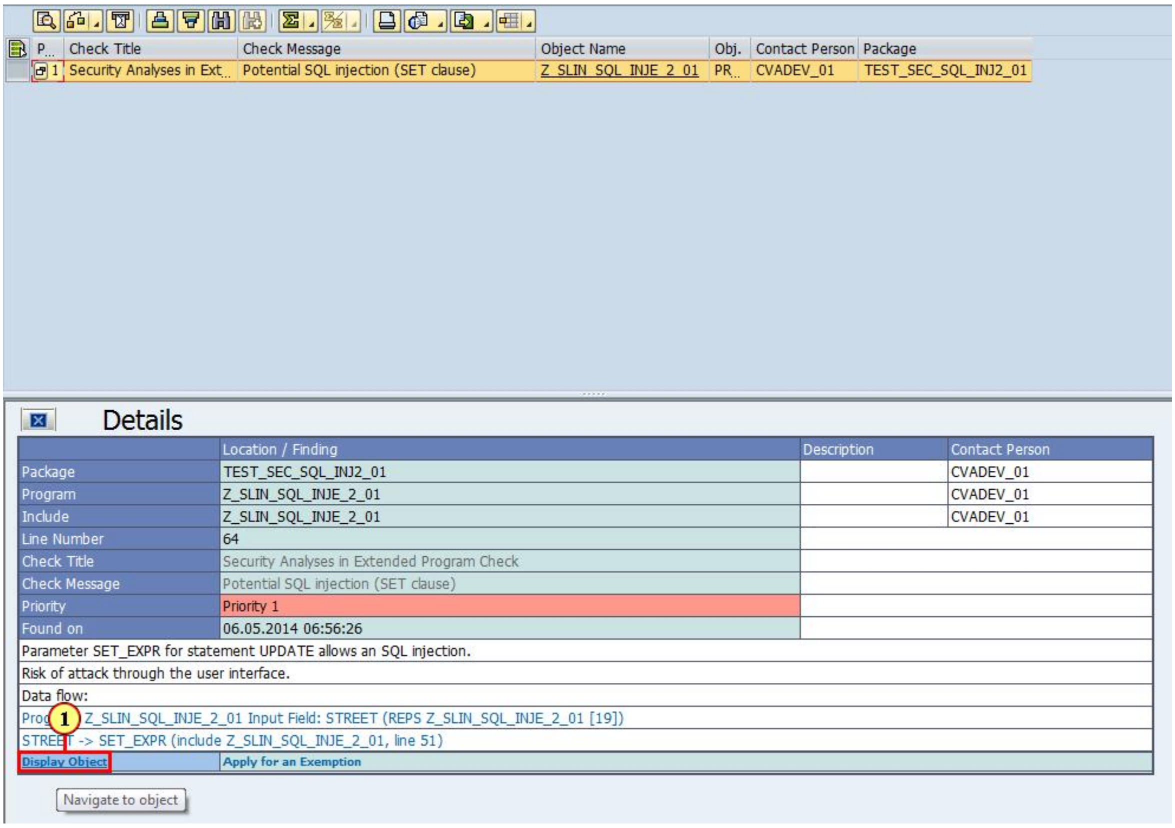
Further methods are described in detail in the class documentation of the class CL_ABAP_DYN_PRG.

Availability of CL_ABAP_DYN_PRG

The individual methods in the class CL_ABAP_DYN_PRG became available in different Support Packages or SAP Notes. SAP Note 1852318 provides an overview of these methods.

You can further navigate to get even more details on how to make use of the sanitization class CL_ABAP_DYN_PRG. Close the window, when done.

7.1.11. Navigation to the sources



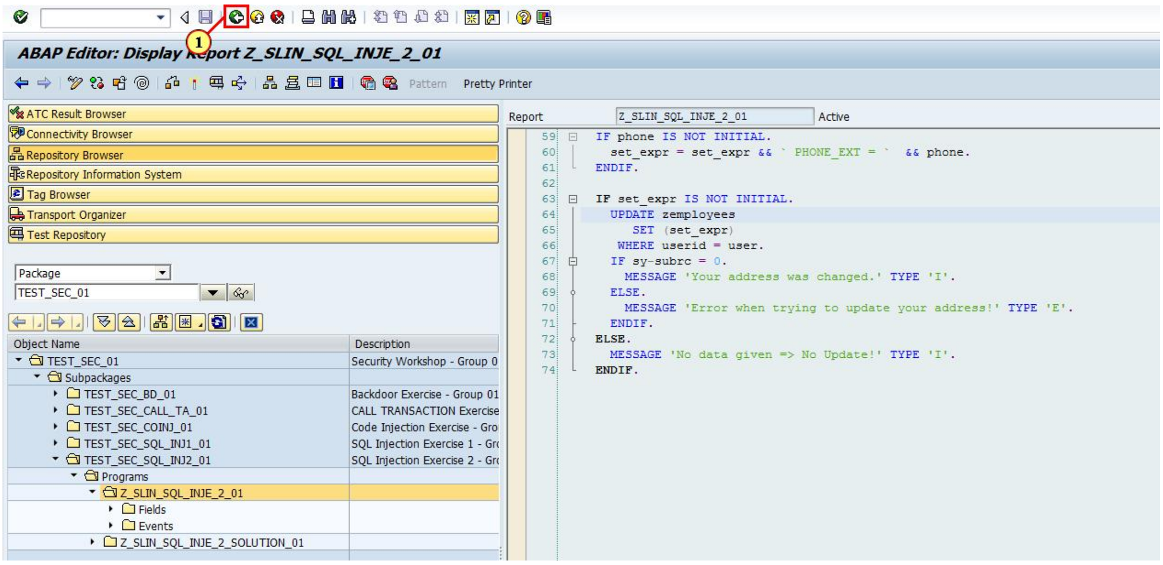
P...	Check Title	Check Message	Object Name	Obj.	Contact Person	Package
1	Security Analyses in Ext...	Potential SQL injection (SET clause)	Z_SLIN_SQL_INJE_2_01	PR...	CVADEV_01	TEST_SEC_SQL_INJ2_01

Details

	Location / Finding	Description	Contact Person
Package	TEST_SEC_SQL_INJ2_01		CVADEV_01
Program	Z_SLIN_SQL_INJE_2_01		CVADEV_01
Include	Z_SLIN_SQL_INJE_2_01		CVADEV_01
Line Number	64		
Check Title	Security Analyses in Extended Program Check		
Check Message	Potential SQL injection (SET clause)		
Priority	Priority 1		
Found on	06.05.2014 06:56:26		

Parameter SET_EXPR for statement UPDATE allows an SQL injection.
Risk of attack through the user interface.
Data flow:
Prog 1 Z_SLIN_SQL_INJE_2_01 Input Field: STREET (REPS Z_SLIN_SQL_INJE_2_01 [19])
STREET -> SET_EXPR (include Z_SLIN_SQL_INJE_2_01, line 51)

You can directly navigate to the insecure statement, by clicking Display Object



ABAP Editor: Display Report Z_SLIN_SQL_INJE_2_01

Report Z_SLIN_SQL_INJE_2_01 Active

```
IF phone IS NOT INITIAL.
  set_expr = set_expr && ` PHONE_EXT = `  && phone.
ENDIF.

IF set_expr IS NOT INITIAL.
  UPDATE zemployees
     SET (set_expr)
   WHERE userid = user.
  IF sy-subrc = 0.
    MESSAGE 'Your address was changed.' TYPE 'I'.
  ELSE.
    MESSAGE 'Error when trying to update your address!' TYPE 'E'.
  ENDIF.
ELSE.
  MESSAGE 'No data given => No Update!' TYPE 'I'.
ENDIF.
```

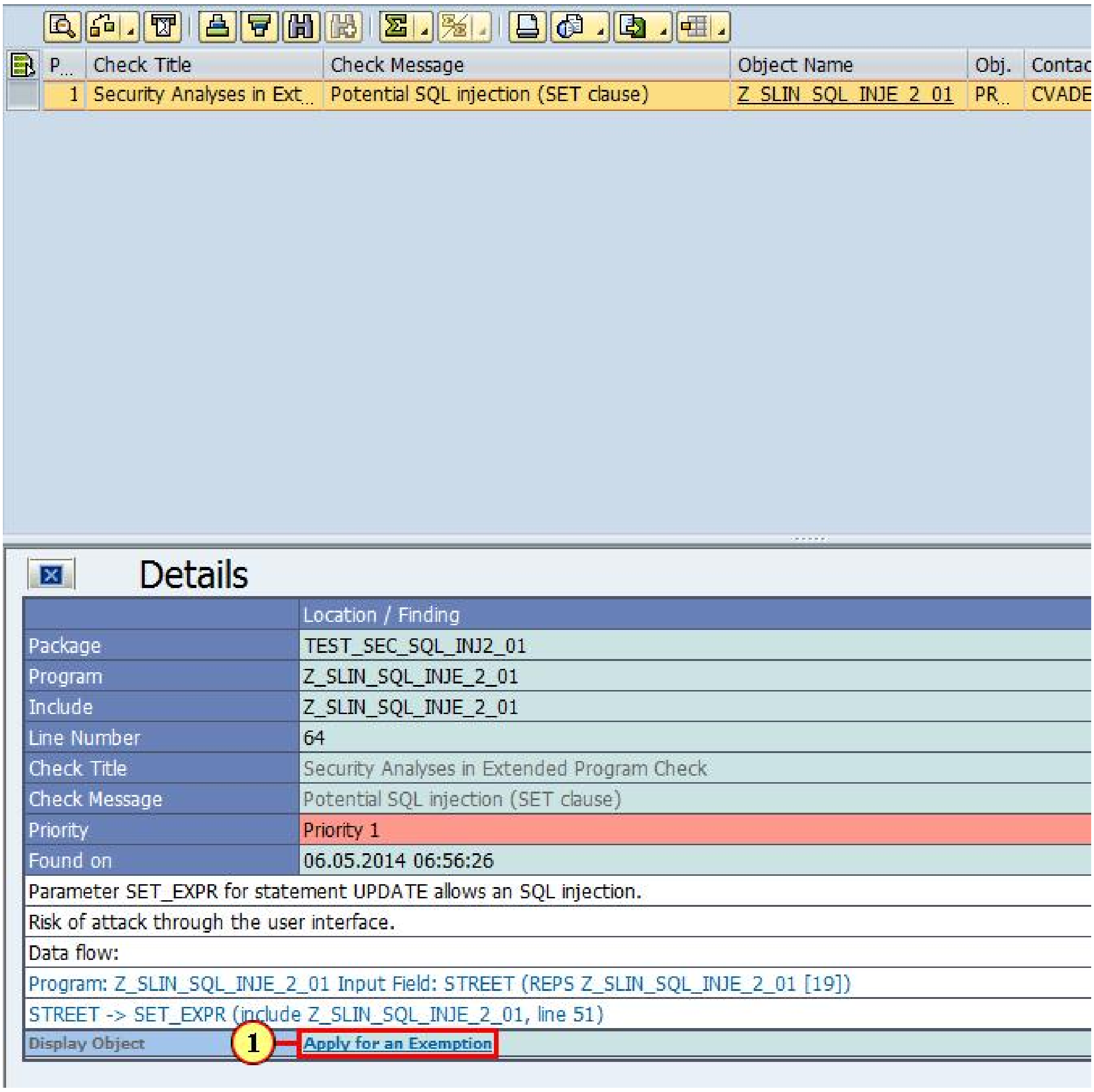
Object Name	Description
TEST_SEC_01	Security Workshop - Group 0
Subpackages	
TEST_SEC_BD_01	Backdoor Exercise - Group 01
TEST_SEC_CALL_TA_01	CALL TRANSACTION Exercise
TEST_SEC_COINJ_01	Code Injection Exercise - Gro
TEST_SEC_SQL_INJ1_01	SQL Injection Exercise 1 - Gr
TEST_SEC_SQL_INJ2_01	SQL Injection Exercise 2 - Gr
Programs	
Z_SLIN_SQL_INJE_2_01	
Fields	
Events	
Z_SLIN_SQL_INJE_2_SOLUTION_01	

The cursor will be positioned at the line of the insecure statement, in this case the statement **UPDATE zemployees.**

However in case of SQL injection issues, this is usually not the right place to fix the coding.

Click to get back to the previous view.

7.1.12. Applying for an exemption



P...	Check Title	Check Message	Object Name	Obj.	Contac
1	Security Analyses in Ext...	Potential SQL injection (SET clause)	Z_SLIN_SQL_INJE_2_01	PR...	CVADE

Details

	Location / Finding
Package	TEST_SEC_SQL_INJ2_01
Program	Z_SLIN_SQL_INJE_2_01
Include	Z_SLIN_SQL_INJE_2_01
Line Number	64
Check Title	Security Analyses in Extended Program Check
Check Message	Potential SQL injection (SET clause)
Priority	Priority 1
Found on	06.05.2014 06:56:26

Parameter SET_EXPR for statement UPDATE allows an SQL injection.

Risk of attack through the user interface.

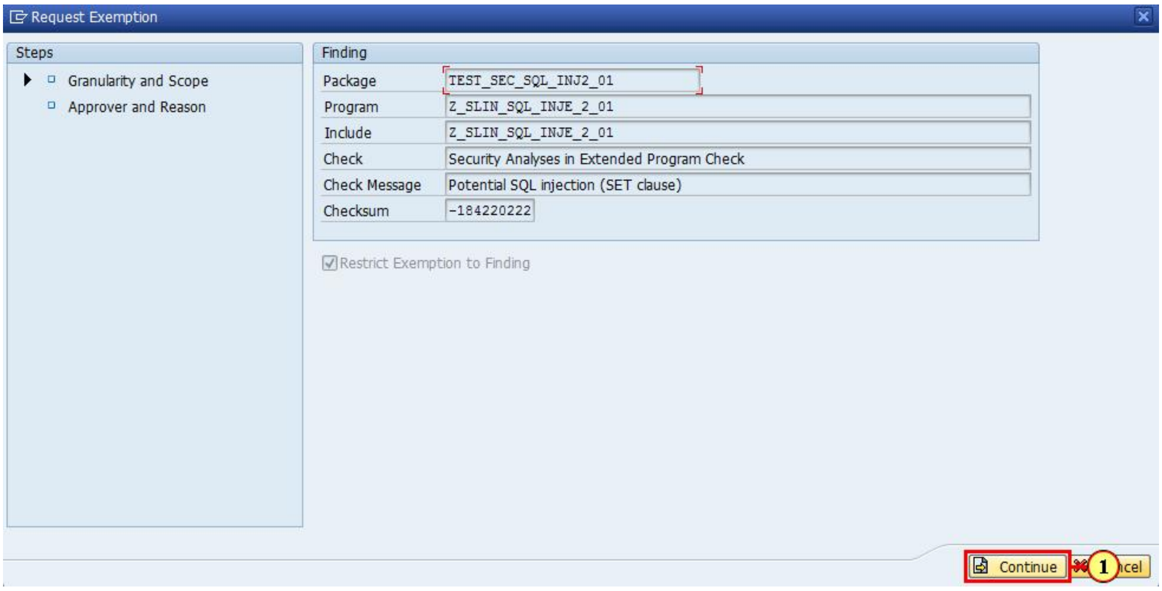
Data flow:

Program: Z_SLIN_SQL_INJE_2_01 Input Field: STREET (REPS Z_SLIN_SQL_INJE_2_01 [19])

STREET -> SET_EXPR (include Z_SLIN_SQL_INJE_2_01, line 51)

In case a developer believes the message being a false positive (secure code, which the system incorrectly believes to be dangerous), he would be able to apply for an exemption.

Click on Apply for an Exemption to start creating such an exemption.



Request Exemption

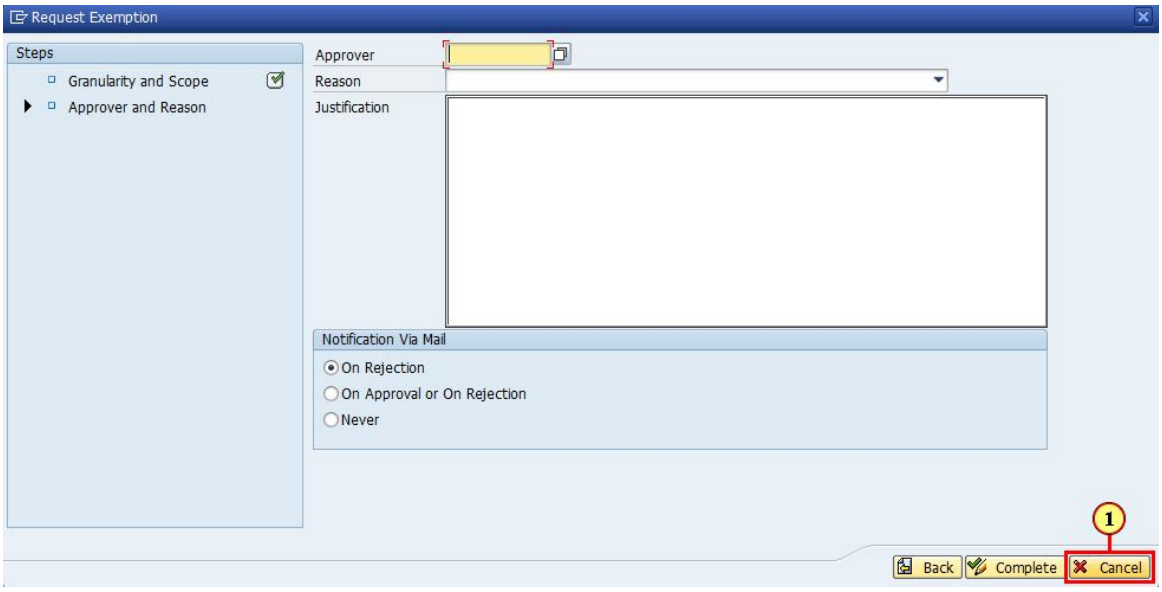
Steps
- Granularity and Scope
- Approver and Reason

Finding

Package	TEST_SEC_SQL_INJ2_01
Program	Z_SLIN_SQL_INJE_2_01
Include	Z_SLIN_SQL_INJE_2_01
Check	Security Analyses in Extended Program Check
Check Message	Potential SQL injection (SET clause)
Checksum	-184220222

☑ Restrict Exemption to Finding

When requesting an exemption, the system will first display some technical information about the exemption. Just click on Continue.



Request Exemption

Steps
- Granularity and Scope ☑
- Approver and Reason

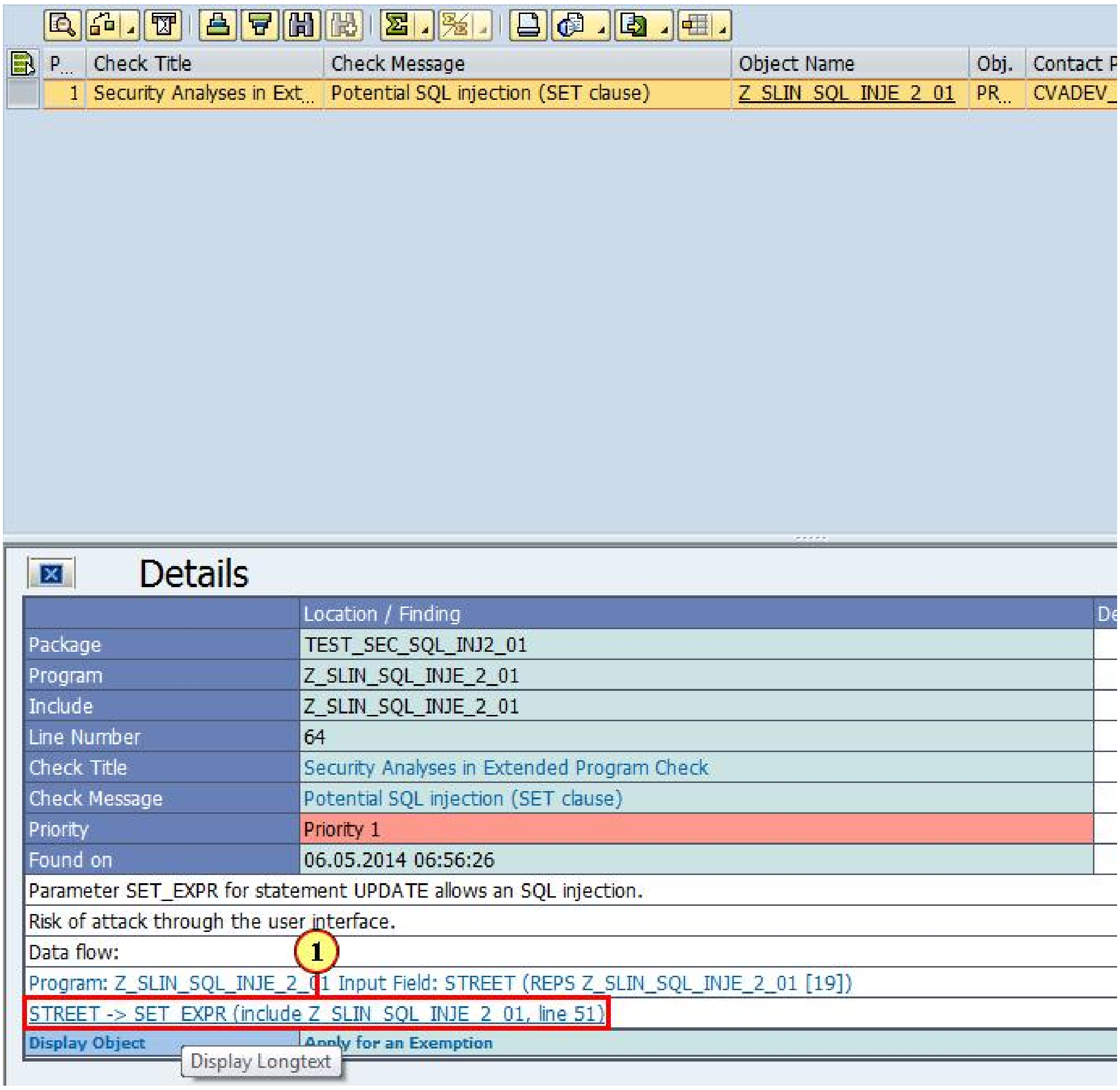
Approver	
Reason	
Justification	

Notification Via Mail
- ● On Rejection
- ○ On Approval or On Rejection
- ○ Never

When requesting an exemption, the developer needs to select an approver and a reason for the exemption. In addition, he should provide a justification for the exemption. Creating an exemption will only create a workflow. Only after the exemption has been approved, the error message will be suppressed.

Click Cancel to get back to the error message.

7.1.13. Navigating using the dataflow navigation



P...	Check Title	Check Message	Object Name	Obj.	Contact P
1	Security Analyses in Ext...	Potential SQL injection (SET clause)	Z_SLIN_SQL_INJE_2_01	PR...	CVADEV_

Details

	Location / Finding	De
Package	TEST_SEC_SQL_INJ2_01	
Program	Z_SLIN_SQL_INJE_2_01	
Include	Z_SLIN_SQL_INJE_2_01	
Line Number	64	
Check Title	Security Analyses in Extended Program Check	
Check Message	Potential SQL injection (SET clause)	
Priority	Priority 1	
Found on	06.05.2014 06:56:26	

Parameter SET_EXPR for statement UPDATE allows an SQL injection.

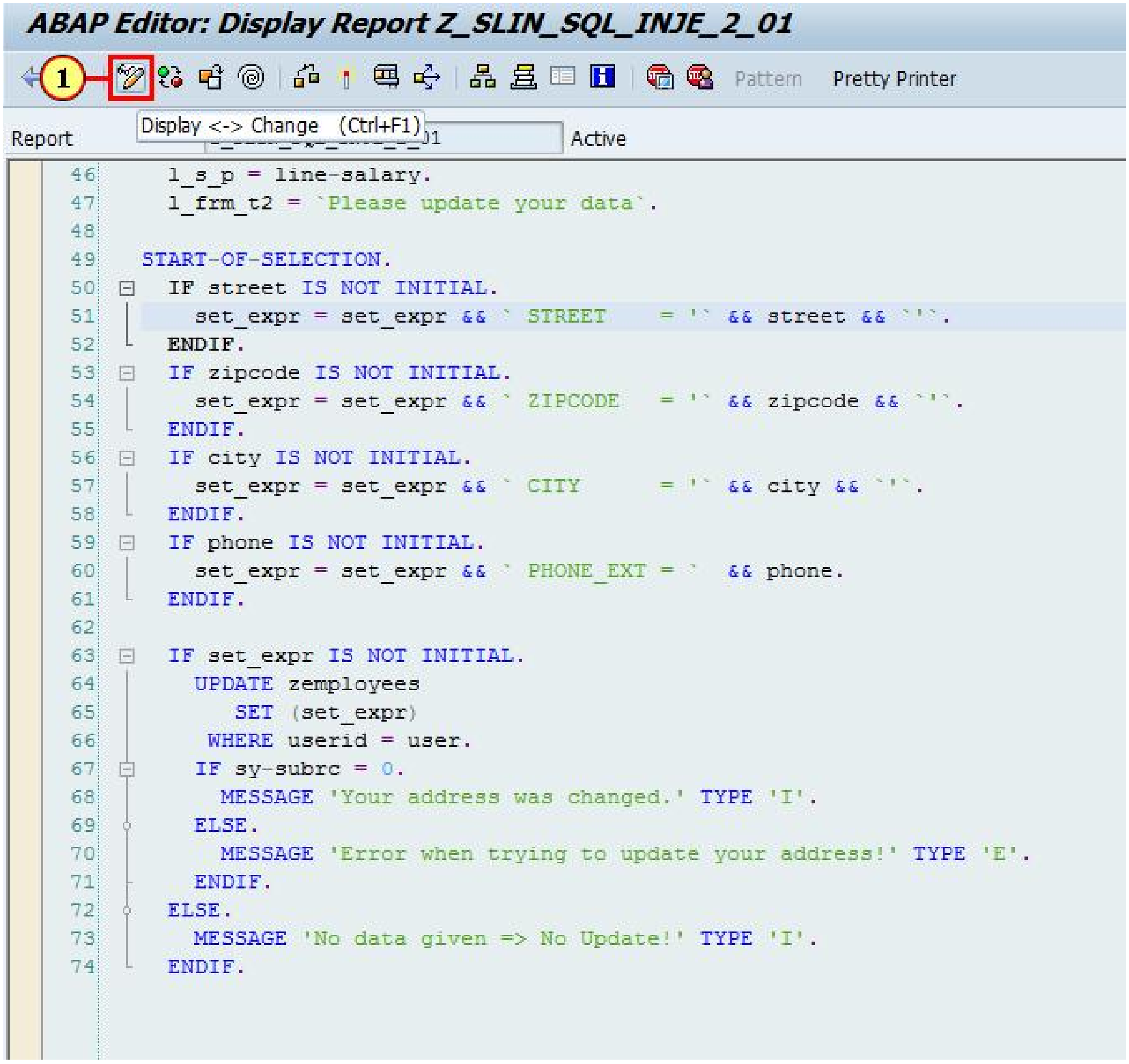
Risk of attack through the user interface.

Data flow:

Program: Z_SLIN_SQL_INJE_2_01 Input Field: STREET (REPS Z_SLIN_SQL_INJE_2_01 [19])

STREET -> SET_EXPR (include Z_SLIN_SQL_INJE_2_01, line 51)

The checks do support direct navigation to lines in the source, where the data was processed before it was used in a dangerous statement. To navigate to such a statement, click on the navigation link STREET -> SET_EXPR (include Z_SLIN_SQL_INJE_2_01, line 51).

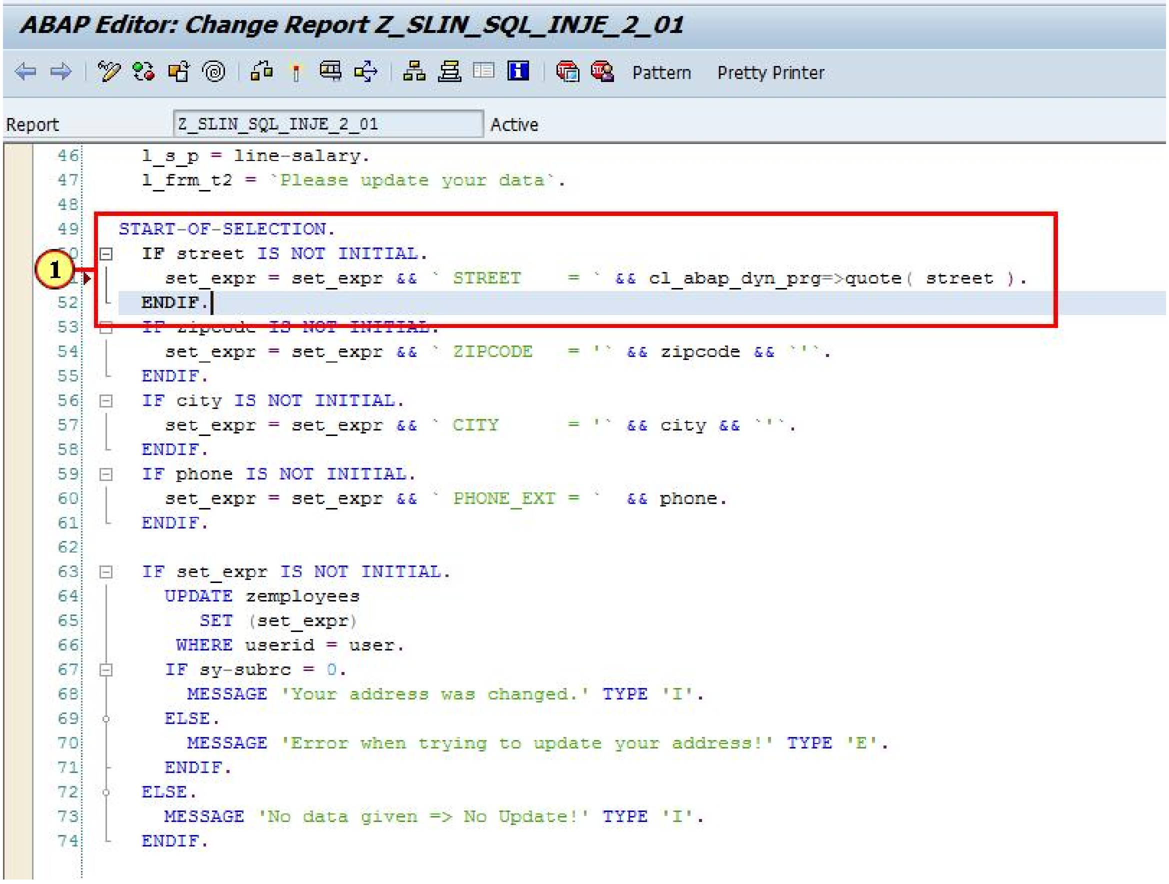

ABAP Editor: Display Report Z_SLIN_SQL_INJE_2_01
1
Pattern
Pretty Printer
Display <-> Change (Ctrl+F1)
Report
Active

```
  l_s_p = line-salary.
  l_frm_t2 = `Please update your data`.

START-OF-SELECTION.
  IF street IS NOT INITIAL.
    set_expr = set_expr && ` STREET    = '` && street && `'`.
  ENDIF.
  IF zipcode IS NOT INITIAL.
    set_expr = set_expr && ` ZIPCODE   = '` && zipcode && `'`.
  ENDIF.
  IF city IS NOT INITIAL.
    set_expr = set_expr && ` CITY      = '` && city && `'`.
  ENDIF.
  IF phone IS NOT INITIAL.
    set_expr = set_expr && ` PHONE_EXT = `  && phone.
  ENDIF.

  IF set_expr IS NOT INITIAL.
    UPDATE zemployees
       SET (set_expr)
     WHERE userid = user.
    IF sy-subrc = 0.
      MESSAGE 'Your address was changed.' TYPE 'I'.
    ELSE.
      MESSAGE 'Error when trying to update your address!' TYPE 'E'.
    ENDIF.
  ELSE.
    MESSAGE 'No data given => No Update!' TYPE 'I'.
  ENDIF.
```

The navigation will position the cursor in the line containing the dataflow.
In the case of this example, we need to change this line to protect the coding.

Click **Display <-> Change** to enable editing.

7.1.14. Fixing the code



Report Z_SLIN_SQL_INJE_2_01 Active

```
    l_s_p = line-salary.
    l_frm_t2 = `Please update your data`.

  START-OF-SELECTION.
    IF street IS NOT INITIAL.
      set_expr = set_expr && ` STREET    = ` && cl_abap_dyn_prg=>quote( street ).
    ENDIF.
    IF zipcode IS NOT INITIAL.
      set_expr = set_expr && ` ZIPCODE   = '` && zipcode && `'`.
    ENDIF.
    IF city IS NOT INITIAL.
      set_expr = set_expr && ` CITY      = '` && city && `'`.
    ENDIF.
    IF phone IS NOT INITIAL.
      set_expr = set_expr && ` PHONE_EXT = `  && phone.
    ENDIF.

    IF set_expr IS NOT INITIAL.
      UPDATE zemployees
         SET (set_expr)
       WHERE userid = user.
      IF sy-subrc = 0.
        MESSAGE 'Your address was changed.' TYPE 'I'.
      ELSE.
        MESSAGE 'Error when trying to update your address!' TYPE 'E'.
      ENDIF.
    ELSE.
      MESSAGE 'No data given => No Update!' TYPE 'I'.
    ENDIF.
```

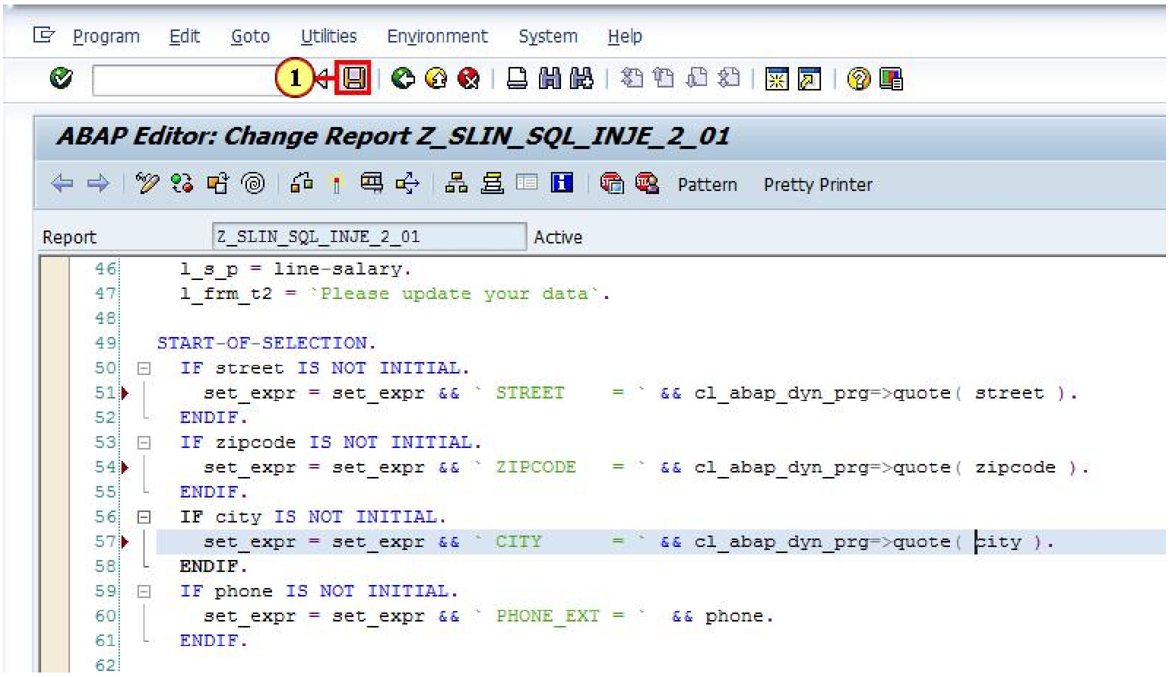
Based on the information found in the documentation, we will now replace the set expression

set_expr = set_expr && ` STREET = ' ` && street && `' `.

by

set_expr = set_expr && ` STREET = ` && cl_abap_dyn_prg=>quote (street).

in order to secure the coding. Make sure, you remove the ' after the =, as the quote method will put quotes around the string.



Report Z_SLIN_SQL_INJE_2_01 Active

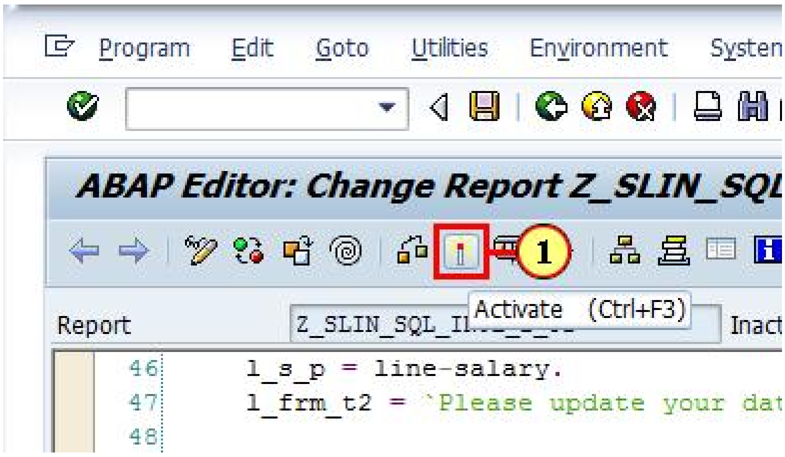
```
    l_s_p = line-salary.
    l_frm_t2 = `Please update your data`.

  START-OF-SELECTION.
    IF street IS NOT INITIAL.
      set_expr = set_expr && ` STREET    = ` && cl_abap_dyn_prg=>quote( street ).
    ENDIF.
    IF zipcode IS NOT INITIAL.
      set_expr = set_expr && ` ZIPCODE   = ` && cl_abap_dyn_prg=>quote( zipcode ).
    ENDIF.
    IF city IS NOT INITIAL.
      set_expr = set_expr && ` CITY      = ` && cl_abap_dyn_prg=>quote( city ).
    ENDIF.
    IF phone IS NOT INITIAL.
      set_expr = set_expr && ` PHONE_EXT = `  && phone.
    ENDIF.

```

(1) Do the same for the set expressions for ZIPCODE and CITY and set them to

set_expr = set_expr && ` ZIPCODE  =  ` && cl_abap_dyn_prg=>quote (zipcode).

set_expr = set_expr && ` CITY        =  ` && cl_abap_dyn_prg=>quote (city).

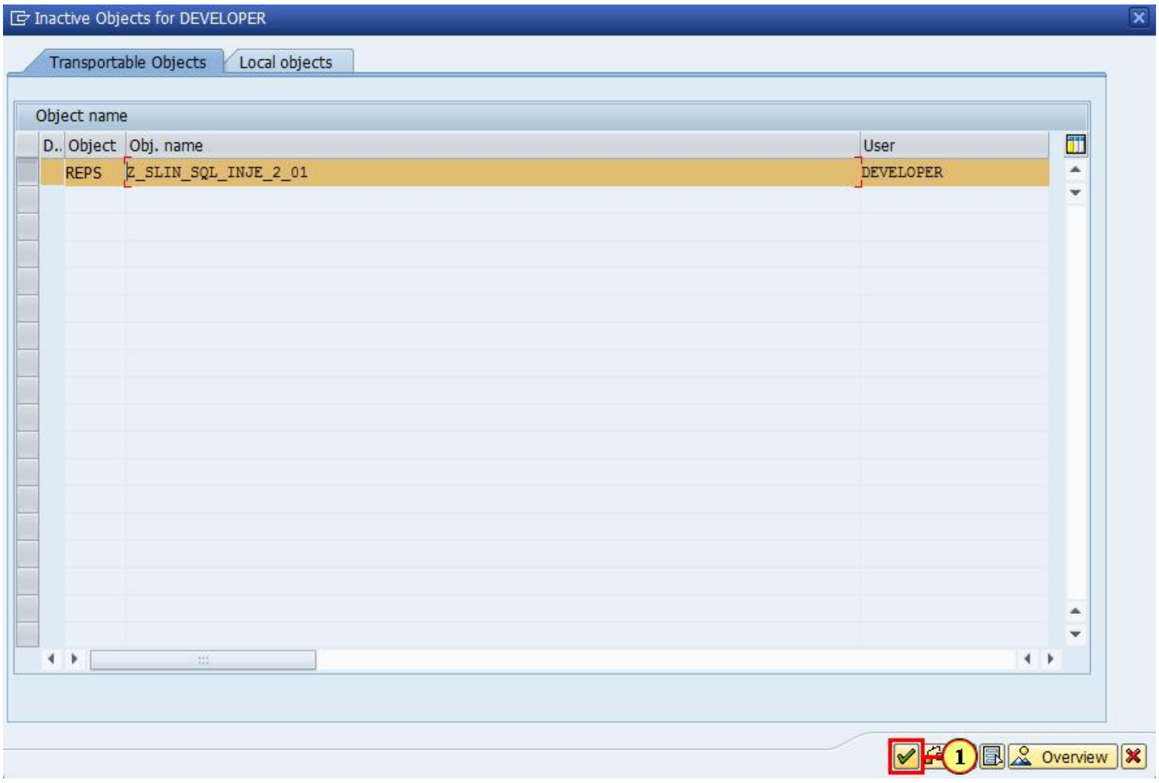
When done click to save your coding.



Report Z_SLIN_SQL_I Inact

```
      l_s_p = line-salary.
      l_frm_t2 = `Please update your dat

```

Click **Activate** to make this coding active, as the scanner will only inspect active coding.

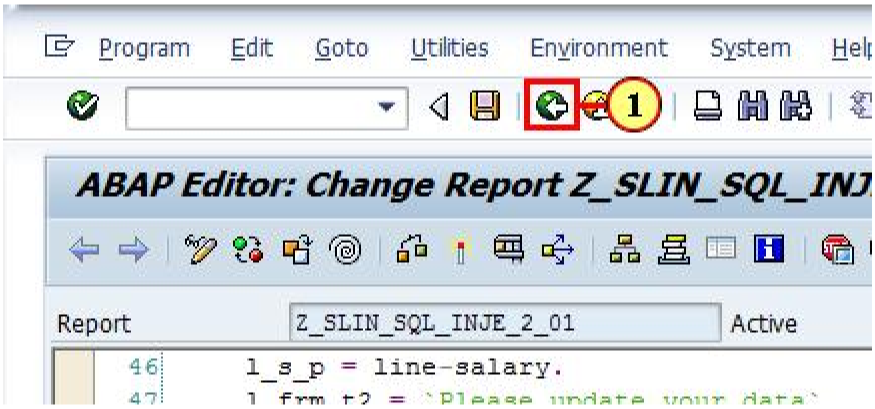


Object name

D..	Object	Obj. name	User
	REPS	Z_SLIN_SQL_INJE_2_01	DEVELOPER

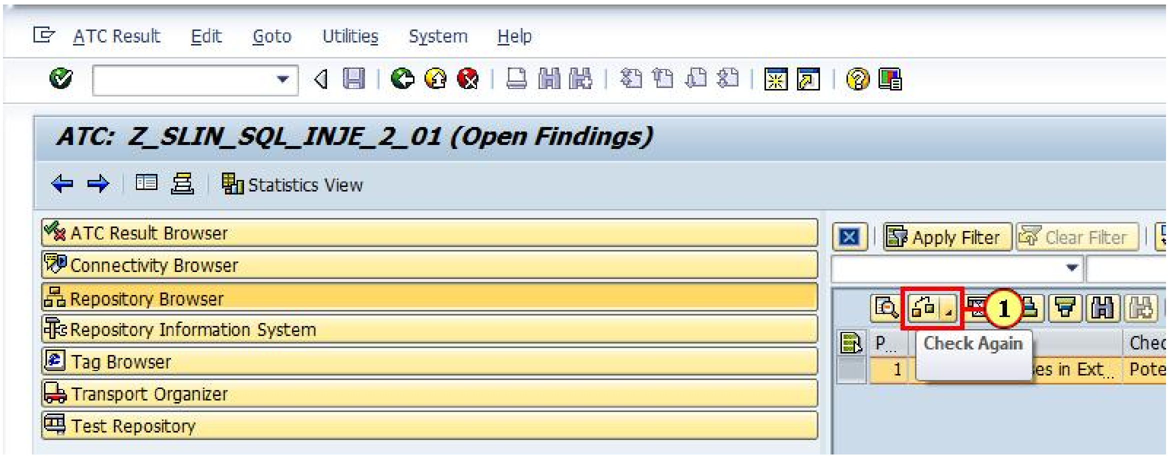
Click **Continue** ✓.

7.1.15. Re-check the program

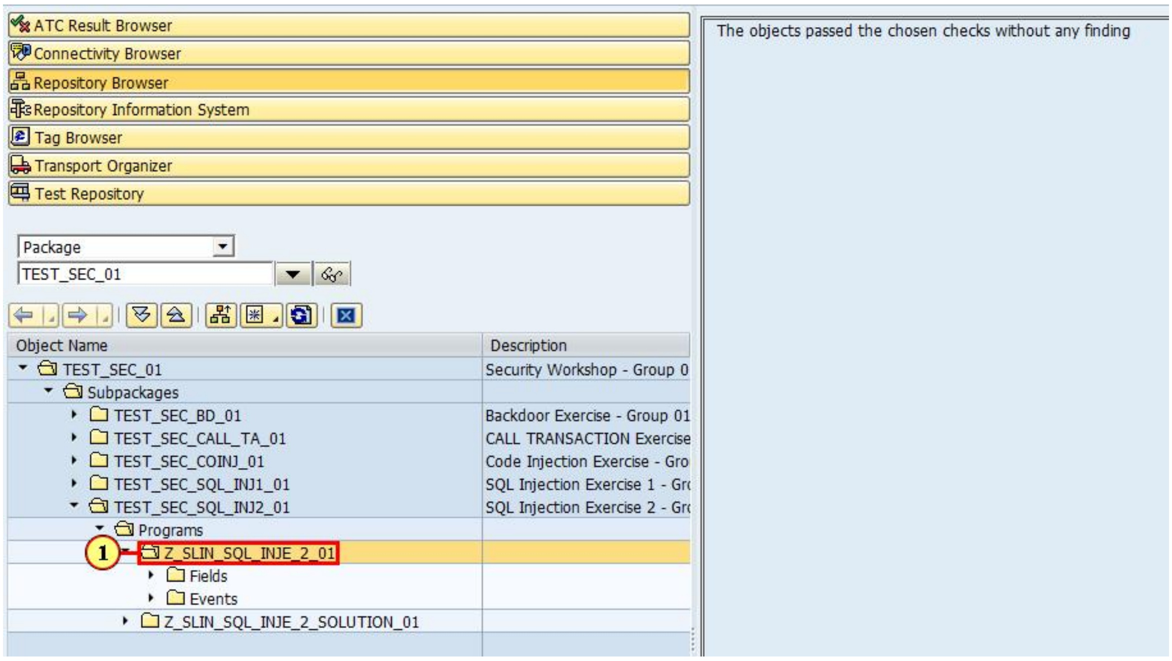


```
46        l_s_p = line-salary.
47        l_frm_t2 = `Please update your data`
```

Click to get back to the previous screen.



Click to check the coding again.

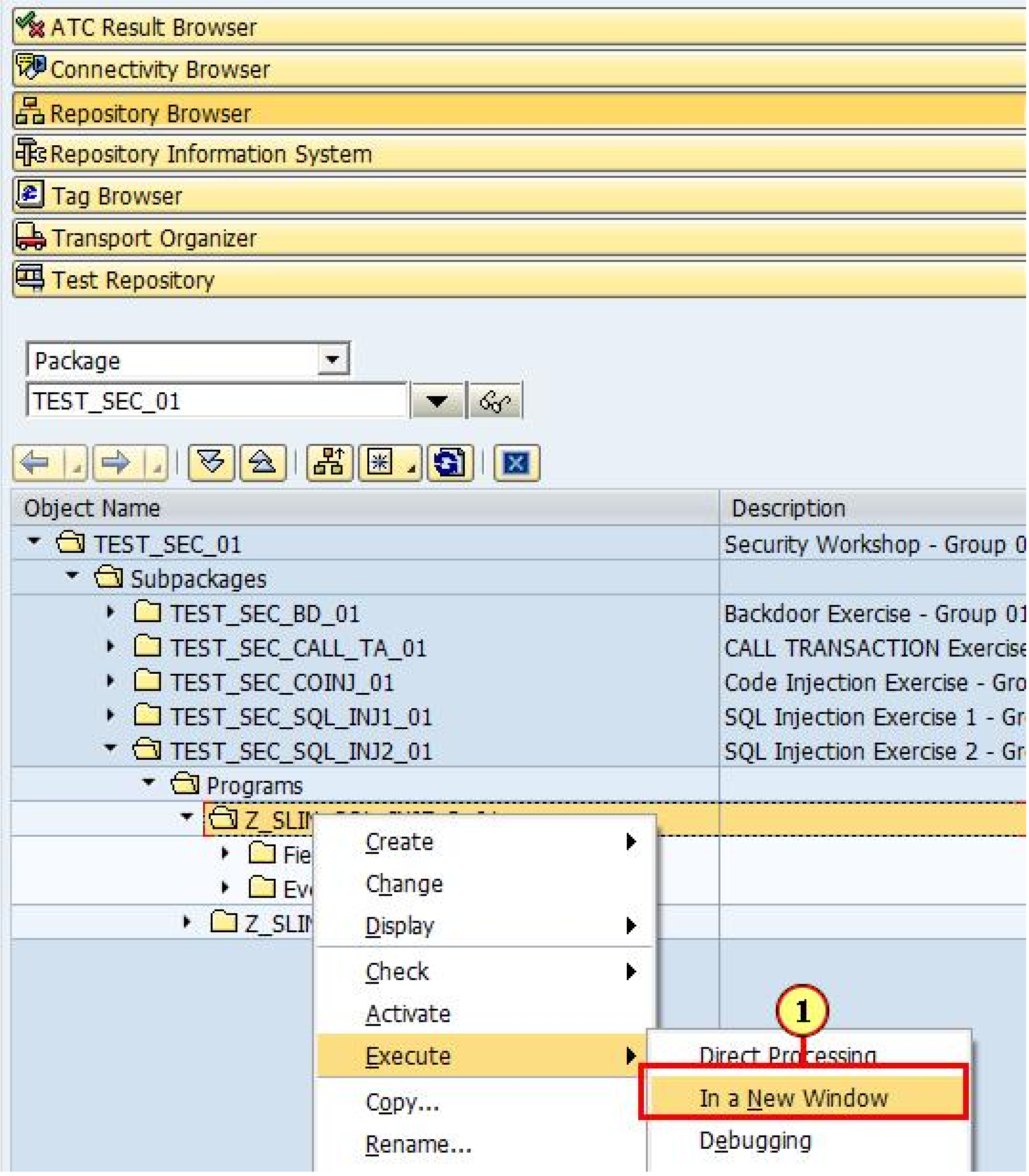


Object Name	Description
TEST_SEC_01	Security Workshop - Group 0
Subpackages	
TEST_SEC_BD_01	Backdoor Exercise - Group 01
TEST_SEC_CALL_TA_01	CALL TRANSACTION Exercise
TEST_SEC_COINJ_01	Code Injection Exercise - Gro
TEST_SEC_SQL_INJ1_01	SQL Injection Exercise 1 - Gro
TEST_SEC_SQL_INJ2_01	SQL Injection Exercise 2 - Gro
Programs	
Z_SLIN_SQL_INJE_2_01	
Fields	
Events	
Z_SLIN_SQL_INJE_2_SOLUTION_01	

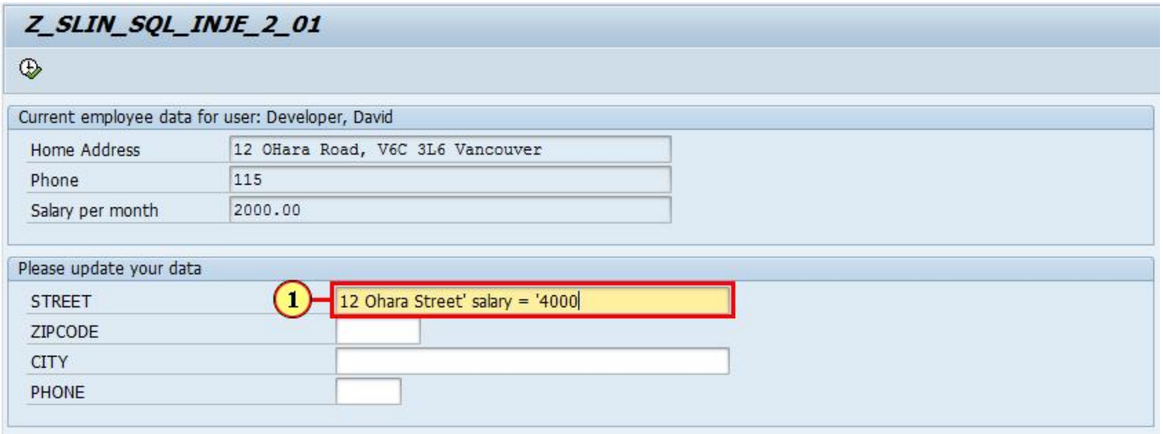
The objects passed the chosen checks without any finding

As an alternative, you can also manually check the program again, right-click on Z_SLIN_SQL_INJE_2_01 with the mouse to open the context menu and execute the ATC-checks again. The system now can't find any security issue in this program anymore.

7.1.16. Manually test the result



Click on **Execute->In a New Window** to start the report.


Z_SLIN_SQL_INJE_2_01

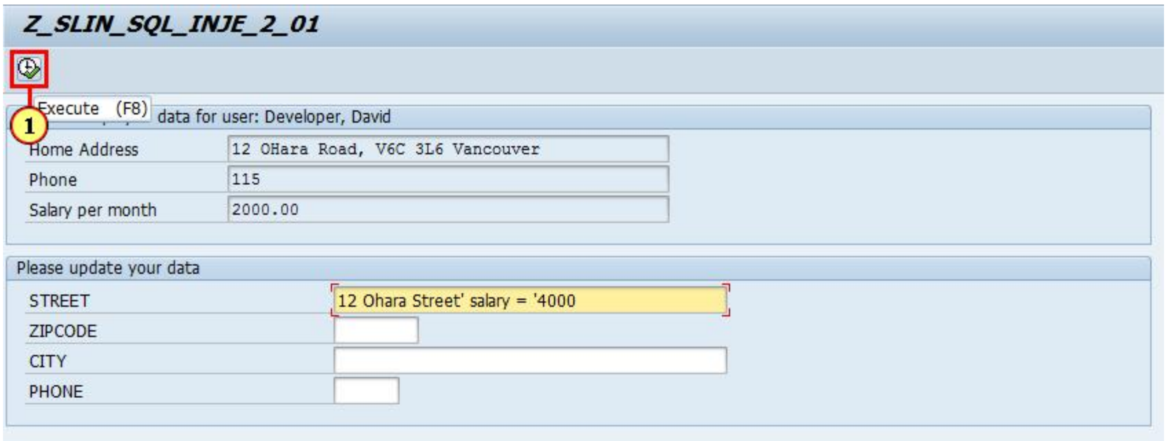
Current employee data for user: Developer, David

Home Address	12 OHara Road, V6C 3L6 Vancouver
Phone	115
Salary per month	2000.00

Please update your data

STREET	12 Ohara Street' salary = '4000
ZIPCODE	
CITY	
PHONE	

Enter the following data into the field street 12 Ohara Street' salary = '4000.


Z_SLIN_SQL_INJE_2_01

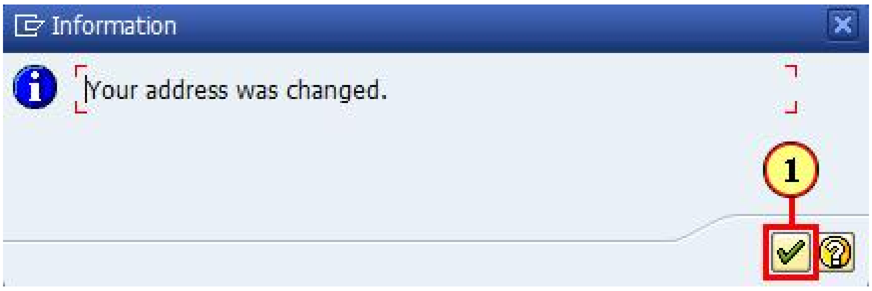
Current employee data for user: Developer, David

Home Address	12 OHara Road, V6C 3L6 Vancouver
Phone	115
Salary per month	2000.00

Please update your data

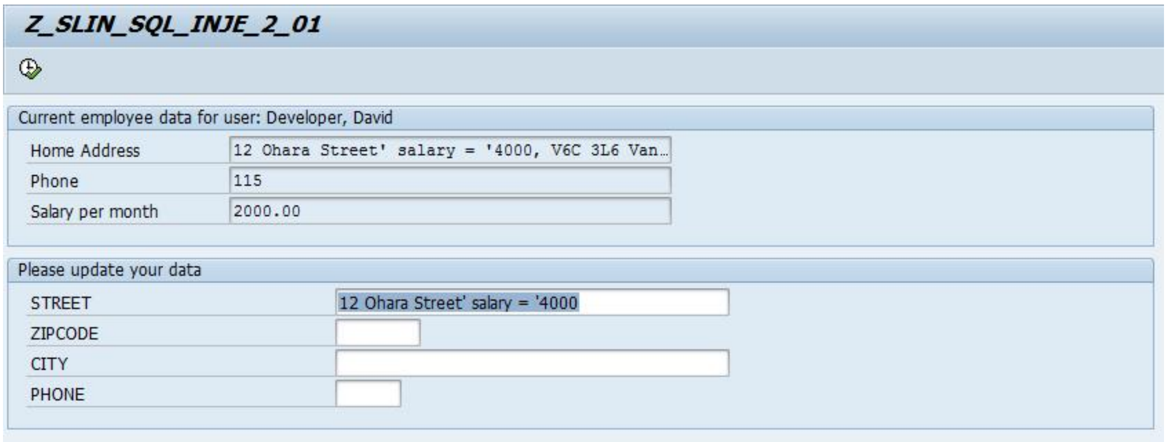
STREET	12 Ohara Street' salary = '4000
ZIPCODE	
CITY	
PHONE	

Click **Execute**.


Information

Your address was changed.

Click **Continue** ✓.



Z_SLIN_SQL_INJE_2_01

Current employee data for user: Developer, David

Home Address	12 Ohara Street' salary = '4000, V6C 3L6 Van…
Phone	115
Salary per month	2000.00

Please update your data

STREET	12 Ohara Street' salary = '4000
ZIPCODE	
CITY	
PHONE	

As you can see, the information this time ended up all in the right place. The security risk was found, fixed and now is gone.

8. Security

8.1. System Parameter Summary

This section gives an overview of all relevant system parameters.

Parameter Description	Parameter Value	Note
Master Password	*<Master Password>*	You specify the master password during instance creation in the SAP Cloud Appliance Library.
Server domain	dummy.nodomain	If no specific domain has been maintained yet.
Private key	*<Private Key File>*	The private key file is provided during instance creation in the SAP Cloud Appliance Library. It is used for SSH access to the host.
Server IP Address	*<IP Address>*	The IP address of your instance from the SAP Cloud Appliance Library
Host names	abapci, hanadb	Predefined name of the host on which HANA and ABAP are running.
Root user / password	root / *<Private Key File>*	Default OS Administrator user for Linux SUSE.
HDB System ID	HDB	-
HDB Instance Number	02	-
DB User / Password	SYSTEM / *<Master Password>* SAPHANAABAP / *<Master Password>* DBACOCKPIT / *<Master Password>*	-
HDB Administrator OS level / Password	hdbadm / *<Master Password>*	-
ABAP System ID	A4H	-
ABAP Instance number	00	-
ABAP Client/ User / Password	000 / SAP* / *<Master Password>* 000 / DDIC / *<Master Password>* 001 / SAP* / *<Master Password>* 001 / DDIC / *<Master Password>* 001 / DEVELOPER / *<Master Password>*	-
ABAP Administrator OS level / Password	a4hadm / *<Master Password>*	-
SAP System Administrator	sapadm / *<Master Password>*	-

Diagnostic Agent User	daaadm / *<Master Password>*	-

8.2. Security Recommendations

This section provides an overview of the security-relevant information.

To mitigate the potential security risks (for example, OS users can obtain the password of the solution while the initial provisioning is in process) we recommend changing the password of the following users:

- **SYSTEM** – this is a DB user.
  The procedure can be executed from SAP HANA Studio:
  1. Start the installed SAP HANA Studio.
  2. From the *Systems* view right click with the mouse on HANA instance SID (user SYSTEM).
  3. Select *SQL editor.*
  4. In the editor, enter the following string:
     - `ALTER USER system PASSWORD` **`<new_password>`**.
  5. Choose *Execute*.
  6. Switch on OS to a4hadm user and execute the following command:
     "`/usr/sap/A4H/hdbclient/hdbuserstore set default hanadb:30215 SYSTEM` **`<new_password>`**"
  7. Change password also in properties of the HDB system in the HANA Studio. From the *Systems* view right click with the mouse on HANA instance SID. Select Properties → Database User Logon and change the password.

- **SAPHANAABAP** – this is a DB user.
  As the user is used for the ABAP server connection to the database you should stop the ABAP system during the password change procedure:
  1. As the user is used for the ABAP server connection to the database you should stop the ABAP system during the password change procedure. On operating system level execute:

     ```
     su – a4hadm
     stopsap r3
     exit
     ```
  2. Start the installed SAP HANA Studio.
  3. Open the *Systems* view and right click with the mouse in this view and then choose *Add System…*
  4. In the *System* wizard, specify the following parameters:

Parameter ID	Parameter Value	Note
Hostname	*<IP Address>*	The IP address of the instance from the SAP Cloud Appliance Library
Instance Number	02	HANA instance number used for the appliance.
User Name	SAPHANAABAP	User used for ABAP DB connections
Password	*<Master Password>*	The password is the same as the master password provided during instance creation in the SAP Cloud Appliance Library.

  5. From the *Navigator* view right click with the mouse on HANA instance SID (User SAPHANAABAP).
  6. Select *SQL editor.*
  7. In the editor, enter the following string:
     - `ALTER USER SAPHANAABAP PASSWORD` **`<new_password>`**.

8. Choose *Execute*.
9. Switch on OS to a4hadm user and execute the following command: "`/usr/sap/A4H/hdbclient/hdbuserstore set default hanadb:30215 SAPHANAABAP` **`<new_password>`**"
10. Change password also in Properties of the HDB system in the HANA Studio. From the *Navigator* view right click with the mouse on HANA instance SID. Select Properties → Database User Logon and change the password.
11. Restart the ABAP system: On operating system level execute:

```
su – a4hadm
startsap r3
exit
```

- **hdbadm** – this is an OS user.

  To change the password you have to logon with the root user to the Linux OS and change the password of the *hdbadm* user. For more information, see Connecting to Your Backend on OS Level.

  In the Linux console you have to execute the following command: `passwd hdbadm` and then enter the new password.
- **a4hadm** – this is an OS user.

  To change the password you have to logon with the root user to the Linux OS and change the password of the *a4hadm* user. For more information, see Connecting to Your Backend on OS Level.

  In the Linux console you have to execute the following command: `passwd a4hadm` and then enter the new password.
- **SAP*, DDIC, DEVELOPER, BWDEVELOPER** – these are ABAP users.

  The default password for all three users is the <*`Master Password`*> provided during the instance creation via CAL. To change the password insert user and password in the SAP GUI login screen and press the button *new password*.

9. Additional Information

9.1. Frontend Settings

9.1.1. Windows Proxy Settings

If you want to access the internet from your frontend instance, which is not required for working with this solution but allows you to update your frontend tools, please proceed as follows:

a) If you created your instance in the **public** AWS cloud environment, no additional proxy settings should be required to access the internet.

b) If you created your instance in a **private/corporate subnet** of a virtual private cloud (VPC), you have to enter the address of your proxy server or the internet gateway in the Windows internet settings: Open Internet Explorer > Internet options and add a proxy exception for *hanadb, abapci, *.dummy.nodomain* (default dummy domain) or the fully qualified host names. SAP HANA Studio will automatically update it's proxy settings according to the Windows internet settings if you don't change the default proxy settings in Eclipse (*Active Provider: Native*).

9.1.2. Enhanced Security Configuration

On the Windows Server image the enhanced security configuration (ESC) of Internet Explorer is activated by default (recommended by Microsoft). This makes working with Web UIs and external sites like SCN inconvenient and forces you to deal with several security pop-ups and notifications. Thus, you can deactivate IE ESC with the following procedure (at your own risk):

1. In the Start menu navigate to All Programs > Administrative Tools > Server Manager.

2. In the *Server Manager* root node click on *Configure IE ESC* in the Security Information section.
3. Deactivate the *IE ESC* for administrators.

9.2. Using Local Client Software

If you want to access SAP HANA or the ABAP application server of your backend instance from your local client (not via the associated frontend instance), we recommend the following procedure for a local Windows environment (other operating systems require different SAP clients):

Downloading the SAP client software

1. Before connecting to the frontend instance using the *Remote Desktop Connection*, open the *Options* dialog and navigate to the *Local Resources* tab.
2. Hit the more button in the *Local devices and resources* section and activate one of your local drives.
3. After logging into your frontend instance you can use *Windows Explorer* to exchange files between your local machine and your frontend instance.
4. The SAP software is located on the D: drive of your Windows instance:
   - Copy the *hdbstudio70* folder to your local environment and adapt the JVM location in the *hdbstudio.ini* file to the location of your local JVM.
   - Copy the SAPGUI-BI-Core.exe file to your local environment and start the installation of SAP GUI for Windows (take a look at the SAP GUI documentation for prerequisites).

Accessing your Backend Instance locally

Before you can use your local SAP client software you have to ensure, that your backend instance is accessible locally and all required TCP ports are open.

> ❗ We strongly recommend not to use a public instance for this setup, but instances running in a virtual private cloud (VPC) with a VPN connection to your local network. For more information, how to create instances in a VPC please revisit the Using SAP Cloud Appliance Library section.

If your instances are running in a VPC with VPN connection to your local network, you could open all ports by checking the *Open all TCP ports* option in the *Access Points* setting (Virtual Machine tab of your CAL instance). If you only want to expose the required ports (recommended approach), the list below shows all required TCP ports:

Protocol	Port	Description
SSH	22	Used for SSH connection to the server
Custom TCP	3200	SAP Dispatcher, used by SAP GUI
Custom TCP	3300	SAP Gateway. Used for CPIC and RFC communication.
Custom TCP	3601	Message Server
HTTP	8002	HTTP (HANA XS)
Custom TCP	30215	External SQL Interface. Used by SAP HANA Studio.
HTTP	50000	HTTP (AS ABAP)
HTTPS	50001	HTTPS (AS ABAP)
Custom TCP	50213	Instance agent. SAP Start administrative channel for low-level access to the SAP HANA instance to allow features such as starting or stopping of the SAP HANA database.
Custom TCP	50214	Instance agent (SSL).

After following the steps above you should be able to access your cloud instances from your local environment and use the locally installed client software.

> ✎ Please keep in mind that in case of latency or bandwidth issues a remote desktop connection to your frontend instance might be the better choice.

10. Troubleshooting

- **Symptom: You cannot connect to your frontend instance via Remote Desktop Connection.**
  Please ensure that your local network permits outbound RDP connections on port 3389 (TCP/UDP), i.e. your firewall/router doesn't block these connections.

- **Symptom: You cannot log on to the frontend instance via Remote Desktop.**
  The frontend user account name is 'Administrator'. The password used is the password you initially did specify as master password. Please allow the cloud infrastructure to set up the accounts on the system. This may take up to 30 minutes after the initial deployment. During this time, authentication may fail for the administrator account.

- **Symptom: You cannot connect to your backend instance via SSH.**
  Please ensure that your local network permits outbound SSH connections on port 22, i.e. your firewall/router doesn't block these connections.

- **Symptom: You cannot select the *Corporate Network* option when creating a new solution instance.**
  The AWS subnet you want to use might be located in the wrong region. Please ensure to create a subnet in the Amazon region US-East (Virginia).

- **Symptom: You can't connect to your backend instance using your local SAP GUI.**

  - Check, if the ABAP server is running:

  Logon as `root` to the server on which the database is running (see Connecting to Your Backend on OS Level). Then execute the following statements to check the status of the ABAP server:

  ```
  su – a4hadm
  sapcontrol -nr 00 -function GetProcessList
  ```

  - Check if all required TCP ports are open and accessible from your local network:
  Please refer to section Accessing your Backend Instance locally for more information.

For more information about how to use the SAP Cloud Appliance Library, you can read the official documentation of the product by choosing the following navigation from SAP Cloud Appliance Library: *Related Links & Help* → *Documentation*. If you cannot find the needed information in the documentation, you can open a normal support ticket within the SAP Cloud Appliance Library (BC-VCM-CAL) component and your ticket will be processed by the SAP Cloud Appliance Library Operators. If you have AWS related problems, you can report them directly to AWS support or alternatively on the BC-OP-LNX-AWS component in SAP Service Marketplace.