IBM Safer Payments Hands-On Technical Primer Training (V6.3)

Course#:6A302G
Duration:32 Hours
Price:3900.00

Course Description

IBM Safer Payments is an innovative real-time payment fraud prevention and detection solution for all cashless payment types.

In this course, technical resources that will be involved in any aspect of deploying, customizing, and implementing the IBM Safer Payment solution will get a hands-on overview of the product. Students will get an understanding of how to install, configure, and setup the system for Analysts as well as basics on system administration.

Objectives

Introduction to Safer Payments Architecture
Install Safer Payments and create service scripts
Introduction to system configuration
Examine mandator concept
Introduction to user accounts and user roles
Examine data import
Introduction to modelling concept
Create a model for processing authorization transactions
Simulate and test rules
Invoke python callouts
Export and import PMML models, and enable sampling techniques
Introduction to Safer Payments rule generator
Introduction to compliance list
Introduction to reports in Safer Payments
Introduction to Safer Payments case management

Audience

IBM Safer Payments users (System Administrators, Fraud Analysts, Fraud Investigators), IBM Lab experts, and IBM Business Partners.

Prerequisites

- Must be familiar with Unix command line navigation and configuration actions
- Some familiarity with payment processing systems
- Familiarity with the concepts of clustering and network/firewall topics
- Familiarity and comfort with troubleshooting and problem determination processes

Content

Introduction to Safer Payments Architecture

- Examine cluster concept
- Safer Payments Interfaces
- Failover concept

Install Safer Payments and create service scripts

Install and set up a cluster of 3 Safer Payments instances

Log in to Safer Payments user interface

Start instances interactively

Create service scripts to start Safer Payments instances

Introduction to system configuration

Examine Safer Payments various system configuration settings

Set or modify settings

Examine mandator concept

Examine message computation flow

Update existing mandator settings

Add new sub mandators into the hierarchy

Introduction to user accounts and user roles

Define new roles

Examine user account settings

Create new user accounts and assign a role

Examine data import

Create Messages, Mappings, and Job scheduler to prepare import of data
Configure and import masterdata as well as transaction data
Send messages via online real-time interface

Introduction to modelling concept

Understand challenger and champion concept
Examine model components
Understand modelling workflow
Review indexes and sequences

Create a model for processing authorization transactions

Create some fraud prevention rules manually
Step through the various elements to create and promote new rules
Examine profiling techniques
Create Rules using profiling attributes
Run investigation queries

Simulate and test rules

Configure data set for simulation
Configure Analyses
Run a simulation

Examine Rule statistics

Invoke python callouts

Load Python scripts
Invoke python function in preprocessing rules
Use python function in rules
Test function in simulation environment

Export and import PMML models, and enable sampling techniques

Export data of Safer Payments
Import PMML files to ingest externally created models within Safer Payments
Start simulation for a PMML model analysis
Examine analyses results
Activate and use sampling techniques

Introduction to Safer Payments rule generator

Examine semi-automatic rule generation with Safer Payments
Understand how rule generator is slicing data to find rules

Introduction to compliance list

Examine difference between real-time and adhoc checks

Create a blacklist and add entries

Test the created blacklist

Introduction to reports in Safer Payments

Examine Transaction and Investigation Reports

Create various Reports

Introduction to Safer Payments case management

Examine Case Investigation workflows and working queues

Create case close codes and case states

Define a case workflow

Create a case and push it to the next case state in the workflow