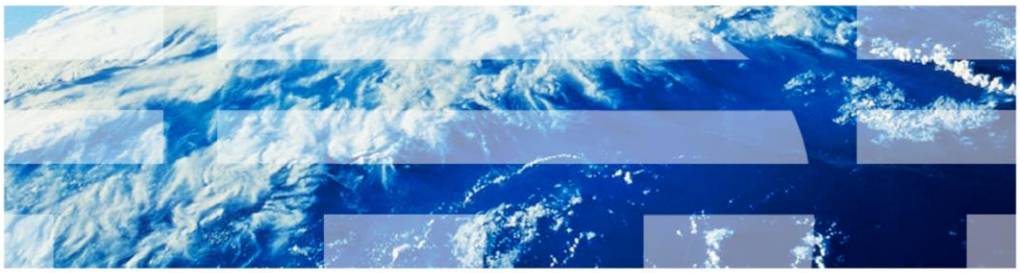
IBM

OMEGAMON XE for CICS on z/OS V4.1.0, V4.2.0, V5.1.0

OMEGAMON CICS menu system security



OMEGAMON® XE for CICS® on z/OS® for V4.1.0, V4.2.0, and V5.1.0: OMEGAMON CICS menu system security. This module describes the different ways security can be implemented in the OMEGAMON CICS Classic or menu system.

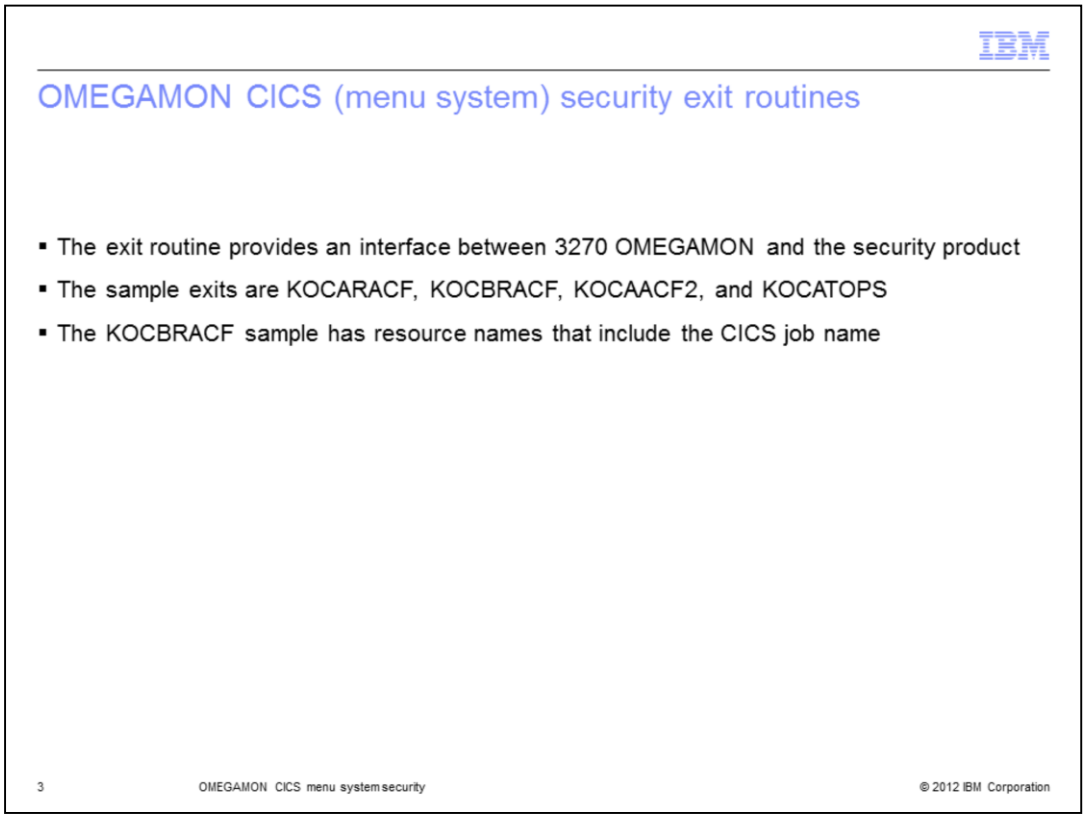
When you have completed this module, you can perform these tasks:

- Describe 3270 OMEGAMON (menu system) security exit routines use

- Describe external security for log on and internal security for commands

- Describe external security for log on and external security for commands

- Describe external security for both log on and a mixture of internal security and external security for commands



The security exit routine provides an interface between 3270 OMEGAMON and the security product. You can specify any unique name for your routine, but that name must also be specified in the control statements that update the security table. The exit routine can be shared between systems. KOCARACF, KOCBRACF, KOCAACF2, and KOCATOPS are members of the TKANSAM data set that contain models of RACF®, CA-ACF2, and CA-TOP SECRET routines.

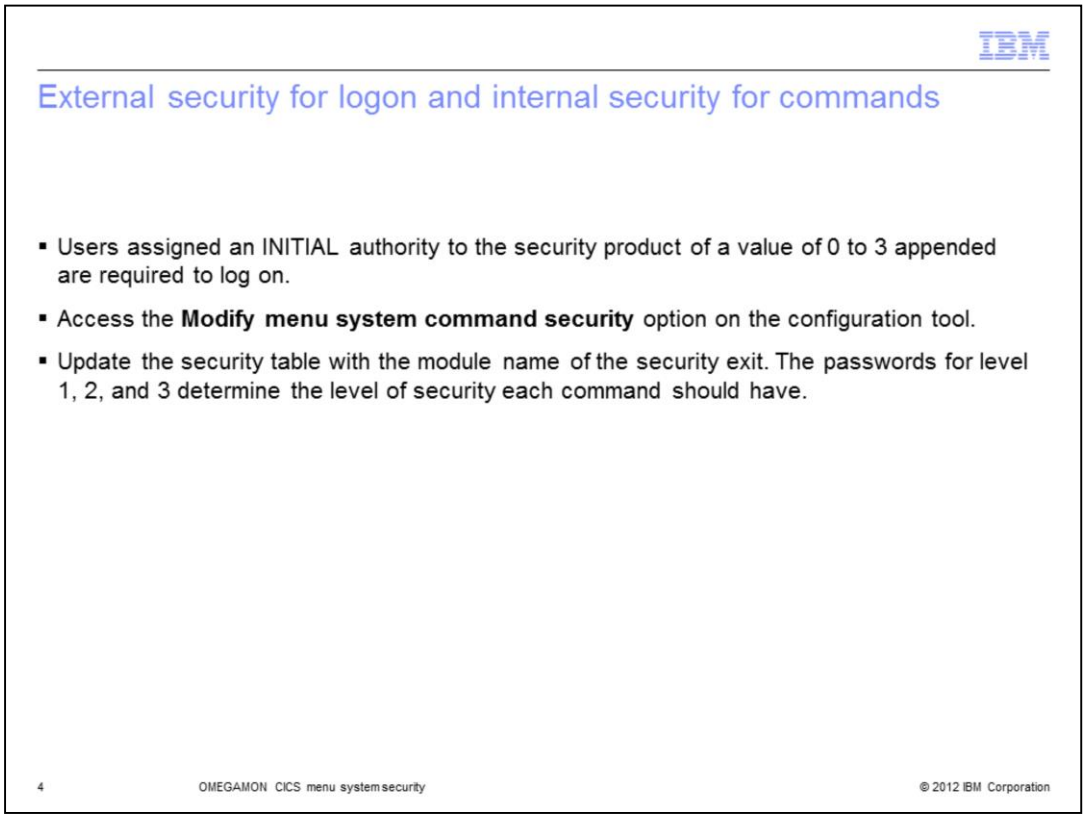
They are supplied as examples only. Verify that the resource class you define in the exit routine has the same name as the resource class you defined when modifying RACF, CA-ACF2, or CA-TOP SECRET rules. The TKANSAM data set contains members, KOCJACF2, KOCJRACF, and KOCJTOPS, which supply sample JCL to help you assemble and link-edit your routine.

The KOCBRACF sample uses resource names that include the CICS job name.

This example generates a resource name of CICSNAME:

- CICSTEST.INITIAL3

- CICSPROD.CMT



Users assigned **INITIAL** authority are required to logon and are allowed to change their internal security level by using the **/PWD** command. This **RDEFINE** is an example that permits the **/PWD** command to work.

RDEFINE class name INITIAL UACC(READ)

Access the Configuration Tool, option 2, Configure Tivoli® OMEGAMON II® for CICS panel. Access the **Modify menu system command security** option. The needed control statements are presented to you.

Specify the security module name in the statement.

MODULE= …

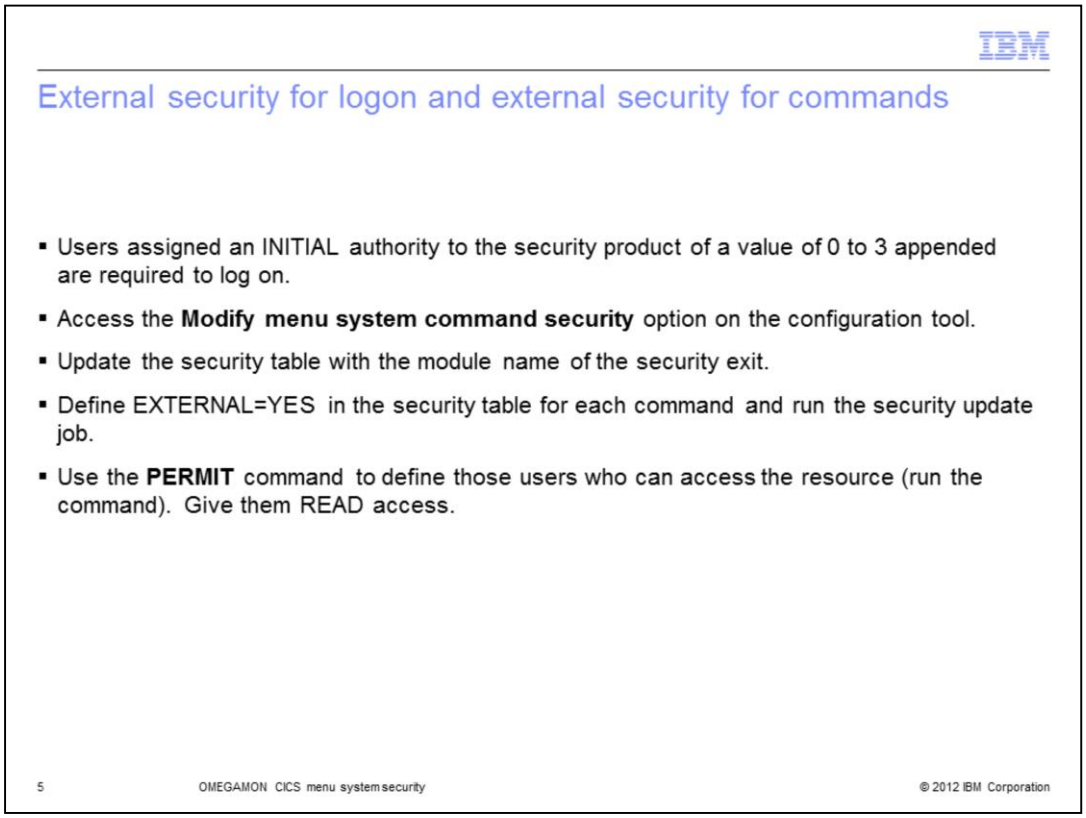
Edit the control statements to indicate the three level passwords.

PASSWORD=CANDLE1,LEVEL=1 …

PASSWORD=CANDLE2,LEVEL=2 …

PASSWORD=CANDLE3,LEVEL=3 …

All 3270 OMEGAMON commands major, minor, immediate, and INFO-line have a security level of 0, 1, 2, or 3. Level **3** provides the highest degree of protection. A setting of **0** means that any user can access the command. Specify the security level for each command. When you have edited the control statements and pressed **F3**, you are presented with the JCL that starts the KOBSUPDT program, which updates the OMEGAMON for CICS security table.



Users assigned **INITIAL** authority are required to logon and are allowed to change their internal security level by using the **/PWD** command. This command is an example that uses **RDEFINE** to allow the **/PWD** command to work.

RDEFINE class name INITIAL UACC(READ) …

Access the Configuration Tool. Option **2**, Configure Tivoli OMEGAMON II for CICS panel. Access the **Modify menu system command security** option. The needed control statements display. Specify the security module name in the statement.

MODULE= …

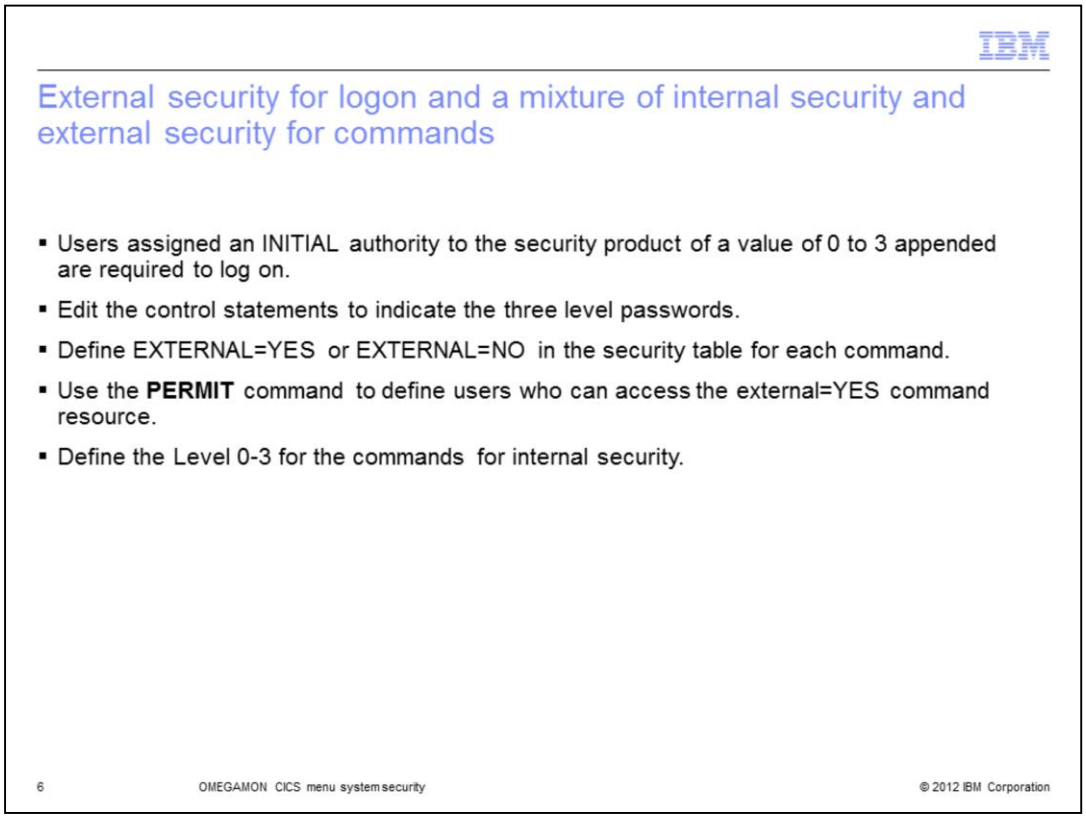
Define EXTERNAL=YES in the security table for each command and run the security update job.

Run the **PERMIT** command to define those users who can access the resource.

Give them **READ** access. This example shows how to authorize a user to run the **KILL** command with RACF:

RDEFINE class name KILL UACC(NONE) …

PERMIT KILL CLASS (class name) ID(USER01) ACCESS(READ)



Users assigned **INITIAL** authority are required to log on and allowed to change their internal security level by using the **/PWD** command. This **RDEFINE** example permits the **/PWD** command to work.

RDEFINE class name INITIAL UACC(READ) …

Access the Configuration Tool. Option **2**, Configure Tivoli OMEGAMON II for CICS panel. Access the **Modify menu system command security** option. The needed control statements display. Specify the security module name in the statement.

MODULE= …

Edit the control statements to indicate the three level passwords.

PASSWORD=CANDLE1,LEVEL=1 …

PASSWORD=CANDLE2,LEVEL=2 …

PASSWORD=CANDLE3,LEVEL=3 …

Define EXTERNAL=YES or EXTERNAL=NO in the security table for each command.

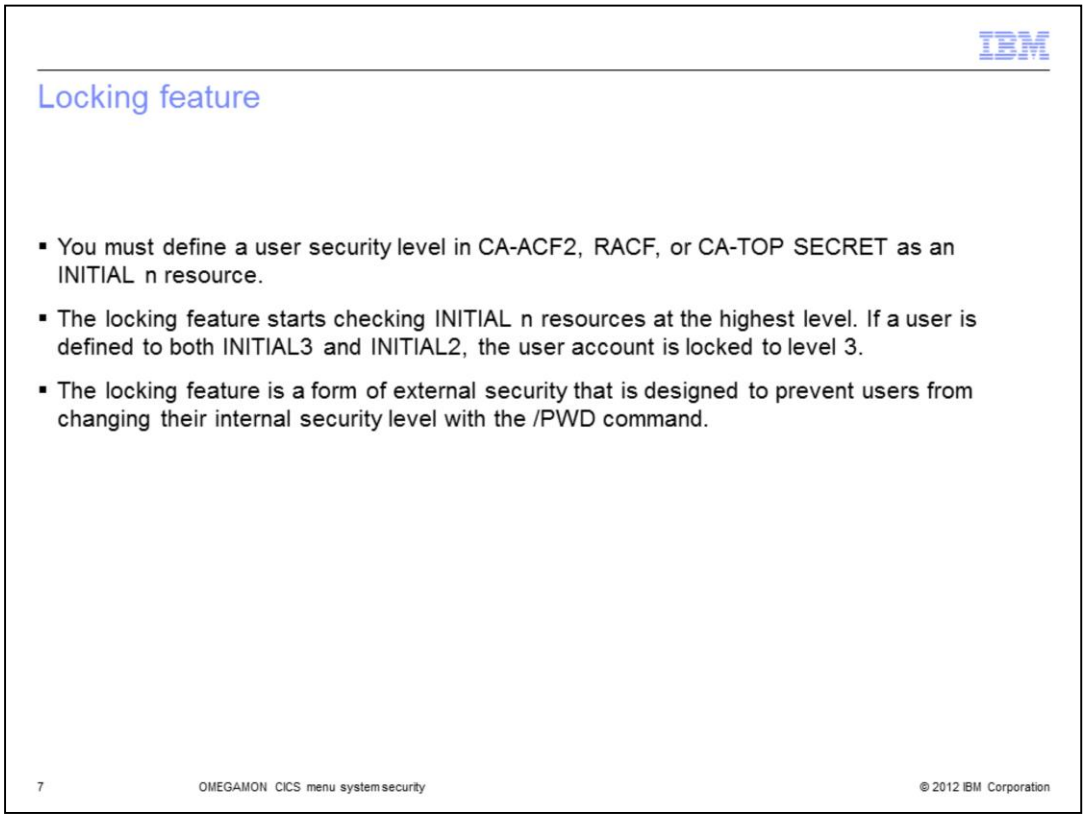
Use the PERMIT command to define those users who can access the **EXTERNAL=YES** command.

Example:

RDEFINE class name KILL UACC(NONE) …

PERMIT KILL CLASS (class name) ID(USER01) ACCESS(READ) …

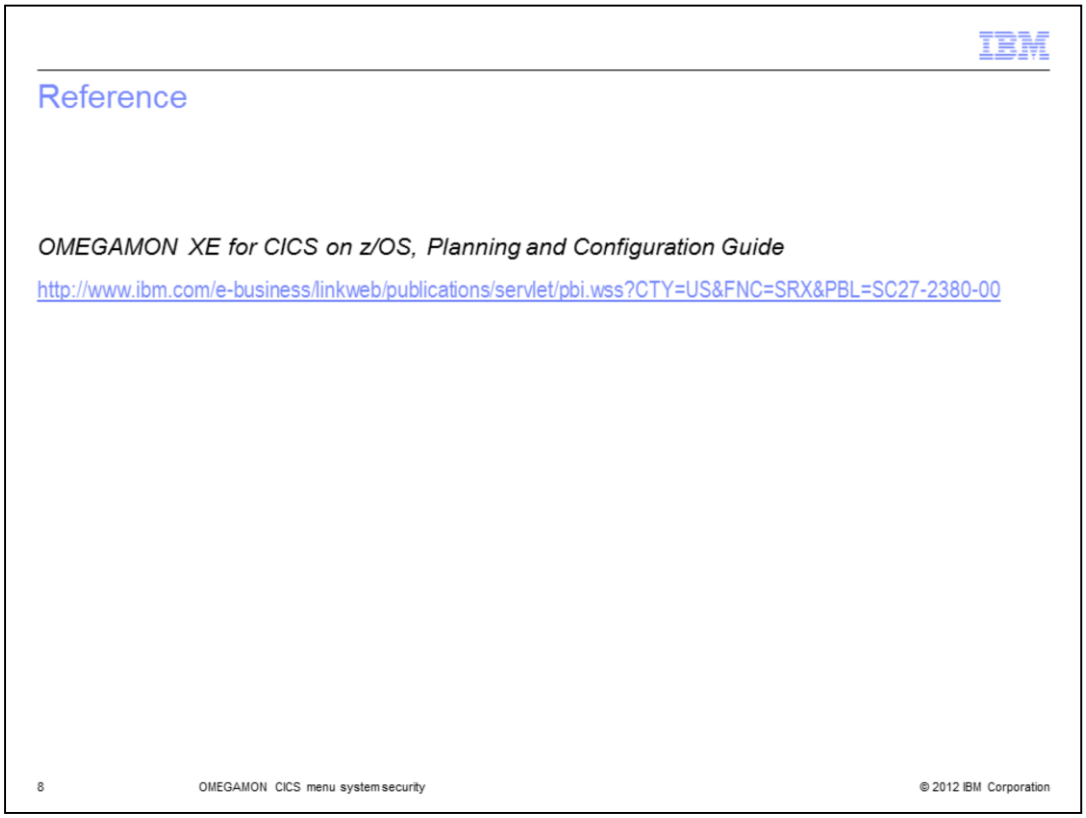
Define the Level **0** to **3** for the commands for internal security **…**



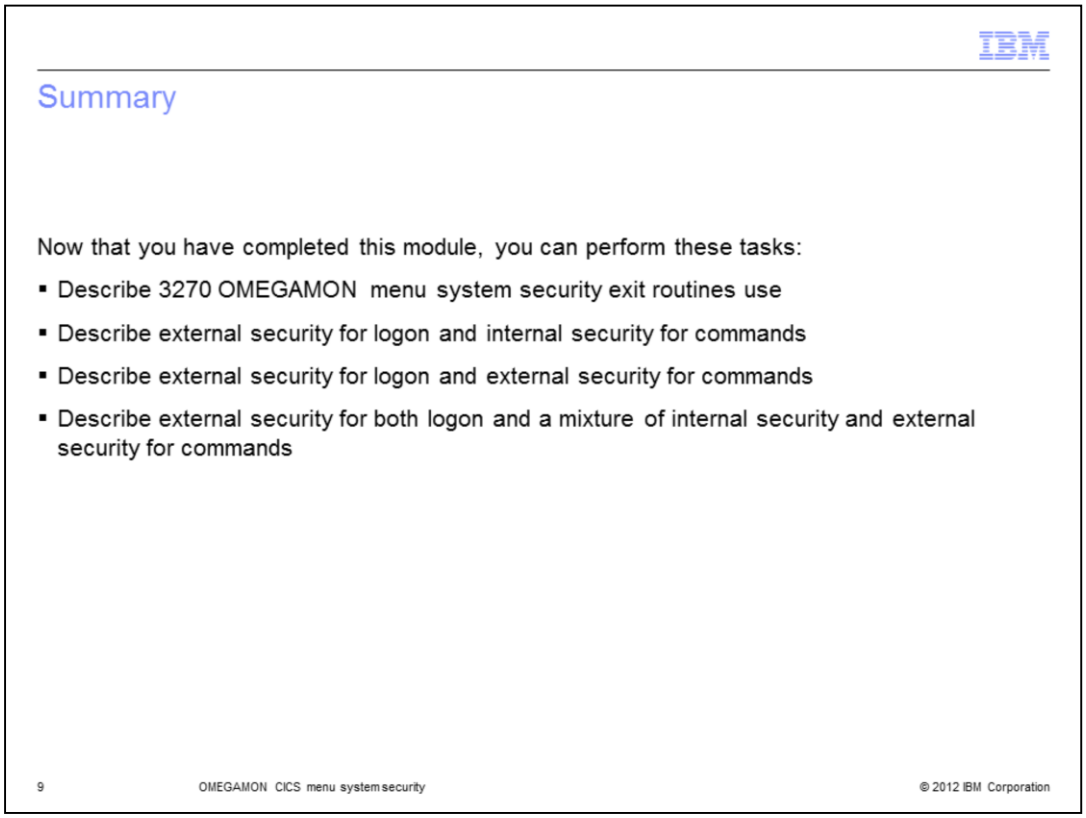
The locking feature is a form of external security designed to prevent users from changing their internal security level. Their level of authority is set once only at log on. It can be fixed to one of four levels.

Level 0, 1, 2, or 3. Consider these two items when using the locking feature:

> Although the *locking feature* is implemented in the external security exit routine, it is designed to lock the users internal security level. Therefore, it affects only those commands marked as **EXTERNAL=NO**.
>
> You must define a *user security level* in CA-ACF2, RACF, or CA-TOP SECRET as an **INITIAL n** resource, where **n** is a number **0** to **3**, and assign corresponding values to commands in the security update program.



The *Planning and Configuration Guide* is a very good reference.



Now that you have completed this module, you can perform these tasks:

- Describe 3270 OMEGAMON (menu system) security exit routines use
- Describe external security for log on and internal security for commands
- Describe external security for log on and external security for commands
- Describe external security for both log on and a mixture of internal security and external security for commands

Trademarks, disclaimer, and copyright information