文章
Nicky Zhu · 一月 10, 2021 阅读大约需 8 分钟

Open Exchange

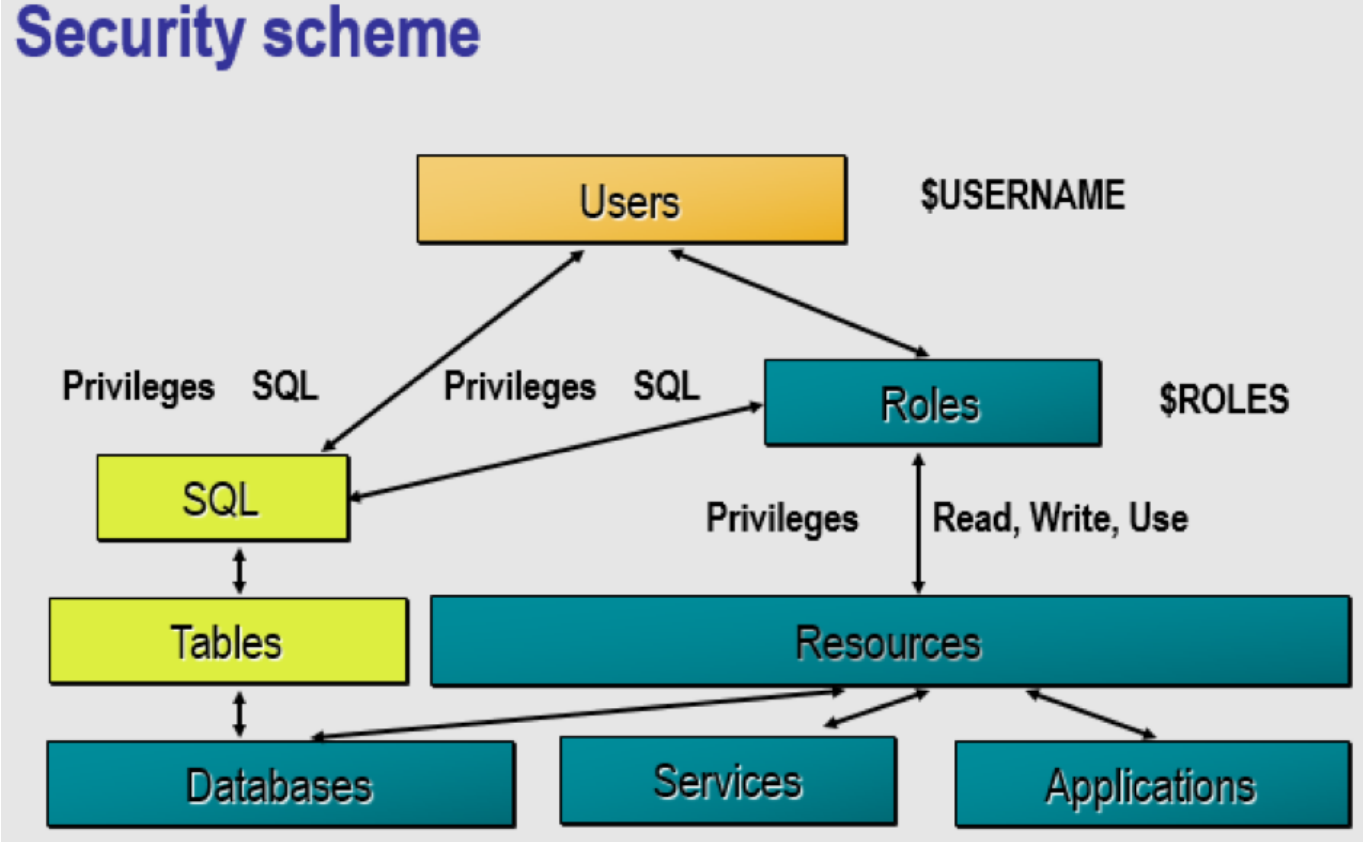
增强InterSystems IRIS DBMS的安全性

当您首次使用InterSystems IRIS时，通常只需安装最低安全级别的系统。您输入密码的次数会比较少，这样有利于快速了解和操作开发服务和Web应用程序。而且，最低的安全性有时更便于部署开发项目或解决方案。

然而，有时需要将项目移出开发环境，迁移到一个可能很不友好的互联网环境中。在部署到生产环境之前，需要使用最大的安全设置（即，完全锁定）对其进行测试。这就是我们在本文中将要讨论的内容。

如果想更全面地了解InterSystems Caché、Ensemble和IRIS中的DBMS安全性问题，请阅读我的另一篇文章《在生产环境中安装InterSystems Caché DBMS的相关建议》。

InterSystems IRIS中安全系统的设计概念是针对不同的类别（用户、角色、服务、资源、特权和应用程序）应用不同的安全设置。



可以为用户分配角色。用户和角色可以对资源（数据库、服务和应用程序）拥有不同的读、写和使用权限。用户和角色还可以对数据库中的SQL表拥有SQL权限。

安全级别的差异

用户在安装InterSystems IRIS时，可以为其选择安全级别：最小、正常或锁定。这些级别在用户参与程度、可用角色和服务，以及服务和应用程序的身份验证方法的配置方面存在差异。如需了解更多信息，请阅读《InterSystems IRIS安装准备》指南中的《InterSystems安全性准备》章节。

在文档中，您可以找到下面这些表格，其中显示了每个级别的安全设置。这些安全设置可以在系统管理门户界面进行更改。

初始用户安全设置

Security Setting	Minimal	Normal	Locked Down
Password Pattern	3.32ANP	3.32ANP	8.32ANP
Inactive Limit	0	90 days	90 days
Enable SYSTEM User	Yes	Yes	No
Roles assigned to UnknownUser	%All	None	None

初始服务属性设置

Service Property	Minimal	Normal	Locked Down
Use Permission is Public	Yes	Yes	No
Requires Authentication	No	Yes	Yes
Enabled Services	Most	Some	Fewest

初始服务启用设置

Service	Minimal	Normal	Locked Down
%ServiceBindings	Enabled	Enabled	Disabled
*%ServiceCSP	Enabled	Enabled	Enabled
%ServiceCacheDirect	Enabled	Disabled	Disabled
%ServiceCallIn	Enabled	Disabled	Disabled
%ServiceComPort	Disabled	Disabled	Disabled
%ServiceConsole	Enabled	Enabled	Enabled
%ServiceECP	Disabled	Disabled	Disabled
%ServiceMSMActivate	Disabled	Disabled	Disabled
%ServiceMonitor	Disabled	Disabled	Disabled
%ServiceShadow	Disabled	Disabled	Disabled
%ServiceTelnet	Disabled	Disabled	Disabled
%ServiceTerminal	Enabled	Enabled	Enabled
%ServiceWebLink	Disabled	Disabled	Disabled

*InterSystems IRIS环境下，%ServiceCSP应用%ServiceWebGateway。
不同的操作系统所使用的服务略有不同。

如何提高安全性

您需要为每个启用的服务选择合适的身份验证方法，包括：无认证（unauthenticated）、密码、Kerberos或授权。

您还需要禁用系统中未使用的web应用程序。对已启用的web应用程序选择正确的身份验证方法：认证、密码、Kerberos、授权、登录或cookie。

当然，管理员可以为每一个项目和解决方案选择安全设置，以满足客户的项目要求。整个过程应始终保持一种平衡，即，一方面要保证系统足够方便以支持用户完成实际工作，另一方面又要保证系统足够安全能够阻止入侵者。不过众所周知，被禁用的系统才是最安全的系统。

如果遇到需要多次手动提高系统安全级别的情况，这就是一个明确的迹象，表明需要编写一个软件模块来解决这些问题。

实际上，InterSystems Open
Exchange提供了一个锁定（LockDown）程序，可以帮助您提高安全性。该程序的源代码可以在InterSystems isc-apptools-lockdown页面的存储库中找到。

LockDown程序有以下几种作用：

首先，更改以下预安装用户的密码：

- Admin,
- CSPSystem,
- IAM,
- SuperUser,
- UnknownUser,
- Ensemble,
- SYSTEM.

其次，禁用除以下服务之外的所有服务：

- %%serviceweb gateway
- %serviceconsole
- %servicelogin
- %serviceterminal

再次，为所有web应用程序设置密码保护，包括：

- /csp/ensdemo
- /csp/samples
- /csp/user
- /isc/studio/usertemplates
- /csp/docbook
- /csp/documatic
- /isc/studio/rules
- /isc/studio/templates

最后，设置系统范围内的安全参数，包括：

- 密码复杂度为 "8.32 ANP"
- 限制90天内不活跃的用户
- 开启审计和其他所有安全相关的事件

您可以从GitHub下载LockDown.cls，在系统上安装好LockDown程序。然后在终端输入以下内容：

```
USER>zn “%SYS”
%SYS>do $system.OBJ.Load("/home/irisusr/LockDown.cls","ck")
```

或者可以使用以下命令从公共注册中心通过ZPM批处理管理器进行安装：

```
USER>zn “%SYS”
%SYS> zpm “install isc-apptools-lockdown”
```

执行LockDown程序

强烈建议在执行LockDown程序之前先进行备份。

必须从%SYS命名空间执行LockDown程序。如果您不想更改所有预安装用户的密码，请将第一个参数保留为空。

如果希望保留使用IRIS Studio、Atelier或VSCode编辑程序和类的能力，请不要禁用%ServiceBindings服务，只需将bindings参数设置为1即可。下面是一个示例：
do ##class(App.Security.LockDown).Apply("New Password 123",.msg,1)
此模块还包含一个功能，该功能在系统密码被盗用、需要替换所有预装帐户但无需执行锁定的情况下很有用。可以参考下面的运行：
do ##class(App.Security.LockDown).Change Password("New Password 123",
"Admin,CSPSystem,IAM,SuperUser,Unknown User, Ensemble,SYSTEM")
在执行锁定之后，应用程序或项目极有可能会停止工作。为了解决这个问题，需要将一些安全设置恢复到原始状态。这个操作可以通过管理门户界面（安全性部分）实现或以编程方式完成。

锁定后更改安全设置

锁定之后，如果您的web应用程序使用了密码以外的身份验证方法，就需要进行启用。

建议运行软件模块zpm-registry-test-deployment，它有一个在ZPM-registry项目中使用LockDown的示例。

在IRIS上以最低的安全级别安装该项目，安装结束时将开始运行其中的代码。代码可以用来：
- 更改所有预安装用户的密码。
- 禁用此项目中未使用的所有服务。
- 为系统上的所有应用程序启用密码保护，但Web应用程序/注册表（允许未经授权的用户获取注册表中的软件包列表）除外。
- 创建一个拥有在注册表中发布新包特权的新用户。该用户必须对IRISAPP数据库中的项目表具有写权限。

创建一个新用户：

```
set tSC= ##class(App.Security.LockDown).CreateUser(pUsername, "%DB_"_Namespace, pPass
word, "ZMP registry user",Namespace)
If $$$ISERR(tSC) quit tSC
write !,"Create user "_pUsername
```

为新的未授权用户添加权限：

```
set tSC=##class(App.Security.LockDown).addSQLPrivilege(Namespace, "1,ZPM.Package", "s
", "UnknownUser")
set tSC=##class(App.Security.LockDown).addSQLPrivilege(Namespace, "1,ZPM.Package", "s
", pUsername)
set tSC=##class(App.Security.LockDown).addSQLPrivilege(Namespace, "1,ZPM.Package_depe
ndencies", "s", pUsername)
set tSC=##class(App.Security.LockDown).addSQLPrivilege(Namespace, "1,ZPM_Analytics.Ev
ent", "s", pUsername)
set tSC=##class(App.Security.LockDown).addSQLPrivilege(Namespace, "9,ZPM.Package_Exte
nt", "e", pUsername)
set tSC=##class(App.Security.LockDown).addSQLPrivilege(Namespace, "9,ZPM_Analytics.Ev
ent_Extent", "e", pUsername)
If $$$ISERR(tSC) quit tSC
write  !,"Add privileges "
```

运行LockDown项目：

```
set tSC= ##class(App.Security.LockDown).Apply(NewPassSys)
If $$$ISERR(tSC) quit tSC
```

```
Change the settings for the web app so that an unknown user can log in:
set prop("AutheEnabled")=96
set tSC=##class(Security.Applications).Modify("/registry",.prop)
If $$$ISERR(tSC) quit tSC
write !,"Modify /registry "

Change the settings for the %service_terminal service, changing the authorization met
hod to Operating System, Password:
set name="%service_terminal"
set prop("Enabled")=1
set prop("AutheEnabled")=48 ; Operating System,Password
set tSC=##class(Security.Services).Modify(name,.prop)
If $$$ISERR(tSC) quit tSC
write !,"Modify service terminal"
```

总结

在本文中，我讨论了为何要提高系统安全性级别的原因，并且通过一个InterSystems LockDown程序运行示例，演示了如何通过编程的方式提升安全性。

在本文所介绍的方法中，我们首先关闭了系统中的所有内容（即，设置最大安全级别）。然后通过开放项目运行所需的服务和应用程序（但仅限于这些）来控制安全性。我相信还有许多其他的方法和最佳实践，欢迎大家在文章评论区留言告诉我们。

#安全 #新手 #系统管理 #Caché #Ensemble #InterSystems IRIS #InterSystems IRIS for Health
在 InterSystems Open Exchange 上检查相关应用程序

源URL:https://cn.community.intersystems.com/post/%E5%A2%9E%E5%BC%BAintersystems-iris-dbms%E7%9A%84%E5%AE%89%E5%85%A8%E6%80%A7