記事
Toshihiko Minamoto · 2021年10月26日 10m read

Open Exchange

InterSystems IRIS DBMSのセキュリティ強化

InterSystems IRISを初めて使用し始める際には、最低限のセキュリティレベルでのみシステムをインストールするのが通例です。 パスワードを入力する回数が少なくて済むため、初めて作業を始めるときに、開発サービスやWebアプリケーションの操作がより簡単になるからです。 また、開発済みのプロジェクトまたはソリューションをデプロイする際には、最小限のセキュリティを適用している方が便利な場合があります。 それでも、プロジェクトを開発環境から非常に敵対的な可能性のあるインターネット環境に移行する時が来れば、本番環境にデプロイされる前に、最大限のセキュリティ設定（つまり、完全なロックダウン状態）でテストしなければなりません。 これがこの記事の論点です。
InterSystems Caché、Ensemble、およびIRISにおけるDBMSセキュリティ問題をさらに包括的に説明した記事については、私の別の記事、「Recommendations on installing the InterSystems Caché DBMS for a production environment」（本番環境向けにInterSystems Caché DBMS
をインストールするための推奨事項）をご覧ください。 InterSystems IRISのセキュリティシステムは、さまざまなカテゴリ（ユーザー、サービス、リソース、特権、およびアプリケーション）に異なるセキュリティ設定を適用

ユーザーにはロールを割り当てることができます。 ユーザーとロールには、データベース、サービス、およびアプリケーションといったリソースに対し、さまざまな読み取り、書き込み、および使用の権限を付与することができます。 ユーザーとロールには、データベースのSQLテーブルに対するSQL特権も与えられます。

セキュリティレベルの違い

InterSystems
IRISをインストールするときに、最小、通常、またはロックダウンというセキュリティレベルを選択できます。 レベルは、ユーザーエンゲージメントの程度、使用可能なロールとサービス、およびサービスとアプリケーションの認証方法の構成に違いがあります。 詳細については、『InterSystems
IRISのインストール準備』ガイドの「InterSystemsセキュリティの準備」セクションをご覧ください。
ドキュメントには、レベルごとのセキュリティ設定を説明する、以下の表が含まれています。
この設定の変更は、システム管理ポータルのインターフェースで行えます。

ユーザーセキュリティの初期設定

次の概要に基づいています。

セキュリティ設定	最小限	通常	ロックダウン
パスワードのパターン	3.32ANP	3.32ANP	8.32ANP
非アクティブになる期間	0	90日	90日
SYSTEMユーザーの有効化	可	可	不可
UnknownUserに割り当てられるロール	%All	なし	なし

サービスの初期プロパティ

サービスプロパティ	最小限	通常	ロックダウン
パブリック許可を使用	はい	はい	いいえ
認証が必要	いいえ	はい	はい
有効化されたサービス	ほとんど	いくつか	最小限

初期設定で有効にされているサービスの設定

サービス	最小限	通常	ロックダウン

サービス	最小限	通常	ロックダウン
%ServiceBindings	有効	有効	無効
*%ServiceCSP	有効	有効	有効
%ServiceCacheDirect	有効	無効	無効
%ServiceCallIn	有効	無効	無効
%ServiceComPort	無効	無効	無効
%ServiceConsole	有効	有効	有効
%ServiceECP	無効	無効	無効
%ServiceMSMActivate	無効	無効	無効
%ServiceMonitor	無効	無効	無効
%ServiceShadow	無効	無効	無効
%ServiceTelnet	無効	無効	無効
%ServiceTerminal	有効	有効	有効
%ServiceWebLink	無効	無効	無効

*InterSystems IRISの場合、%ServiceCSPは%ServiceWebGatewayを適用します。
使用されるサービスは、オペレーティングシステムごとにわずかに異なります。

セキュリティを向上させるには

有効なサービスごとに、適切な認証方法（非認証、パスワード、Kerberos、または委任）を選択する必要があります。
また、システムで使用されていないWebアプリケーションを無効にする必要もあります。 有効になっているWebアプリケーションについては、正しい認証方法（認証済み、パスワード、Kerberos、委任、ログイン、またはcookie）を選択する必要があります。
もちろん、プロジェクトは顧客の要件に従って機能できるように、管理者がプロジェクトやソリューションごとにセキュリティ設定を選択するため、 ユーザーが作業を実際に行えるようにシステムの利便性を十分に維持しながら、侵入者を寄せ付けないように十分なセキュリティを保つバランスを常に取らなければなりません。
ご承知のとおり、最も安全なシステムは無効化されたシステムです。

システムのセキュリティレベルを何度も手動で引き上げる必要があるのであれば、それは、そういった問題を解決するソフトウェアモジュールを記述する必要があるという兆候に違いありません。 実際、InterSystems Open Exchangeには、セキュリティの向上に役立つロックダウンプログラムがあります。
プログラムのソースコードは、InterSystems isc-apptools-lockdownページのリポジトリにあります。
LockDownプログラムは以下のことを行います。

まず、プレインストールされた以下のユーザーのパスワードを変更します。

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

次に、以下を除くすべてのサービスを無効にします。

- %%servicewebゲートウェイ
- %serviceconsole
- %servicelogin
- %serviceterminal

さらに、以下を含むすべてのWebアプリケーションにパスワード保護を設定します。

- /csp/ensdemo
- /csp/samples

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

最後に、以下のようなシステムワイドのセキュリティパラメーターを設定します。

- パスワードの複雑さ「8.32 ANP」
- 90日間の非利用アカウント制限
- 監査およびすべてのセキュリティ関連イベント
  LockDownプログラムは、GitHubからLockDown.cls
  をダウンロードして、システムにインストールできます。
  そして、ターミナルモードで、以下を入力します。

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

または、以下のコマンドを使用して、パブリックレジスタからZPMバッチマネージャを使ってインストールすることも可能です。

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

ロックダウンの実行

ロックダウンを実行する前に、バックアップを作成しておくことを強くお勧めします。
LockDownプログラムは、%SYS領域から実行する必要があります。 プレインストールされたすべてのユーザーのパスワードを変更しない場合は、最初のパラメーターを空のままにします。 IRIS Studio、Atelier、またはVSCodeを使ってプログラムとクラスを編集する機能を維持する場合は、%ServiceBindingsサービスを無効にしないでください。 これを確実に機能させるには、バインディング引数を1に設定する必要があります。
次に例を示します。
do ##class(App.Security.LockDown).Apply("New Password 123",.msg,1)
このモジュールには、システムパスワードが改ざんされた場合やロックダウンを実行せずにプレインストールされたすべてのアカウントを入れ替える必要がある場合に役立つ関数も含まれています。
次のようにして実行することができます。
do ##class(App.Security.LockDown).Change Password("New Password 123",
"Admin,CSPSystem,IAM,SuperUser,Unknown User, Ensemble,SYSTEM")
ほとんどの場合、ロックダウンを実行した後は、アプリケーションあまたはプロジェクトが動作しなくなります。
これを修正するには、いくつかのセキュリティ設定を元の状態に復元する必要があります。
これは、管理ポータルインターフェース（セキュリティセクション）またはプログラムで実行できます。

ロックダウン後のセキュリティ設定の変更

ロックダウン後、Webアプリケーションがパスワード以外の認証方法を使用している場合、それらを有効にする必要があります。 zpm-registry-test-deployment
というソフトウェアモジュールを実行することをお勧めします。これには、ZPM-registryプロジェクト向けのLockDownの使用例が含まれています。
次のコードは、インストールの最後に適用されています。
プロジェクトは、最小限のセキュリティレベルでIRISにインストールされました。
以下は、コードが実行する必要のあった項目です。
- プレインストールされたすべてのユーザーのパスワードを変更する。
- このプロジェクトで使用されていないすべてのサービスを無効にする。

- Webアプリケーション/レジストリ（許可されていないユーザーはレジストリのパッケージのリストを取得できます）を除く、システム上の全アプリケーションのパスワード保護を有効にする。
- レジストリに新しいパッケージを公開する権限を持つ新しいユーザーを作成する。
このユーザーには、IRISAPPデータベースのプロジェクトテーブルに対する書き込み権限が必要です。

新しいユーザーを作成します。

```
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
```

新しいユーザーと許可されていないユーザーの特権を追加します。

```
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 "
```

LockDownプログラムを実行します。

```
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"
```

まとめ

この記事では、システムのセキュリティレベルを引き上げる理由とこれをプログラムで実行する方法を説明し、In

terSystems LockDownプログラムを使った例を紹介しました。
最初にシステム内のすべてを終了する方法を使用しました（つまり、最大セキュリティレベルを設定しました）。次に、プロジェクトが機能するのに必要なサービスとアプリケーションのみを開いて、セキュリティを緩和しました。 他の方法やベストプラクティスが必ず存在すると思っています。コミュニティによるこの記事のディスカッションの一環として、それらについてぜひ聞かせてください。

#システム管理 #セキュリティ #初心者 #Caché #Ensemble #InterSystems IRIS #InterSystems IRIS for Health
InterSystems Open Exchangeで関連アプリケーションを確認してください

ソースURL:https://jp.community.intersystems.com/post/intersystems-iris-dbms%E3%81%AE%E3%82%BB%E3%82%AD%E3%83%A5%E3%83%AA%E3%83%86%E3%82%A3%E5%BC%B7%E5%8C%96