

Framework do plugin do Gerenciador de Segurança

Equipe de documentação de Boca Raton



Framework do plugin do Gerenciador de Segurança

Equipe de documentação de Boca Raton

Copyright © 2023 HPCC Systems®. All rights reserved

Sua opinião e comentários sobre este documento são muito bem-vindos e podem ser enviados por e-mail para <docfeedback@hpccsystems.com>. Inclua a frase **Feedback sobre documentação** na linha de assunto e indique o nome do documento, o número das páginas e número da versão atual no corpo da mensagem.

LexisNexis e o logotipo Knowledge Burst são marcas comerciais registradas da Reed Elsevier Properties Inc., usadas sob licença. Os demais produtos, logotipos e serviços podem ser marcas comerciais ou registradas de suas respectivas empresas. Todos os nomes e dados de exemplo usados neste manual são fictícios. Qualquer semelhança com pessoas reais, vivas ou mortas, é mera coincidência.

2023 Version 8.10.24-1

Desenvolvimento do Gerenciador de Segurança (Security Manager)	4
Desenvolvimento do Plugin	5
Configurar e implementar o Plugin do Gerenciador de Segurança	8
Como Configurar um Plugin do Gerenciador de Segurança	9

Desenvolvimento do Gerenciador de Segurança (Security Manager)

O framework do Plugin do Gerenciador de Segurança (Security Manager Plugin) oferece um mecanismo de criação e implementação de Security Manager Plugins customizados.

Desenvolvimento do Plugin

Um Security Manager Plugin customizado consiste de um arquivo de **biblioteca** (.so ou .dll) que fornece implementações da interface `ISecManager`, uma **definição de configuração** (articulada como um esquema XSD) e uma declaração de componente (arquivo `buildset.xml`).

Requisitos da biblioteca

- Deve implementar a interface `ISecureManager`
- Deve expor um método de fábrica que retorne instâncias da implementação do `ISecManager`.

Exemplo de um nome padrão de método de fábrica:

```
extern "C"
{
    ISecManager * createInstance(const char *serviceName,
                                IPropertyTree &secMgrCfg,
                                IPropertyTree &bndCfg);
}
```

O framework espera ter acesso ao método `"createInstance()"`. Se o desenvolvedor optar por fornecer outros métodos de fábrica, ele pode substituir o nome padrão na configuração, mas deve conter a seguinte assinatura:

```
ISecManager methodname(const char *, IPropertyTree &, IPropertyTree &)
```

Definição de Buildset O plugin declara a si mesmo como um componente do Security Manager Plugin do HPCC, assim como declara a localização dos arquivos do plugin e o esquema de definição de configuração.

EXEMPLO:

```
<Environment>
  <Programs>
    <Build name="_" url="/opt/HPCCSystems">
      <BuildSet deployable="no"
                installSet="deploy_map.xml"
                name="mysecuritypluginname"
                path="componentfiles/mysecuritypluginname"
                processName="MySecurityPluginName"
                schema="myCustom_secmgr.xsd">
      </BuildSet>
    </Build>
  </Programs>
</Environment>
```

Definição de configurações O plugin deve fornecer uma definição dos elementos da configuração e a estrutura que espera receber no momento em que é instanciado. O arquivo XSD é utilizado pelo componente do Gerenciador de Configurações do HPCC e é apresentado como um formulário GUI. A definição de configurações é definida como um elemento do nome do componente (como declarado no `buildset`) seguido pelos atributos e/ou elementos complexos.

Cada plugin deve declarar quatro atributos em sua definição de configurações, além de qualquer configuração customizada definida pelo plugin: `'type'`, `'name'`, `'libName'`, e `'instanceFactoryName'`

- **type** – Este atributo deve estar na forma de somente leitura e definido para `'SecurityManager'`
- **name** – O nome da instância do Security Manager Plugin customizado
- **libName** O nome da biblioteca que fornece instâncias para este tipo de Security Manager Plugin
- **instanceFactoryName** – Nome do método fornecido pela biblioteca, que é responsável pela criação das instâncias do Security Manager Plugin

EXEMPLO:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="MySecurityPluginType">
    <xs:complexType>
      <xs:attribute name="type" type="SecurityManager"
        use="required" default="SecurityManager">
        <xs:annotation><xs:appinfo>
          <viewType>hidden</viewType>
        </xs:appinfo></xs:annotation>
      </xs:attribute>
      <xs:attribute name="name" type="xs:string" use="required">
        <xs:annotation><xs:appinfo>
          <tooltip>Name for this Security Manager Plugin instance</tooltip>
          <required>true</required>
        </xs:appinfo></xs:annotation>
      </xs:attribute>
      <xs:attribute name="libName" type="xs:string" use="optional">
        <xs:annotation><xs:appinfo>
          <tooltip>The Security Manager library name (.so)</tooltip>
        </xs:appinfo></xs:annotation>
      </xs:attribute>
      <xs:attribute name="instanceFactoryName" type="xs:string"
        use="optional" default="createInstance">
        <xs:annotation><xs:appinfo>
          <tooltip>The factory method name in the
            Security Manager library (.so)</tooltip>
        </xs:appinfo></xs:annotation>
      </xs:attribute>
      <xs:sequence>
        <xs:element name="compoundOption" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute name="Option" type="xs:string" use="required">
              <xs:annotation><xs:appinfo>
                <tooltip>This is an example compound option element
                  which Security Manager Plugins can define</tooltip>
              </xs:appinfo></xs:annotation>
            </xs:attribute>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="optionalAttribute" type="xs:string" use="optional">
        <xs:annotation><xs:appinfo>
          <tooltip>This is an example optional attribute
            which Security Manager Plugins can define</tooltip>
        </xs:appinfo></xs:annotation>
      </xs:attribute>
      <xs:attribute name="samplepasswordfile" type="xs:string" use="required">
        <xs:annotation><xs:appinfo>
          <tooltip>An attribute which defines a file name required
            by this Security Manager Plugin</tooltip>
        </xs:appinfo></xs:annotation>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

genenvrules.conf – (opcional) Este arquivo permite que o plugin adicione a si mesmo na lista "do_not(automatically)_generate". Embora seja um arquivo opcional, ele é frequentemente necessário.

EXEMPLO:

```
do_not_generate=mysecuritypluginname
```

Regras de Transformação de Configuração – (opcional) especificado como modelo xsl, este conjunto de regras pode ser aplicado às configurações XML. Consulte os modelos XSL na árvore de origem do HPCC Systems.

Exemplo Concreto

A plataforma HPCC inclui a implementação do plugin do gerenciador de segurança (HTPasswd) e pode ser usada como orientação para o processo de desenvolvimento do plugin:

<https://github.com/hpcc-systems/HPCC-Platform/tree/master/system/security/plugins/httpasswdSecurity>

Configurar e implementar o Plugin do Gerenciador de Segurança

As seções a seguir detalham o processo de configuração de seu HPCC Systems para usar o Plugin do Gerenciador de Segurança.

Como Configurar um Plugin do Gerenciador de Segurança

Após ter sido instalado, o plugin pode ser configurado na plataforma HPCC através do Configuration Manager.

1. Pare a execução de todos os componentes do HPCC.

Verifique se eles não estão mais sendo executados. É possível usar um comando único, como:

```
sudo /opt/HPCCSystems/sbin/hpcc-run.sh -a hpcc-init status
```

2. Conecte seu navegador de Internet à interface da Web do Gerenciador de Configurações (Configuration Manager).

Use o URL `http://<configmgr_IP_Address>:8015`

onde `<configmgr_IP_Address>` é o endereço IP do nó que está executando o Gerenciador de Configurações.

3. Selecione o botão de opção **Advanced View**.
4. Selecione o arquivo de configuração em XML adequado.

Observação: O Gerenciador de Configurações nunca funciona no arquivo de configurações ativo. Após terminar a edição, será necessário copiar o arquivo `environment.xml` para o local ativo e forçá-lo a todos os nós.

5. Marque a caixa de seleção **Write Access**.
6. Crie uma instância do Plugin do Gerenciador de Segurança:
 - a. Clique com o botão direito no Painel de navegação ao lado esquerdo.
 - b. Selecione **New Components**
 - c. Selecione o componente adequado `<name_of_Security_Manager_plugin>`

7. Configure o Plugin do Gerenciador de Segurança: (Exemplo mostrado usando o plugin Htpasswd*)

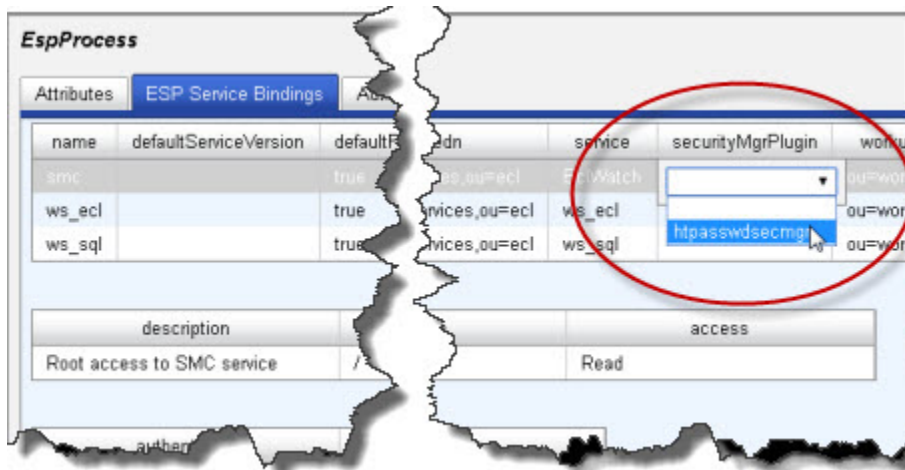
Figure 1. Página “Security Mgr Configuration” (Configuração do Security Manager)



- Forneça um nome para a instância
- Forneça um nome da biblioteca (totalmente qualificado)
- O InstanceFactoryName será definido para "createInstance" se a biblioteca especificada na etapa anterior fornecer um método de fábrica alternativo.
- Forneça quaisquer entradas customizadas exigidas. No exemplo mostrado, *htpasswdFile* é uma entrada customizada.

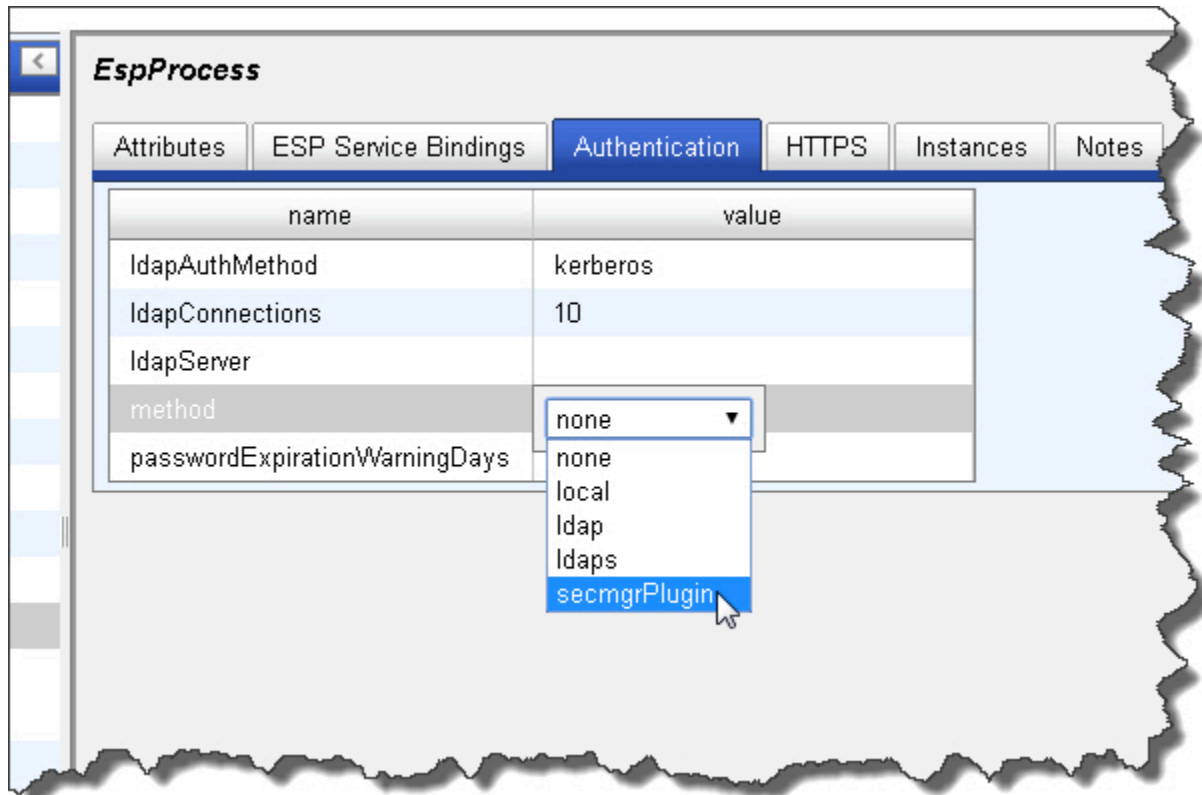
8. Associe o Plugin do Gerenciador de Segurança às conexões do ESP
 - a. Clique no Esp de destino no painel do navegador ao lado esquerdo.
 - b. Selecione a aba **ESP (Process) Service Bindings**
 - c. Nas ligações de destino, selecione a instância securityMgrPlugin adequada a partir da lista suspensa.

Figure 2. Ligação ao ESP



9. Ativar o uso do Plugin do Gerenciador de Segurança – Selecione a aba **Authentication** (Autenticação) e na entrada do método selecione **secmgrPlugin**

Figure 3. Página “Security Mgr Configuration” (Configuração do Security Manager)



10. Salve o arquivo do ambiente
11. Distribua o arquivo do ambiente para todos os nós em seu cluster
12. Reinicialize seu ambiente.

Tutorial em Vídeo

Precisa de informações adicionais? Assista ao vídeo do tutorial em que demonstramos como configurar um security plugin (plugin de segurança).

<https://www.youtube.com/watch?v=INVwEOFkKgY&feature=youtu.be>

Clique no link acima para assistir ao vídeo.