Red Hat JBoss Enterprise Application Platform 6.4

Kerberos で SSO を設定する方法

Kerberos で SSO を設定する方法

Last Updated: 2023-02-21

Red Hat JBoss Enterprise Application Platform 6.4 Kerberos で SSO を設定する方法

Kerberos で SSO を設定する方法

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

本書の目的は、Red Hat JBoss Enterprise Application Platform 6 内における Kerberos によるシングルサインオン (SSO) を検証し、JBoss EAP 6 で Kerberos を用いた SSO の実用的な設定例を提供することです。本書では Kerberos を用いた SSO について深く掘り下げ、JBoss EAP 6 内でのセットアップおよび設定方法を説明します。本書をお読みいただく前に、Red Hat JBoss Enterprise Application Platform 6 の『セキュリティーアーキテクチャー』をお読みいただき、このドキュメントに記載されている SSO および Kerberos の情報を理解できるようにしてください。また、本書では JBoss EAP 6 CLI インターフェースを使用して設定の変更を行います。スタンドアロンの JBoss EAP 6 インスタンスと JBoss EAP 6 ドメインの両方で CLI を使用する方法の詳細

は、Red Hat JBoss Enterprise Application Platform 6 管理および設定ガイドの管理 CLI セクションを参照してください。本書をお読みいただくと、SSO および Kerberos、JBoss EAP 6 との関係、および設定方法を深く実践的に理解することができます。

目次

第1章 KERBEROS を用いた SSO の検証 **3**
1.1. SSO および KERBEROS とは 3
1.2. KERBEROS のコンポーネント 3
1.3. その他のコンポーネント 4
1.3.1. SPNEGO 4
1.3.2. JBoss Negotiation 4
1.4. KERBEROS の統合 5
1.5. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM における KERBEROS による SSO の提供方法 5
1.5.1. デスクトップベース SSO での Kerberos による認証および承認 5
1.5.2. Kerberos および Red Hat JBoss Enterprise Application Platform 6 6

第2章 KERBEROS を使用した RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 6 での SSO の設定方法 **7**
2.1. コンポーネント 7
2.1.1. JBoss Negotiation Toolkit 7
2.1.2. Kerberos 環境 7
2.1.3. Red Hat JBoss Enterprise Application Platform のこれまでのバージョンとの相違点 7
2.1.4. Red Hat JBoss Enterprise Application Platform 6 インスタンスの設定 8
2.1.4.1. 1.サーバーアイデンティティー（ホスト）セキュリティードメインの設定 8
2.1.4.2. 2.Web アプリケーションセキュリティードメインの設定 10
2.1.4.3. 3.関連するシステムプロパティーの設定 10
2.1.5. Web アプリケーションの設定 11
2.1.5.1. 1.SPNEGO 認証メソッドを使用する web.xml の設定 12
2.1.5.2. 2.設定されたセキュリティードメインを使用する jboss-web.xml の設定 12
2.1.5.3. 3.JBoss Negotiation 依存関係のデプロイメントへの追加 13
2.2. ACTIVE DIRECTORY に関する注意点 13
2.2.1. Microsoft Windows ドメインでの JBoss Negotiation の設定 13
2.2.1.1. 1.既存のサービスプリンシパルマッピング消去 13
2.2.1.2. 2.ホストユーザーアカウントを作成します。 14
2.2.1.3. 3.USER_NAME と HOST_NAME との間のマッピングを定義します。 14
2.2.1.4. 4.EAP JBoss がインストールされているサーバーに、ユーザーのキータブをエクスポートします。 14
2.2.1.5. 5.セキュリティードメイン内でプリンシパルを定義します。 15

第3章 その他の機能 **16**
3.1. FORM ログインをフォールバックとして追加 16
3.1.1. 1.Kerberos および SPNEGO を使用する Red Hat JBoss Enterprise Application Platform 6 と Web アプリケーションの設定 16
3.1.2. 2.フォールバック認証用のセキュリティードメインの更新 16
3.1.3. 3.ログインおよびエラーページの追加 17
3.1.4. 4.web.xml の編集 18
3.2. KERBEROS での管理インターフェースのセキュア化 18
3.2.1. 1.補助システムプロパティーの有効化 19
3.2.2. 2.Kerberos サーバーアイデンティティーのセキュリティーレルムへの追加 19
3.2.3. 3.セキュリティーレルムでの認証方法の更新 19

第1章 KERBEROS を用いた SSO の検証

1.1. SSO および KERBEROS とは

SSO および Kerberos の基本的な背景は、Red Hat JBoss Enterprise Application Platform 6 『セキュリティーアーキテクチャー』の「**シングルサインオン(SSO)**」セクションを参照してください。

1.2. KERBEROS のコンポーネント

Kerberos は、秘密鍵の暗号化を使用してクライアント/サーバーアプリケーションのユーザー認証を可能にするネットワークプロトコルです。通常、Kerberos はネットワーク上のデスクトップユーザーの認証に使用されますが、追加のツールを使用すると、web アプリケーションのユーザー認証に使用でき、複数の web アプリケーションに SSO を提供することができます。そのため、デスクトップネットワーク上で認証済みのユーザーは、再認証しなくても web アプリケーションのセキュアなリソースにシームレスにアクセスできます。ユーザーはデスクトップベースの認証メカニズムを使用して認証され、認証されたユーザーの認証トークンまたはチケットは web アプリケーションによっても使用されるため、この概念はデスクトップベースの SSO と呼ばれます。この SSO メカニズムは、すべてブラウザーを使用してユーザーの認証とトークンの発行を行うブラウザーベースの SSO など、他の SSO メカニズムとは異なります。

Kerberos プロトコルは、認証および承認で使用する複数のコンポーネントを定義します。

チケット

チケット は、プリンシパルに関する認証および承認決定を発行および実行するために Kerberos が使用するセキュリティートークンの形式です。

認証サービス (AS)

認証サービス (AS) は、プリンシパルが最初にネットワークにログインするときにプリンシパルを確認します。認証サービスは、**チケット保証チケット (TGT)** の発行を行います。TGT は、**チケット保証サービス (TGS)** の認証に必要で、セキュアなサービスおよびリソースに再度アクセスするときにも必要になります。

チケット保証サービス (TGS)

チケット保証サービス (TGS) は、**サービスチケット** と特定のセッション情報を、プリンシパルとアクセスするターゲットサーバーに発行します。これは 、プリンシパルによって提供される TGT と接続先の情報を基にします。このサービスチケットとセッション情報は、接続先への接続を確立し、セキュアなサービスまたはリソースへアクセスするために使用されます。

キー配布センター (KDC)

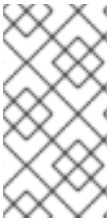
キー配布センター (KDC) は、TGS と AS の両方を格納するコンポーネントです。KDC、クライアントまたはプリンシパル、およびサーバーまたはセキュアなサービスの 3 つのコンポーネントは、Kerberos 認証の実行に必要です。

チケット保証チケット (TGT)

チケット保証チケット (TGT) は、AS がプリンシパルに発行するチケットのタイプです。プリンシパルがユーザー名とパスワードを使用して AS に対して認証されると、TGT が付与されます。TGT はクライアント (プリンシパル) によってローカルでキャッシュされますが、KDC のみが読み取れ、クライアントは読み取りできないように暗号化されます。これにより、AS は TGS によって使用される TGT の承認データや他の情報をセキュアに保存することができ、TGS はこのデータを使用して承認決定を行うことができます。

サービスチケット (ST)

service ticket (ST) は、TGT と目的の接続先を基にして TGS がプリンシパルに発行するチケットのタイプです。プリンシパルは TGS に TGT を提供し、TGS は TGT の承認データを基にしてプリンシパルが接続先にアクセスしたことを検証します。成功すると、TGS はクライアントと宛先サーバー (保護されたサービス/リソースを含むサーバー) の両方のクライアント (プリンシパル) に ST を発行し、クライアントが宛先サーバーにアクセスできるようにします。また、クライアントによってキャッシュされ、クライアントとサーバーの両方が読み取り可能な ST には、クライアントとサーバーがセキュアに通信できるようにするセッション情報も含まれています。



注記

Kerberos とネットワークの DNS 設定には密な関係があります。たとえば、実行しているホストの名前を基にクライアントが KDC にアクセスする場合、仮定を行います。そのため、Kerberos 設定だけでなくすべての DNS 設定を適切に行い、クライアントが接続できるようにすることが重要になります。

1.3. その他のコンポーネント

JBoss EAP 6 で Kerberos SSO を有効にするには、Kerberos コンポーネントの他にも複数の項目が必要になります。

1.3.1. SPNEGO

SPNEGO (Simple and Protected GSS_API Negotiation Mechanism)は Web アプリケーションで使用するために Kerberos ベースのシングルサインオン環境を拡張するメカニズムを提供します。

SPNEGO は、クライアントアプリケーションによって使用される認証方法で、クライアントアプリケーション自体をサーバーに対して認証します。この技術は、相互の通信を試みるクライアントアプリケーションとサーバーがお互いの認証プロトコルを把握していない場合に使用されます。SPNEGO は、クライアントアプリケーションとサーバー間の共通の GSSAPI メカニズムを判断し、その後のセキュリティー操作をすべてディスパッチします。

web ブラウザーなどのクライアントコンピューター上のアプリケーションが web サーバーの保護されたページにアクセスしようとすると、サーバーは承認が必要であることを伝えます。その後、アプリケーションは Kerberos KDC からのサービスチケットを要求します。チケットの取得後、アプリケーションはこのチケットをSPNEGO 向けにフォーマットされたリクエストにラップし、ブラウザーを介して web アプリケーションに返信します。デプロイされた web アプリケーションを実行している Web コンテナーはリクエストをアンパックし、チケットを認証します。認証に成功するとアクセスが許可されます。

SPNEGO は、Red Hat Enterprise Linux に含まれる Kerberos サービスや Microsoft Active Directory には不可欠な Kerberos サーバーなど、全タイプの Kerberos プロバイダーと動作します。

1.3.2. JBoss Negotiation

JBoss Negotiation は、JBoss EAP 6 に同梱されるフレームワークで、オーセンティケーターと JAAS ログインモジュールを提供し、JBoss EAP 6 で SPNEGO をサポートします。JAAS ログインモジュールの詳細は、Red Hat JBoss Enterprise Application Platform 6『セキュリティーアーキテクチャー』ガイドの「宣言型セキュリティーと JAAS」および「セキュリティードメイン」を参照してください。

注記

JBoss Negotiation を使用して、REST web サービスなどの一部のアプリケーションをセキュアにする場合、1 つまたは複数のセッションが作成され、クライアントがリクエストを行うと、それらのセッションがタイムアウト期間 (デフォルトでは 30 分) 開かれることがあります。Basic 認証を使用してアプリケーションをセキュアにした場合はセッションを開いたままにしないことが想定されるため、Basic 認証で想定される動作とは異なります。JBoss Negotiation はセッションを使用してネゴシエーション/接続の状態を維持するよう実装されるため、このようなセッションの作成は想定される動作です。

1.4. KERBEROS の統合

Kerberos は、Red Hat Enterprise Linux などの Linux ディストリビューションを含む多くのオペレーティングシステムと統合されています。また、Kerberos は Microsoft Active Directory には不可欠で、Red Hat Directory Server および Red Hat IDM によってサポートされます。

1.5. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM における KERBEROS による SSO の提供方法

Kerberos は、クライアントおよびサーバーによって使用されるチケットを KDC から発行してデスクトップベースの SSO を提供します。JBoss EAP 6 はこれと同じチケットを独自の認証および承認プロセスで使用して、既存のプロセスと統合することができます。JBoss EAP 6 がどのようにこのチケットを再使用するかを理解する前に、チケットがどのように発行され、JBoss EAP 6 がないデスクトップベースの SSO で認証および承認がどのように Kerberos と動作するかを理解することが重要です。

1.5.1. デスクトップベース SSO での Kerberos による認証および承認

Kerberos は認証および承認を提供するため、サードパーティーの KDC に依存してクライアント（プリンシパル）がサーバーにアクセスするサーバー（セキュアなリソース/サービス）の認証および承認決定を提供します。決定は 3 つの手順で行われます。

1. 認証エクスチェンジ
2. チケットの付与 (承認) エクスチェンジ
3. サーバーへのアクセス

1.認証エクスチェンジ

プリンシパルが最初にネットワークにアクセスする場合またはチケット保証チケット (TGT) なしでセキュアなサービスにアクセスする場合、プリンシパルのクレデンシャルを使用して認証サービス (AS) に対して認証することが要求されます。AS はユーザー提供のクレデンシャルを設定済みのアイデンティティーストアと照合します。認証に成功したら、クライアントによってキャッシュされる TGT がプリンシパルに発行されます。TGT には一部のセッション情報も含まれているため、クライアントと KDC の今後の通信も保証されます。

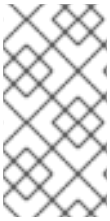
2.チケットの付与 (承認) エクスチェンジ

プリンシパルに TGT が発行されたら、プリンシパルはセキュアなサービスまたはリソースにアクセスします。プリンシパルはチケット保証サービス (TGS) にリクエストを送信し、KDC によって発行された TGT を渡し、特定の接続先 (セキュアなリソース/サービス) に対するサービスチケット (ST) を要求します。TGS はプリンシパルが提供した TGT をチェックし、要求したリソースにアクセスできる適切な権限があることを検証します。検証に成功したら、TGS はプリンシパルがその特定の接続先にアクセスするための ST を発行します。また、TGS はクライアントと接続先サーバーの両方に対してセッション情報を作成し、クライアントと接続先サーバー間のセキュアな接続を可能にします。このセッション

情報は、クライアントとサーバーが、以前のトランザクションで KDC によって個別に提供された長期キーを使用して、独自のセッション情報のみを復号化できるよう個別に暗号化されます。その後、TGS はクライアントとサーバー両方のセッション情報が含まれる ST でクライアントに応答します。

3.サーバーへのアクセス

これで、プリンシパルはセキュアなサービスの ST と、サーバーとセキュアに通信できるメカニズムを取得したため、クライアントは接続を確立し、セキュアなリソースへアクセスできます。クライアントは、宛先サーバーに、その宛先の TGS から受信した ST(セッション情報のサーバーコンポーネントも含む) を渡すことから開始します。サーバーは、KDC からの長期鍵を使用して、クライアントが渡したセッション情報を復号化します。復号化できたら、クライアントはサーバーに対して認証され、サーバーもクライアントに対して認証されたことになります。この時点で信頼が確立され、クライアントとサーバー間でセキュアな通信が行えます。



注記

承認されていないプリンシパルは TGT を使用できませんが、最初に AS で認証に成功した後でのみプリンシパルに TGT が発行されます。これにより、適切に承認されたプリンシパルのみに TGT が発行されるだけでなく、承認されていない第三者がオフラインの辞書攻撃や総当り攻撃などの不正使用が目的で、TGT を取得する可能性が低減されます。

1.5.2. Kerberos および Red Hat JBoss Enterprise Application Platform 6

JBoss EAP 6 を既存の Kerberos デスクトップベースの SSO 環境と統合すると、同じチケットで JBoss EAP 6 インスタンス上でホストされる web アプリケーションへのアクセスを提供することが可能です。一般的な設定では、Kerberos および SPNEGO セキュリティードメインが割り当てられるように JBoss EAP 6 インスタンスが設定されます。これらのセキュリティードメインと JBoss Negotiation を使用するよう設定されたアプリケーションは、その JBoss EAP 6 インスタンスにデプロイされます。ユーザーは、Kerberos によって保護されたデスクトップにログインし、KDC と認証の交換を完了します。その後ユーザーは、JBoss EAP 6 インスタンスが直接 web ブラウザーを使用するデプロイされたアプリケーションのセキュアなリソースにアクセスしようとします。JBoss EAP 6 は、セキュアなリソースへのアクセスには承認が必要であることを伝えます。web ブラウザーはユーザーの TGT チケットを取得し、チケットの許可 (承認) を行い、KDC と交換してユーザーを検証し、サービスチケットを取得します。ST がブラウザーに返されたら、SPNEGO 用にフォーマットされたリクエストに ST をラップし、JBoss EAP 6 上で実行している web アプリケーションに返信します。その後、JBoss EAP 6 は SPNEGO 要求を展開し、設定されたセキュリティードメインと JBoss Negotiation を使用して認証を実行します。認証に成功すると、ユーザーはセキュアなリソースへのアクセスが許可されます。

第2章 KERBEROS を使用した RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 6 での SSO の設定方法

本セクションでは、SSO に Kerberos を使用するように JBoss EAP 6 とデプロイされたアプリケーションを設定する方法を説明します。

2.1. コンポーネント

Kerberos を使用して JBoss EAP 6 に SSO を設定するには、以下のコンポーネントが必要です。

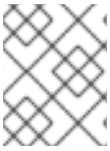
- 適切に設定された Kerberos 環境
- JBoss EAP 6 インスタンス
- Web アプリケーション

2.1.1. JBoss Negotiation Toolkit

JBoss Negotiation Toolkit は、https://community.jboss.org/servlet/JiveServlet/download/16876-2-34629/jboss-negotiation-toolkit.war からダウンロードできるデバッグツールです。これは、アプリケーションを実稼働環境にデプロイする前に認証メカニズムのデバッグおよびテストに役立つ追加ツールとして提供されます。これはサポート対象のツールではありませんが、SPENEGO を web アプリケーションに対して設定するのは難しいこともあるため、大変便利なツールと言えます。

2.1.2. Kerberos 環境

前のセクション で説明したとおり、Kerberos はサードパーティー (KDC) に依存して認証および承認を決定します。これには、KDC との認証にブラウザーなどのクライアントとそれらのホストが適切に設定されている必要もあります。本書では主に JBoss EAP 6 とホストされる web アプリケーションを中心に取り上げるため、KDC および Kerberos ドメインの設定については本書の範囲外となります。



注記

これ以降の項では、KDC と Kerberos ドメインがすでにセットアップされ、適切に設定されていることを仮定します。

2.1.3. Red Hat JBoss Enterprise Application Platform のこれまでのバージョンとの相違点

JBoss EAP 6 と、これまでのバージョンには顕著な違いがいくつかあります。

- セキュリティードメインは、管理対象ドメインの各プロファイル、またはスタンドアロンサーバーごとに設定されます。これらはデプロイメント自体には含まれません。デプロイメントが使用するセキュリティードメインは、デプロイメントの **jboss-web.xml** ファイルまたは **jboss-ejb3.xml** ファイルで指定されます。
- セキュリティープロパティーはセキュリティードメインの一部として設定されます。これらはデプロイメントには含まれません。
- オーセンティケーターはデプロイメントの一部として上書きできなくなりました。ただし、NegotiationAuthenticator バルブを **jboss-web.xml** 記述子に追加して同じ効果を得ることができます。バルブを使用するには、**<security-constraint>** および **<login-config>** 要素を

web.xml に定義する必要があります。これらの要素は、セキュアなリソースを決定するのに使用されます。しかし、選択した **auth-method** は **jboss-web.xml** の NegotiationAuthenticator バルブによって上書きされます。

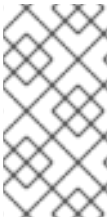
- セキュリティードメインの **CODE** 属性は、完全修飾クラス名ではなく、単純な名前を使用するようになりました。以下の表は、JBoss Negotiation に使用されるクラスとそれらのクラスとの間のマッピングを示しています。

簡単な名前	クラス名	目的
Kerberos	com.sun.security.auth.module.Krb5LoginModule	Oracle JDK を使用する場合の Kerberos ログインモジュール
Kerberos	com.ibm.security.auth.module.Krb5LoginModule	IBM JDK を使用する場合の Kerberos ログインモジュール
SPNEGO	org.jboss.security.negotiation.spnego.SPNEGOLoginModule	Web アプリケーションが Kerberos 認証サーバーに対して認証できるようにするメカニズム。
AdvancedLdap	org.jboss.security.negotiation.AdvancedLdapLoginModule	Microsoft Active Directory 以外の LDAP サーバーで使用されます。
AdvancedAdLdap	org.jboss.security.negotiation.AdvancedADLoginModule	Microsoft Active Directory LDAP サーバーで使用されます。

2.1.4. Red Hat JBoss Enterprise Application Platform 6 インスタンスの設定

JBoss EAP 6 には、デプロイされたアプリケーションと SSO に SPNEGO および JBoss Negotiation を使用して Kerberos を使用するのに必要なすべてのコンポーネントが含まれていますが、以下の設定を変更する必要があります。

1. サーバーアイデンティティー（ホスト）セキュリティードメインの設定
2. Web アプリケーションセキュリティードメインの設定
3. 関連するシステムプロパティーの設定



注記

以下の CLI コマンドは、JBoss EAP 6 のスタンドアロンインスタンスを前提としています。JBoss EAP 6 ドメインでの CLI の使用の詳細は、Red Hat JBoss Enterprise Application Platform 6 Administration and Configuration Guide の **The Management CLI** セクションを参照してください。

2.1.4.1. 1.サーバーアイデンティティー（ホスト）セキュリティードメインの設定

このセキュリティードメインは、コンテナー自体を KDC へ認証します。ユーザーではなくコンテナー自体の認証であるため、静的ログインメカニズムを受容するログインモジュールを使用する必要があります。以下の例では静的プリンシパルを使用し、クレデンシャルが含まれるキータブファイルを参照します。

サーバーアイデンティティーセキュリティードメインを作成する CLI の例

```
/subsystem=security/security-domain=host:add(cache-type=default)
```

```
/subsystem=security/security-domain=host/authentication=classic:add
```

```
/subsystem=security/security-domain=host/authentication=classic/login-module=Kerberos:add( \
  code=Kerberos, \
  flag=required, \
  module-options=[ \
    ("storeKey"=>"true"), \
    ("refreshKrb5Config"=>"true"), \
    ("useKeyTab"=>"true"), \
    ("principal"=>"host/testserver@MY_REALM"), \
    ("keyTab"=>"/home/username/service.keytab"), \
    ("doNotPrompt"=>"true"), \
    ("debug"=>"false") \
  ])
```

```
reload
```

結果の XML

```
<security-domain name="host" cache-type="default">
  <authentication>
    <login-module code="Kerberos" flag="required">
      <module-option name="storeKey" value="true"/>
      <module-option name="useKeyTab" value="true"/>
      <module-option name="principal" value="host/testserver@MY_REALM"/>
      <module-option name="keyTab" value="/home/username/service.keytab"/>
      <module-option name="doNotPrompt" value="true"/>
      <module-option name="debug" value="false"/>
    </login-module>
  </authentication>
</security-domain>
```

IBM JDK を使用している場合、Kerberos モジュールのオプションは異なります。**jboss.security.disable.secdomain.option** システムプロパティーは **true** に設定する必要があります（Relevant システムプロパティーの設定 を参照してください）。さらに、ログインモジュールを以下のように更新する必要があります。

IBM JDK の例

```
<security-domain name="host" cache-type="default">
  <authentication>
    <login-module code="Kerberos" flag="required">
      <module-option name="principal" value="HTTP/testserver@MY_REALM"/>
      <module-option name="credsType" value="acceptor"/>
      <module-option name="useKeytab" value="file:///root/keytab"/>
    </login-module>
  </authentication>
</security-domain>
```

Kerberos ログインモジュールの設定オプションの完全リストは『Red Hat JBoss Enterprise Application Platform 6 セキュリティーガイド』 を参照してください。

2.1.4.2. 2.Web アプリケーションセキュリティードメインの設定

web アプリケーションセキュリティードメインは、個別のユーザーを KDC に対して認証するために使用されます。ユーザーの認証には少なくとも 1 つのログインモジュールが必要であり、ユーザーに適用するロールを検索する方法が必要です。後者は、手動でユーザーをロールにマップする **<mapping>** を追加したり、ユーザーをロールにマップする 2 つ目のログインモジュールを追加するなど、複数の方法で実現できます。

以下は、web アプリケーションセキュリティードメインの例になります。

サーバーアイデンティティーセキュリティードメインを作成する CLI の例

```
/subsystem=security/security-domain=app-spnego:add(cache-type=default)
```

```
/subsystem=security/security-domain=app-spnego/authentication=classic:add
```

```
/subsystem=security/security-domain=app-spnego/authentication=classic/login-
module=SPNEGO:add( \
  code=SPNEGO, \
  flag=required, \
  module-options=[ \
    ("serverSecurityDomain"=>"host") \
  ])
```

```
reload
```

結果の XML

```
<security-domain name="app-spnego" cache-type="default">
  <authentication>
    <!-- Check the username and password -->
    <login-module code="SPNEGO"  flag="required">
      <module-option name="serverSecurityDomain" value="host"/>
    </login-module>
    <!-- Second login module to search for roles -->
  </authentication>
  <!-- Alternatively, a 'mapping' element may be added instead of a second login module to map users
to roles-->
</security-domain>
```

SPNEGO ログインモジュールの設定オプションの完全な一覧は、『Red Hat JBoss Enterprise Application Platform 6 セキュリティーガイド』 を参照してください。

2.1.4.3. 3.関連するシステムプロパティーの設定

JBoss EAP 6 は、Kerberos サーバーへの接続に関連するシステムプロパティーを設定する機能を提供します。KDC、Kerberos ドメイン、および ネットワーク設定に応じて、以下のシステムプロパティーが必要 (または不必要) になります。

```
<system-properties>
  <property name="java.security.krb5.kdc" value="mykdc.mydomain"/>
  <property name="java.security.krb5.realm" value="MY_REALM"/>
  <property name="java.security.krb5.conf" value="/path/to/krb5.conf"/>
  <property name="jboss.security.disable.secdomain.option" value="true"/>
  <property name="sun.security.krb5.debug" value="false"/>
</system-properties>
```

値	説明
java.security.krb5.kdc	KDC のホスト名。
java.security.krb5.realm	レルムの名前
java.security.krb5.conf	設定 **krb5.conf** ファイルへのパス。
jboss.security.disable.secdomain.option	**true** に設定すると、**jboss.security.security_domain** ログインモジュールオプションのセキュリティードメインで宣言されたログイモジュールへの自動追加が無効になります。IBM JDK を使用する場合は **true** に設定する必要があります。
sun.security.krb5.debug	**true** の場合、デバッグモードが有効になります。

注記

デフォルトでは、セキュリティードメインに定義された各ログインモジュールに自動的に **jboss.security.security_domain** モジュールオプションが追加されている必要があります。このオプションは、既知のオプションのみが定義されるようにチェックを行うログインモジュールでは問題が発生します。IBM Kerberos ログインモジュールである **com.ibm.security.auth.module.Krb5LoginModule** はこのようなログインモジュールの 1 つです。このモジュールオプションを追加する動作を無効にするには、JBoss EAP 6 の起動時に **jboss.security.disable.secdomain.option** システムプロパティーを **true** に設定します。これを行うには、管理 CLI または管理コンソールを使用して **<system-properties>** を設定するか、**-Djboss.security.disable.secdomain.option=true** を起動パラメーターに追加します。

システムプロパティーの設定に関する詳細は、『Red Hat JBoss Enterprise Application Platform 6 管理ガイド』の「管理 CLI を使用したシステムプロパティーの設定」を参照してください。

2.1.5. Web アプリケーションの設定

セキュリティードメインが設定されたら、Kerberos 認証を有効にするために、設定したセキュリティードメインを使用するよう web アプリケーションを設定する必要があります。アプリケーションに以下の更新を行う必要があります。

1. SPNEGO 認証メソッドを使用する web.xml の設定
2. 設定されたセキュリティードメインを使用する jboss-web.xml の設定
3. JBoss Negotiation 依存関係のデプロイメントへの追加

アプリケーションを変更した後、JBoss EAP 6 インスタンスにデプロイして認証に Kerberos を使用することができます。

2.1.5.1. 1.SPNEGO 認証メソッドを使用する web.xml の設定

web.xml ファイルには以下が含まれる必要があります。

- セキュアな領域の URL パターンにマップする **<url-pattern>** が含まれる **<web-resource-collection>** を持つ **<security-constraint>**。任意で、**<security-constraint>** に許可されるロールを明記する **<auth-constraint>** を含めることもできます。
- **<auth-constraint>** にロールが指定されている場合、これらのロールを **<security-role>** で定義する必要があります。

<security-constraint> および **<security-role>** 要素を使用すると、管理者は URL パターンおよびロールを基に制限された領域または無制限の領域をセットアップできます。これにより、リソースをセキュリティーで保護することができ、保護しないこともできます。

例: web.xml ファイル

```
<web-app>
  <display-name>App1</display-name>
  <description>App1</description>
  <!-- Define a security constraint that requires the All role to access resources -->
  <security-constraint>
    <display-name>Security Constraint on Conversation</display-name>
    <web-resource-collection>
      <web-resource-name>exampleWebApp</web-resource-name>
      <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
      <role-name>All</role-name>
    </auth-constraint>
  </security-constraint>
<!-- Define the Login Configuration for this Application -->
  <login-config>
    <auth-method>SPNEGO</auth-method>
    <realm-name>SPNEGO</realm-name>
  </login-config>
<!-- Security roles referenced by this web application -->
  <security-role>
    <description>Role required to log in to the Application</description>
    <role-name>All</role-name>
  </security-role>
</web-app>
```

2.1.5.2. 2.設定されたセキュリティードメインを使用する jboss-web.xml の設定

jboss-web.xml ファイルには以下が必要です。

- 認証または承認に使用されるセキュリティードメインを指定する **<security-domain>**。
- NegotiationAuthenticator バルブクラスを使用するように設定した **<valve>** (**org.jboss.security.negotiation.NegotiationAuthenticator**)。

- **任意**: 複数のロール名と照合するために **web.xml** の role-name 要素でアスタリスクの使用を有効にする <jacc-star-role-allow>。

例: jboss-web.xml ファイル:

```
<jboss-web>
  <security-domain>app-spnego</security-domain>
  <valve>
    <class-name>org.jboss.security.negotiation.NegotiationAuthenticator</class-name>
  </valve>
  <jacc-star-role-allow>true</jacc-star-role-allow>
</jboss-web>
```

2.1.5.3. 3.JBoss Negotiation 依存関係のデプロイメントへの追加

SPNEGO および JBoss Negotiation を使用する web アプリケーションでは、JBoss Negotiation クラスが見つかるようにするため、依存関係を **jboss-deployment-structure.xml** に定義する必要があります。JBoss EAP 6 は必要なすべての JBoss Negotiation と関連クラスを提供するため、アプリケーションはこれらの依存関係を宣言して使用することのみが必要となります。

jboss-deployment-structure.xml を使用した依存関係の宣言

```
<jboss-deployment-structure>
  <deployment>
    <dependencies>
      <module name="org.jboss.security.negotiation"/>
    </dependencies>
  </deployment>
</jboss-deployment-structure>
```

この代わりに、依存関係を **META-INF/MANIFEST.MF** ファイルに定義することもできます。

META-INF/MANIFEST.MF を使用した依存関係の宣言

```
Manifest-Version: 1.0
Build-Jdk: 1.6.0_24
Dependencies: org.jboss.security.negotiation
```

2.2. ACTIVE DIRECTORY に関する注意点

本項では、Active Directory ドメインの一部である Microsoft Windows サーバー上で JBoss EAP 6 が実行されている場合に、JBoss Negotiation の使用で必要となるアカウントの設定方法を説明します。

ここでは、サーバーへのアクセスに使用されるホスト名は **HOSTNAME**、レルムは **REALM**、ドメインは **DOMAIN**、JBoss EAP 6 インスタンスをホストするサーバーは **MACHINE_NAME** を使用します。

2.2.1. Microsoft Windows ドメインでの JBoss Negotiation の設定

2.2.1.1. 1.既存のサービスプリンシパルマッピング消去

Microsoft Windows ネットワークでは、一部のマッピングが自動作成されます。自動的に作成されたマッピングを削除し、ネゴシエーションが適切に行われるようにサーバーのアイデンティティーをサービスプリンシパルへマップします。マッピングにより、クライアントコンピューター上の Web ブラウ

ザーがサーバーを信頼し、SPNEGO の実行を試みます。クライアントコンピューターは、HTTP/**HOSTNAME** 形式のマッピングに対し、ドメインコントローラーを検証します。

既存のマッピングを削除する手順は次のとおりです。

以下のコマンドを使用して、コンピューターに対してドメインに登録されたマッピングをリストします。

```
setspn -L MACHINE_NAME
```

以下のコマンドを使用して、既存のマッピングを削除します。

```
setspn -D HTTP/HOST_NAME MACHINE_NAME
```

```
setspn -D host/HOSTNAME MACHINE_NAME
```

2.2.1.2. 2.ホストユーザーアカウントを作成します。

注記

ホスト名には **MACHINE_NAME** 以外の名前を使用してください。

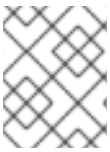
これ以降では、ホストユーザー名として **USER_NAME** を使用します。

2.2.1.3. 3.USER_NAME と HOST_NAME との間のマッピングを定義します。

以下のコマンドを実行して、サービスプリンシパルマッピングを設定します。

```
ktpass -princ HTTP/HOSTNAME@REALM -pass * -mapuser DOMAIN\USER_NAME
```

プロンプトが表示されたら、ユーザー名のパスワードを入力します。



注記

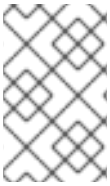
キータブのエクスポートの前提条件として、ユーザー名のパスワードをリセットします。

コマンド setspn -L **USER_NAME** を実行してマッピングを検証します。

2.2.1.4. 4.EAP JBoss がインストールされているサーバーに、ユーザーのキータブをエクスポートします。

以下のコマンドを実行してデータをエクスポートします。

```
ktab -k service.keytab -a HTTP/HOSTNAME@REALM
```



注記

このコマンドは、HTTP/**HOSTNAME** プリンシパルのチケットをキータブ service.keytab にエクスポートします。これは、JBoss EAP 6 でホストセキュリティードメインを設定するために使用されます。

2.2.1.5. 5.セキュリティードメイン内でプリンシパルを定義します。

プリンシパルは以下のようにセキュリティードメインで定義または更新できます。

```
<module-option name="principal">HTTP/HOSTNAME@REALM</module-option>
```

第3章 その他の機能

3.1. FORM ログインをフォールバックとして追加

JBoss EAP 6 およびそこにデプロイされたアプリケーションでは、FORM ログイン認証メカニズムを設定してフォールバックとして使用することもできます。これにより、アプリケーションは Kerberos/SPNEGO トークンが存在しない場合の認証でログインページを提示できます。この認証は、Kerberos認証とは独立して行われます。そのため、FORM ログインフォールバックの設定方法によっては、認証に個別のクレデンシャルが必要なことがあります。

FORM ログインをフォールバックとして設定するには、以下の手順を実行する必要があります。

1. Kerberos および SPNEGO を使用する Red Hat JBoss Enterprise Application Platform 6 と Web アプリケーションの設定
2. フォールバック認証用のセキュリティードメインの更新
3. ログインおよびエラーページの追加
4. web.xml の編集

注記

SPNEGO または NTLM トークンが存在しない場合、または存在する SPNEGO トークンが他の KDC からである場合に、FORM ログインへのフォールバックが利用できます。

3.1.1. 1.Kerberos および SPNEGO を使用する Red Hat JBoss Enterprise Application Platform 6 と Web アプリケーションの設定

認証および承認に Kerberos および SPNEGO を使用するように JBoss EAP 6 および Web アプリケーションを設定するときに必要な手順は、前述の項 を参照してください。

3.1.2. 2.フォールバック認証用のセキュリティードメインの更新

web アプリケーションセキュリティードメインを設定して、フォールバックログインメカニズムをサポートする必要があります。これには以下の手順が必要になります。

- フォールバック認証メソッドとして対応する新しいセキュリティードメインを追加します。
- **usernamePasswordDomain** モジュールオプションを、フォールバックドメインを示す web アプリケーションセキュリティードメインに追加します。

例: フォールバックセキュリティードメインで設定されたセキュリティードメイン

```
<security-domain name="app-spnego" cache-type="default">
  <authentication>
    <login-module code="SPNEGO" flag="requisite">
      <module-option name="password-stacking" value="useFirstPass"/>
      <module-option name="serverSecurityDomain" value="host"/>
      <module-option name="usernamePasswordDomain" value="app-fallback"/>
    </login-module>
    <!--login module for mapping roles -->
    <login-module code="UsersRoles" flag="required">
      <module-option name="password-stacking" value="useFirstPass"/>
```

```
      <module-option name="usersProperties"
        value="file:${jboss.server.config.dir}/users.properties"/>
      <module-option name="rolesProperties"
        value="file:${jboss.server.config.dir}/roles.properties"/>
    </login-module>
  </authentication>
</security-domain>
<security-domain name="app-fallback" cache-type="default">
  <authentication>
    <login-module code="UsersRoles" flag="required">
      <module-option name="usersProperties"
        value="file:${jboss.server.config.dir}/fallback-users.properties"/>
      <module-option name="rolesProperties"
        value="file:${jboss.server.config.dir}/fallback-roles.properties"/>
    </login-module>
  </authentication>
</security-domain>
```

3.1.3. 3.ログインおよびエラーページの追加

FORM ログインを使用するには、ログインおよびエラーページが必要です。これらのファイルは web アプリケーションに追加され、認証プロセスで使用されます。

例: login.jsp ファイル

```
<html>
<head></head>
<body>
  <form id="login_form" name="login_form" method="post"
  action="j_security_check" enctype="application/x-www-form-urlencoded">
    <center>
   <p>Please login to proceed.</p>
 </center>
 <div style="margin-left: 15px;">
   <p>
     <label for="username">Username</label>
 <br />
 <input id="username" type="text" name="j_username"/>
  </p>
  <p>
    <label for="password">Password</label>
 <br />
 <input id="password" type="password" name="j_password" value=""/>
  </p>
  <center>
    <input id="submit" type="submit" name="submit" value="Login"/>
  </center>
 </div>
  </form>
</body>
</html>
```

例: error.jsp ファイル

```
<html>
<head></head>
<body>
  <p>Login failed, please go back and try again.</p>
</body>
</html>
```

3.1.4. 4.web.xml の編集

ログインおよびエラーページを web アプリケーションに追加した後、**web.xml** を更新して FORM ログインでこれらのファイルが使用されるようにする必要があります。**<form-login-config>** 要素を **<login-config>** と、**<form-login-page>** と **<form-error-page>** 要素として指定されたログインおよびエラーページへのパスに追加する必要があります。

例: 更新された web.xml ファイル:

```
<web-app>
  <display-name>App1</display-name>
  <description>App1</description>
  <!-- Define a security constraint that requires the All role to access resources -->
  <security-constraint>
    <display-name>Security Constraint on Conversation</display-name>
    <web-resource-collection>
      <web-resource-name>examplesWebApp</web-resource-name>
      <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
      <role-name>All</role-name>
    </auth-constraint>
  </security-constraint>
<!-- Define the Login Configuration for this Application -->
  <login-config>
    <auth-method>SPNEGO</auth-method>
    <realm-name>SPNEGO</realm-name>
    <form-login-config>
      <form-login-page>/login.jsp</form-login-page>
      <form-error-page>/error.jsp</form-error-page>
    </form-login-config>
  </login-config>
<!-- Security roles referenced by this web application -->
  <security-role>
    <description> role required to log in to the Application</description>
    <role-name>All</role-name>
  </security-role>
</web-app>
```

3.2. KERBEROS での管理インターフェースのセキュア化

JBoss EAP 6 はセキュリティードメインで Kerberos 認証を提供する他に、Kerberos を使用して管理インターフェースをセキュアにする機能も提供します。管理インターフェイスで Kerberos 認証を有効にするには、以下の手順を実行する必要があります。

1. 補助システムプロパティーの有効化

2. Kerberos サーバーアイデンティティーのセキュリティーレルムへの追加
3. セキュリティーレルムでの認証方法の更新

注記

以下の CLI コマンドは、JBoss EAP 6 のスタンドアロンインスタンスを前提としています。JBoss EAP 6 ドメインでの CLI の使用の詳細は、Red Hat JBoss Enterprise Application Platform 6 Administration and Configuration Guide の **The Management CLI** セクションを参照してください。

3.2.1. 1.補助システムプロパティーの有効化

前の項 で説明したとおり、Kerberos サーバーへの接続に必要な JBoss EAP 6 システムプロパティーを有効にします。

3.2.2. 2.Kerberos サーバーアイデンティティーのセキュリティーレルムへの追加

Kerberos 認証をセキュリティーレルムで使用できるようにするには、Kerberos サーバーへの接続を追加する必要があります。以下の例は、Kerberos サーバーアイデンティティーを既存の管理レルムに追加する方法を表しています。

サーバーアイデンティティーをセキュリティーレルムに追加する CLI の例

```
/core-service=management/security-realm=ManagementRealm/server-identity=kerberos:add
```

```
/core-service=management/security-realm=ManagementRealm/server-identity=kerberos/ \
keytab=host\/testserver@MY_REALM:add( \
path=/home\/username\/service.keytab, \
debug=true)
```

```
reload
```

結果の XML

```
<security-realm name="ManagementRealm">
    <server-identities>
        <kerberos>
            <keytab principal="host/testserver@MY_REALM"
                    path="/home/username/service.keytab"
                    debug="true"/>
        </kerberos>
    </server-identities>
    ...
</security-realm>
```

3.2.3. 3.セキュリティーレルムでの認証方法の更新

Kerberos サーバーアイデンティティーが適切に設定された後、使用するためにはセキュリティーレルム認証メソッドを更新する必要があります。

Kerberos 認証をセキュリティーレルムに追加する CLI 例

```
/core-service=management/security-realm=ManagementRealm/authentication=kerberos:add
```

```
reload
```

結果の XML

```
<security-realm name="ManagementRealm">
    <server-identities>
        <kerberos>
            <keytab principal="host/testserver@MY_REALM"
                    path="/home/username/service.keytab"
                    debug="true"/>
        </kerberos>
    </server-identities>
    <authentication>
        <local default-user="$local" skip-group-loading="true"/>
        <kerberos/>
        <properties path="mgmt-users.properties" relative-to="jboss.server.config.dir"/>
    </authentication>
    ...
</security-realm>
```