記事
Toshihiko Minamoto · 2020年12月1日 6m read

SSH 接続する %Net.SSH.Session の使用とデバッグ方法

%Net.SSH.Session クラスを使用すると、SSH を使ってサーバーに接続することができます。
一般的にはSFTP、特に FTP インバウンドアダプタとFTPアウトバウンドアダプタで使用されています。

この記事では、簡単な例を示しながら、このクラスを使用して SSH サーバーに接続する方法、認証のオプションを記述する方法、そして問題が発生した場合のデバッグ方法について説明します。

次は接続を行う例です。

```
Set SSH = ##class(%Net.SSH.Session).%New()
Set return=SSH.Connect("ftp.intersystems.com")?
```

上記のコードは新しい接続を作成してから、ftp.intersystems.com の SFTP
サーバーにデフォルトのポートで接続します。 この時点で、クライアントとサーバーは暗号化アルゴリズムとオプションを選択済みですが、ユーザーはまだログインしていません。

接続したら、認証方法を選択できます。 選択できるメソッドには次の 3 つがあります。

- AuthenticateWithUsername
- AuthenticateWithKeyPair
- AuthenticateWithKeyboardInteractive

上記はそれぞれ異なる認証方式です。 各方式を簡単に説明します。

AuthenticateWithUsername

これは、ユーザー名とパスワードを使用します。

AuthenticateWithKeyPair

これは、公開鍵と秘密鍵のペアを使用します。
公開鍵は事前にサーバーに読み込まれている必要があり、それに一致する秘密鍵が必要となります。 秘密鍵がディスク上で暗号化されている場合、メソッドへの呼び出しで、それを復号化するためのパスフレーズを指定します。 注意: 秘密鍵を他人に送信してはいけません。

公開鍵は OpenSSH 形式であり、秘密鍵は PEM で暗号化されている必要があります。 OpenSSH
の形式は次のような書式です。

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCfi2Vq+u0rtt2OC84pyrkq1k7WkrS+s76u3a+2gdD43KQ2Z
3vSUUfksymJjp11JBZEpOtBVIAy221UKdc7j7Qk6sUjZaK8LIy+bzDVwMyFWgVvQge7EjdWjrJLBRCDXYML6y
1Y25XexThkTWSGyXzGNdr+wfIHYn/mIt0hfvrusauvT/9Wz8K2MGAj4BL7UQZpFJrlXzGmewe6++6cZDQQYi0
aztwLK798oc9j0LsccdMpqWrjqoU1uANFhYIuUu/T47TEhT+e6M+KFYK5TR998eJTO25IjdN2Tgw0feXhQFF/
nngbol0bA4auSPaZQsgokKK+E+Q/8UtBdetEofuV user@hostname
```

PEM で暗号化されている秘密鍵には、ファイルの上部に次のようなヘッダーがあります。

```
-----BEGIN RSA PRIVATE KEY-----
```

そして、最後に次の行があります。

```
-----END RSA PRIVATE KEY-----
```

AuthenticateWithKeyboardInteractive

これは、Cache 2018.1 以降で提供されている新しいオプションです。
チャレンジレスポンス認証を実行できます。 たとえば、テキストメッセージで送信されるか、Google 認証システムアプリで生成されたワンタイムパスワードを要求することがあるでしょう。 この認証方式を使用するには、サーバーが送信するプロンプトを処理するためのラムダ関数を記述する必要があります。

この認証方式を、ユーザーのパスワード認証と同じように、ユーザー名とパスワードのプロンプトのみで使用しているサーバーに遭遇することがあるかもしれませんが、 以下に説明する SSH デバッグフラグを使えば、これに遭遇しているかどうかを判定しやすくなります。

認証に関する注意事項: 1 つの接続で 2 つの認証方式の使用を検討している方は、Cache 2018.1 または InterSystems IRIS を使用してください。 これらのバージョンには、鍵ペアとユーザー名といった、複数の方式を使用できるようにするための更新があります。

問題が発生した場合

発生する可能性のある一般的なエラー

バナーの取得に失敗しました（Failed getting banner）

これは次のように表示されます。

```
ERROR #7500: SSH Connect Error '-2146430963': SSH Error [8010100D]: Failed getting ba
nner [FFFFFFFF8010100D] at Session.cpp:231,0
```

SSH クライアントが最初に行うのは、バナーの取得です。
このエラーが発生した場合、適切なサーバーに接続しており、それが SFTP サーバーであることを確認してください。

たとえば、サーバーが実際には FTPS サーバーであった場合に、このエラーが発生します。 FTPS サーバーは SSH ではなく SSL を使用するため、%Net.SSH.Session クラスでは動作しません。 FTPS サーバーに接続するには、%Net.FtpSession クラスを使用してください。

暗号化鍵を交換できません（Unable to exchange encryption keys）

このエラーは次のように表示されます。

```
ERROR #7500: SSH Connect Error '-2146430971': SSH Error [80101005]: Unable to exchang
e encryption keys [80101005] at Session.cpp:238,0
```

このエラーは通常、クライアントとサーバーの暗号化または MAC アルゴリズムが合致しなかったことを指しています。 これが発生した場合は、新しいアルゴリズムのサポートを追加するために、クライアントかサーバーのいずれかをアップグレードする必要があるかもしれません。

2017.1 より前のバージョンの Cache を使用している場合は、2017.1 以降を使用することをお勧めします。
libssh2 ライブラリは 2017.1 でアップグレードされており、新しいアルゴリズムがいくつか追加されています。

詳細については、以下に説明するデバッグフラグが提供するログを参照してください。

提供された公開鍵の署名が無効です（Invalid signature for supplied public key）

```
Error [80101013]: Invalid signature for supplied public key, or bad username/public k
ey combination [80101013] at Session.cpp:418
```

これは非常に誤解を招きやすいエラーです。 サーバーが 2 つの認証方式を必要としているにも関わらず、1 つしか提供しなかった場合に発生します。 この場合は、そのまま続けて次の方式を試しましょう！
このエラーがあっても、すべてうまく動作する可能性があります。

Error -37

エラー -37 に関するメッセージが表示されることがあります。 たとえば、次のデバッグログを見てください。

```
[libssh2] 0.369332 Failure Event: -37 - Failed getting banner
```

エラー -37 が示されている場合は必ず、失敗した操作が再試行されます。
このエラーが最終的な失敗の原因であることはありません。 ほかのエラーメッセージを確認してください。

SSH デバッグフラグ

接続に SSH デバッグフラグを使うと、SSH 接続の詳細なログを取得できます。
このフラグを有効にするには、SetTraceMethod メソッドを使います。
次に、このフラグを使った接続の例を示します。

```
Set SSH = ##class(%Net.SSH.Session).%New()
Do SSH.SetTraceMask(511,"/tmp/ssh.log")
Set Status=SSH.Connect("ftp.intersystems.com")?
```

SetTraceMask の最初の引数は、何を収集するかを指示します。 ビットの 10進表現です。 511 は 512 を除くすべてのビットを要求しており、最も一般的に使用される設定です。
各ビットに関する詳細については、%Net.SSH.Session クラスのクラスドキュメントをご覧ください。

2 つ目の引数は、接続に関するログ情報をどのファイルに格納するかを指示します。 この例では、/tmp/ssh.log ファイルを指定しましたが、任意の絶対または相対パスを使用できます。

上記の例では、Connect メソッドのみを実行しました。
認証に問題がある場合は、該当する認証方式も実行する必要があります。

テストを実行したら、ログファイルで情報を確認できます。 ログファイルの解釈に不安がある場合は、WRC をご覧ください。

#FTP #デバッグ #ベストプラクティス #Caché #Ensemble #InterSystems IRIS

ソースURL:
https://jp.community.intersystems.com/post/ssh-%E6%8E%A5%E7%B6%9A%E3%81%99%E3%82%8B-netsshse

ssion-%E3%81%AE%E4%BD%BF%E7%94%A8%E3%81%A8%E3%83%87%E3%83%90%E3%83%83%E3%82%B0%E6%96%B9%E6%B3%95