Artículo
Ricardo Paiva · 4 sep, 2020 Lectura de 5 min

Uso y depuración de %Net.SSH.Session para conexiones SSH

¡Hola desarrolladores!

La clase %Net.SSH.Session permite conectarse a servidores mediante SSH. Lo más habitual es usarlo con SFTP, especialmente en los adaptadores de FTP entrantes y salientes.

En este artículo se dará un breve ejemplo de cómo conectarse a un servidor SSH usando la clase, se describirá las opciones para autenticar y cómo hacer la depuración cuando surjan problemas.

A continuación un ejemplo de cómo hacer la conexión:

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

Esto crea una nueva conexión, y luego se conecta al servidor SFTP ftp.intersystems.com en el puerto predeterminado. En este punto, el cliente y el servidor han elegido opciones y algoritmos de cifrado, pero ningún usuario ha iniciado sesión aún.

Una vez conectado, podrá elegir cómo realizar la autenticación. Hay tres métodos principales para elegir:

- AuthenticateWithUsername
- AuthenticateWithKeyPair
- AuthenticateWithKeyboardInteractive

Cada uno de estos es un tipo distinto de autenticación. La siguiente es una breve introducción a cada tipo:

AuthenticateWithUsername

Esta usa un nombre de usuario y contraseña.

AuthenticateWithKeyPair

Esta usa un par de claves pública y privada. La clave pública se debe haber precargado en el servidor, y debe contar con la clave privada correspondiente. Si la clave privada está cifrada en el disco, debe introduzir una contraseña para descifrarla en la llamada al método. Nota: nunca envíe su clave privada a otra persona.

Las claves públicas deben estar en formato OpenSSH, y las claves privadas deben estar cifradas con PEM. El formato OpenSSH se ve así:

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

Las claves privadas cifradas con PEM tienen un encabezado en la parte superior del archivo que se ve así:

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

y terminan con:

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

AuthenticateWithKeyboardInteractive

Permite realizar una autenticación de desafío y respuesta. Por ejemplo, podría pedir el código de un uso enviado por mensaje de texto o generado por una aplicación autenticadora de Google. Para usar este tipo de autenticación, deberá escribir una función lambda para manejar la solicitudes de comandos enviadas por el servidor.
Puede que vea que algunos servidores usan esto con solo una solicitud de nombre de usuario y contraseña, de una forma que para el usuario se ve idéntica a una autenticación por contraseña. Las marcas de depuración SSH descritas a continuación pueden ayudarle a determinar si eso es lo que está viendo.
Un último comentario sobre la autenticación: Si le interesa usar dos formas de autenticación para una única conexión, asegúrese de usar Ensemble/Cache 2018.1+ o cualquier versión de InterSystems IRIS. Esta versión tiene actualizaciones que permiten el uso de múltiples formatos, tales como par de claves y nombre de usuario.

Qué hacer cuando algo sale mal...

Algunos errores comunes que podría encontrarse son:

Error al intentar obtener el banner

Esto podría verse así:

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

Obtener el banner es lo primero que hace un cliente SSH. Si ve este error, debería verificar que se está conectando al servidor correcto y que este es un servidor SFTP.

Por ejemplo: si el servidor es en realidad un servidor FTPS, verá este error. Los servidores FTPS usan SSL, no SSH, y por lo tanto no funcionan con la clase %Net.SSH.Session. Puede usar la clase %Net.FtpSession class para conectarse a un servidor FTPS.

No es posible intercambiar claves de cifrado

Este error podría verse así:

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

Este error generalmente significa que el cliente y el servidor no pudieron negociar algoritmos de MAC o cifrado. Si ve este error, puede que necesite actualizar ya sea el cliente o el servidor para agregar compatibilidad con nuevos algoritmos.

Si está usando una versión de Ensemble/Caché anterior a la 2017.1, le recomiendo actualizar a InterSystems IRIS

o probar con 2017.1 o posterior. La biblioteca libssh2 se actualizó en la versión 2017.1 y se agregaron múltiples algoritmos nuevos.

Puede ver más detalles en los registros provistos por las marcas de depuración que describo a continuación.

Firma inválida para clave pública suministrada

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

Este error podría ser fácil de malinterpretar. Verá este error si su servidor pidió dos formas de autenticación y usted solo facilitó una. Si ese es el caso, continue y pruebe con la próxima. Es posible que todo se arregle.

Error -37

Puede ver mensajes sobre el error -37. Por ejemplo, aquí está en el registro de depuración:

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

Siempre que aparezca el error -37, la operación que fracasó volverá a intentarse. Este error no es lo que causó la falla final. Busque otros mensajes de error.

Las marcas de depuración de SSH

Se puede habilitar el registro detallado de conexiones SSH para una conexión mediante las marcas de depuración de SSH. Las marcas se habilitan con el método SetTraceMethod. Este es un ejemplo de una conexión que las usa:

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

El primer argumento de SetTraceMask le indica qué recolectar. Es una representación decimal de bits. 511 solicita todos los bits excepto el 512, y es la configuración usada más comúnmente. Si desea conocer más acerca de cada bit, están enumerados en la documentación de la clase %Net.SSH.Session.

El segundo argumento le indica en qué archivo colocar la información de registro sobre la conexión. En este ejemplo usé el archivo /tmp/ssh.log, pero puede ingresar cualquier ruta absoluta o relativa que quiera usar.

En el ejemplo anterior, solo ejecuté el método Connect. Si su problema está en la autenticación, deberá ejecutar también el método de autenticación correspondiente.

Luego de ejecutar su prueba, podrá buscar información en el archivo de registro. Si no está seguro de cómo interpretar el archivo de registro, el Centro Mundial de Respuesta de Intersystems (WRC) puede ayudar.

#Depuración #FTP #Mejores prácticas #Caché #Ensemble #InterSystems IRIS