

HULFTの通信をよりセキュアに

HULFT と SSH Tectia を組み合わせたセキュアで強力 なファイル転送

Compatibility Note

2008年9月

株式会社セゾン情報システムズの企業内、企業間通信ミドルウェアである HULFT は、ファイル転送のアプリケーションとして、主に流通業、製造業で大きなシェ アを誇るパッケージソフトウェアです。

SSH Tectia ソリューションを HULFT と組み合わせることで、HULFT によるデータ通信をより強力なセキュリティで保護することができます。

本書では、SSH Tectia の透過的な TCP トンネリング機能を使い、HULFT の設定を変更 せずに強力な暗号化通信を実現する方法を紹介します。

© 2008 SSH Communications Security Corp. All rights reserved. ssh® is a registered trademark of SSH Communications Security Corp in the United States and in certain other jurisdictions. The SSH logo and Tectia are trademarks of SSH Communications Security Corp and may be registered in certain jurisdictions. All other names and marks are the property of their respective owners.

CONTENTS

С	ONTE	NTS	1
1	シナ	-リオ	2
			_
	1.1	はじめに	2
	1.1.	1 HULFTの主な機能	2
	1.1.2	2 <i>透過的TCPトンネリング</i>	2
	1.1	3 SSH Tectia でサポートしている暗号化アルゴリズム	2
	1.1.4	4 FIPSモードの場合の暗号化アルゴリズム	2
	1.2	利用シナリオ	3
	1.2.	1 利用イメージ	3
	1.3	ハードウェア/ソフトウェア	3
2	イン	/ストールガイド	5
	2.1	クライアント	5
	2.1.	1 クライアントのインストール	5
	2.2	サーバ	6
	2.2.	1 サーバのインストール	6
	2.2.2	2 サーバの設定	6
	2.2	3 サーバの動作確認	6
3	設定	<u> </u>	8
	3.1	鍵の作成とアップロード	8
	3.2	SSHのトンネリング設定	11
4	動作	戶確認	.14

1 シナリオ

1.1 はじめに

本書は、株式会社セゾン情報システムズの企業内、企業間通信ミドルウェアである HULFT と SSH Communications Security 社の SSH Tectia を組み合わせてセキュアで強力なファイル転送を 実現するための相互接続に関する資料となっています。

HULFTでは主に以下のような機能をサポートしており、ファイル転送のアプリケーションと しては導入企業数 5700 社、販売本数 105,000 本(2008 年 3 月時点)の実績を有し、主に流 通業、製造業で大きなシェアを誇るパッケージソフトウェアとなっています。

1.1.1 HULFTの主な機能

- 集配信機能、集配信管理機能
- アプリケーション連携機能、ファイル連携機能
- 非同期/同期転送機能
- リモートジョブ実行
- コード変換機能、データ圧縮機能
- メッセージ送信機能

1.1.2 透過的TCPトンネリング

SSH Tectia Client の機能の一つとして透過的 TCP トンネリング機能があります。これを使うこ とで上位のアプリケーションの設定を変更することなく、高度な暗号化通信を提供するこ とが可能で、安全なファイル転送を実現します。

1.1.3 SSH Tectiaでサポートしている暗号化アルゴリズム

SSH Tectia でサポートしている暗号化アルゴリズムは、以下のとおりです。

3des-cbc、aes128-cbc、aes192-cbc、aes256-cbc、arcfour、blowfish-cbc、twofish-cbc、

twofish128-cbc、twofish192-cbc、twofish256-cbc、crypticore128@ssh.com、seed-cbc@ssh.com

また、SSH Tectia では FIPS140-2 認定を受けた暗号化ライブラリをサポートしており、これを 使うことで、より高度なセキュリティ要件にも柔軟に対応することが可能です。

1.1.4 FIPSモードの場合の暗号化アルゴリズム

FIPS モードの場合の暗号化アルゴリズムは、以下のとおりです。

3des-cbc、aes128-cbc、aes192-cbc、aes256-cb

1.2 利用シナリオ

既に HULFT が導入されているファイル転送環境において、SSH Tectia を利用して強力な暗号 化通信を行うまでを想定します。

1.2.1 利用イメージ

各店舗からセンターに対して日々の売り上げデータなどが HULFT を使ってセンターに配信さ れているような環境において、店舗側 Windows 端末に Tectia Client、センター側 Unix やメイ ンフレームに Tectia Server を導入することで安全なファイル転送を実現します。



1.3 ハードウェア/ソフトウェア

本書は、以下の環境を想定しています。

	クライアント	サーバ
OS	Windows XP SP2	Red Hat Exnterprise Linux 4
ソフトウェア	SSH Tectia Client 6.0.1	SSH Tectia Server 6.0.1
HULFT ソフトウェア	HULFT for Windows Type WIN-CL Ver.06.03.03A	HULFT V06.03.02 Type-U1

HULFT を利用するにあたっては、以下の2点に注意する必要があります。

- 1. 配信側、集信側双方のマシンで HULFT ソフトウェアが必要です。
- 2. ホスト名を利用して通信を行うため、名前解決ができる必要があります。

2 インストールガイド

2.1 クライアント

2.1.1 クライアントのインストール

透過的 TCP トンネリングの機能を使用するためには SSH Tectia Client のインストーラを起動後、 セットアップタイプのメニューで Custom か Complete を選択する必要があります。

🛃 SSH Tectia Client Setup 🛛 🔀
Choose Setup Type Choose the setup type that best suits your needs
Image:
All program features will be installed. (Requires most disk space)
< <u>B</u> ack <u>N</u> ext > Cancel

Custom を選択した場合は、Transparent TCP tunneling (透過的 TCP トンネリング)の機能を有効にしてインストールを行ってください。



インストールは再起動を行うことで完了します。



2.2 サーバ

2.2.1 サーバのインストール

SSH Tectia Server のインストールは、必要な各パッケージをコマンドからインストールしま す。Red Hat Enterprise Linux の x86 用のインストールには、以下の 2 つのパッケージが必要で す。

- ssh-tectia-common-6.0.1.10-linux-x86.rpm
- ssh-tectia-server-6.0.1.10-linux-x86.rpm

rpm コマンドでパッケージをインストールした後、製品版ライセンスファイル (sts60.dat) を手動で/etc/ssh2/license ディレクトリにコピーする必要があります。

2.2.2 サーバの設定

/etc/ssh2 ディレクトリ配下には、いくつかのサンプル設定があります。

基本設定として ssh-server-config-default.xml がありますので、ssh-server-config.xml としてコ ピーし、これを編集して利用してください。デフォルトの内容で公開鍵認証が利用できる 状態なので特に変更することなく利用することが出来ます。

2.2.3 サーバの動作確認

SSH Tectia Server サービスの起動と停止はコマンドから行います。設定を変更した後は、必ずサービスの再起動を行ってください。

起動:/etc/init.d/ssh-server-g3 start

停止:/etc/init.d/ssh-server-g3 stop

サーバが動作しているかは、ps コマンドを利用して確認してください。

[root@redhat ~]# ps -ef g	rep tectia;		
root 2914 1 0 0	12:10 ? 00:00	:00 /opt/tectia/sbin/ssh-server-g3start-ser	
Vice	0.10 0 00.00	•00 /+/++:-/!:h/h+ -0	
start-service	12:10 : 00:00	.00 /opt/tectia/libexec/ssn-servant-gaslave	
root 2949 2914 0 0	12:10 ? 00:00	:00 /opt/tectia/libexec/ssh-servant-g3slave	
start-service			
root 2950 2914 0 0	12:10 ? 00:00	:UU /opt/tectia/libexec/ssh-servant-g3slave	
root 2951 2914 0 0	12:10 ? 00:00	:00 /opt/tectia/libexec/ssh-servant-g3slave	
start-service			
root 2952 2914 U U	12:10 7 00:00	:UU /opt/tectia/libexec/ssh-servant-g3slave	
root 4461 4137 0 0	12:28 pts/1 00:00	:00 grep tectia	
[root@redhat ~]#		···· •···	~

3 設定

3.1 鍵の生成とアップロード

タスクトレイにある SSH Tectia のアイコンを右クリックし、Configuration を選択します。

	- A second se Second second s Second second sec	
~	Transparent tunneling enabled	20.00
	Configuration	we they
	Status	
	About	
	Exit	
KONO	- 🖞 - Community -	99

User Authentication⇒Keys and Certificate を選択します。



New Key を選択して、鍵の生成を開始します。



鍵の長さは 768bit から 2048bit の範囲で生成が可能です。長ければ長いほどセキュリティの 強度が高くなります。

The recommended key length is 2048 bits.				
Key type:	DSA 💌			
Key length:	2048 💌			
Click Next t	o start the key generation process			

指定する鍵のファイル名は任意です。また、自動接続を行うため、<u>パスフレーズは設定し</u> <u>ないで</u>次へ進んでください。

Filename: 🤇	test-key
Comment:	
Passphrase:	
Retype passphrase:	

鍵の生成が終了したら接続先の SSH サーバへ公開鍵をアップロードします。

生成した鍵ファイルを選択し、Upload ボタンをクリックします。

Comment			Filename			
[2048-bit dsa, dit@V	VINXP, ? 4-30-20:3	3:07 2008]	C:¥Documents	and Settin	ngs¥dit¥A	Applica
•						•
٩ [•

Quick connect のラジオボタンを選択し、Host name に 192.168.1.100、User name に dit と入力 し、Upload をクリックします。

🐸 Upload Public Ke	y ?🛛
Upload your public key	to the remote host.
Destination host —	
	Host name 192.168.1.100
Quick connect	User name dit
	Port number 22
C Connection profile	Demo-profile 💌
Public key file	test-key.pub
Destination folder	.ssh2
Authorization file	authorization
View Authorization file	Γ
	Upload Cancel

パスワードを求められますのでパスワードを入力します。

🗟 SSH Tectia 🛛 💽 🗙				
dit@192.168.1.100's passv	word	*****		
		ОК	Cancel	

鍵のアップロードが成功すると下記のようなダイアログが表示されます。



3.2 SSHのトンネリング設定

HULFT の通信を SSH でトンネリングするために、Transparent TCP tunneling の設定を行います。

Configuration⇒Transparent Tunnels⇒Filter Rules を選択、Add をクリックします。

🕂 Transparent Tunnels		
Connection Capture		
Filter Rules		
ⁱ Automatic Tunnels		
	•	▶
	🔂 🦊	Add Edit Delete

Silter Rule	? 🔀
Application to Capture	
Tunnel all applications	
	Add
	Edit
1	Delete
Filter by Address	
 Any host or IP address 	
C Hostname	
C IP address	
Filter by Port	
Any port	
⊂ Single port	
C Port range -	
Action	
Connect directly	
C Block connection	
C Transparent TCP tunneling using Defau	ult Settings 💌
C Transparent FTP tunneling using Defau	ult Settings 💌
C FTP-SFTP conversion using Defau	ult Settings 💌
$\hfill \square$ Use user name from the application	
${\textstyle \fbox}$ Use host name from the application	
$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	nection fails
$\higside \begin{tabular}{c} \higside \begin \begi$	ietwork
ОК	Cancel

Filter Ruleの内容として以下を設定します。

- トンネル対象とするアプリケーションプログラム
- トンネル対象とする接続先ホスト
- トンネル対象とする通信のポート番号

ここでは HULFT の通信をトンネル対象とするため以下のように設定を行います。

- トンネル対象とするアプリケーション:
 C:\HULFT Family\hulft6\binnt\hulsdd.exe
- トンネル対象とする接続先ホスト: 192.168.1.100
- トンネル対象とするポート番号:
 30000

Application to Capture						
Tunnel all applications						
O##HULFT Family##hulft6##binnt##hulsdds Add						
Edit						
Delete						
Filter by Address						
○ Any host or IP address						
C Hostname						
© IP address 192.1681.100						
Filter by Port						
Any port						
© Single port 30000						
C Port range -						
Action						
C Connect directly						
C Block connection						

Action として、Transparent TCP tunneling を選択します。

Filter Rules						
Define filter rules for transparent TCP tunneling. Rules are read in the order shown and the first matching rule is used.						
Application Address Port Action Profi						
C:¥¥HULFT Family¥¥hulft6¥¥binnt¥¥hulsdd.exe	IP: 192.168.1.100	30000	TUNNEL	Defau		

設定が終了したら、Apply⇒OKをクリックして終了してください。

4 動作確認

実際に HULFT を使ってファイル転送を行い、通信が暗号化されるか確認をします。

スタートメニュー⇒プログラム⇒HULFT Family⇒HULFT for Windows Ver.6 から HULFT の管理画 面を選択して立ち上げます。



あらかじめ設定されている配信設定を確認します。



配信側設定

集信側設定

🥶 192.168.1.100 - Tera Term VT	
ファイル(E) 編集(E) 設定(S) コントロール(Q) ウィンドウ(W) ヘルプ(H)	
集信管理情報更新 ファイルID:TEST002 Ctr+p:管理情報額写 ESC :キャンセル Ctr+d:管理情報酮除 TAB : 項目移動	
ファイル名 : /usr/tmp/Data1.txt オーナー名 : root グループ名 : sys パーミッション : 読み 書き 実行 0:権限無し 1:権限有り オーナー 1 1 1 グループ 1 1 1 その他 1 1 1	
EBCDICコードセット: A (A:KN B:AL C:AC D:AP E:LOW F:EX G:NEC V:U1 W:U2 X:U3) 登録モード : R (N:新規作成 R:置き換え M:追加) 異常時の処置 : K (D:削除 K:保持 R:復元) 集信形態 : S (S:単一集信 M:複数集信) 世代管理 : N (N:無 Y:有) 世代管理数 : 0 正常時ジョブID : 異常時ジョブID : 転送グルーブID : grp02 暗号キー : 集信完了通知 : T (T:受信完了 J:正常時ジョブ完了) エクスチェンジャ : N (N:無 Y:有)	
コメント:	

要求発行から配信要求を選択し、redhat サーバ(グループ ID:grp02)の HULFT へ配信を行います。

配信要求		
77√ルIDΦ	TEST002	E∎配信(S)
優先度(P)		¥ ≠+>>セル(©)
		Q 参照(B)



要求発行が正常終了すると、同じタイミングでファイル転送が行われ、実際のファイル転送が SSH を経由して行われます。

ファイル転送中のステータスは SSH Tectia の Status ウィンドウで確認をすることができます。

2	SSH Tecti	a Status					
	(sshi te	ctia SSH Te	ectia	nte tra est			
		Connection	ID	Upload	Download	Upload Speed	Download Sp
	2	dit@192.168.1.100#22	13	12,679,414	62,057	393.6 kB/s	1.9 kB/s
	Connections	i≺local listener at 127.	0.0.1:0>14	12,679,414	62,057	393.6 kB/s	1.9 kB/s
	Keys						
	Logs						
	1	<u></u>					<u>C</u> lose