

「暗号化通信におけるリスク」

～ SSH に潜む落とし穴 ～

“暗号化すれば安全ですか？”

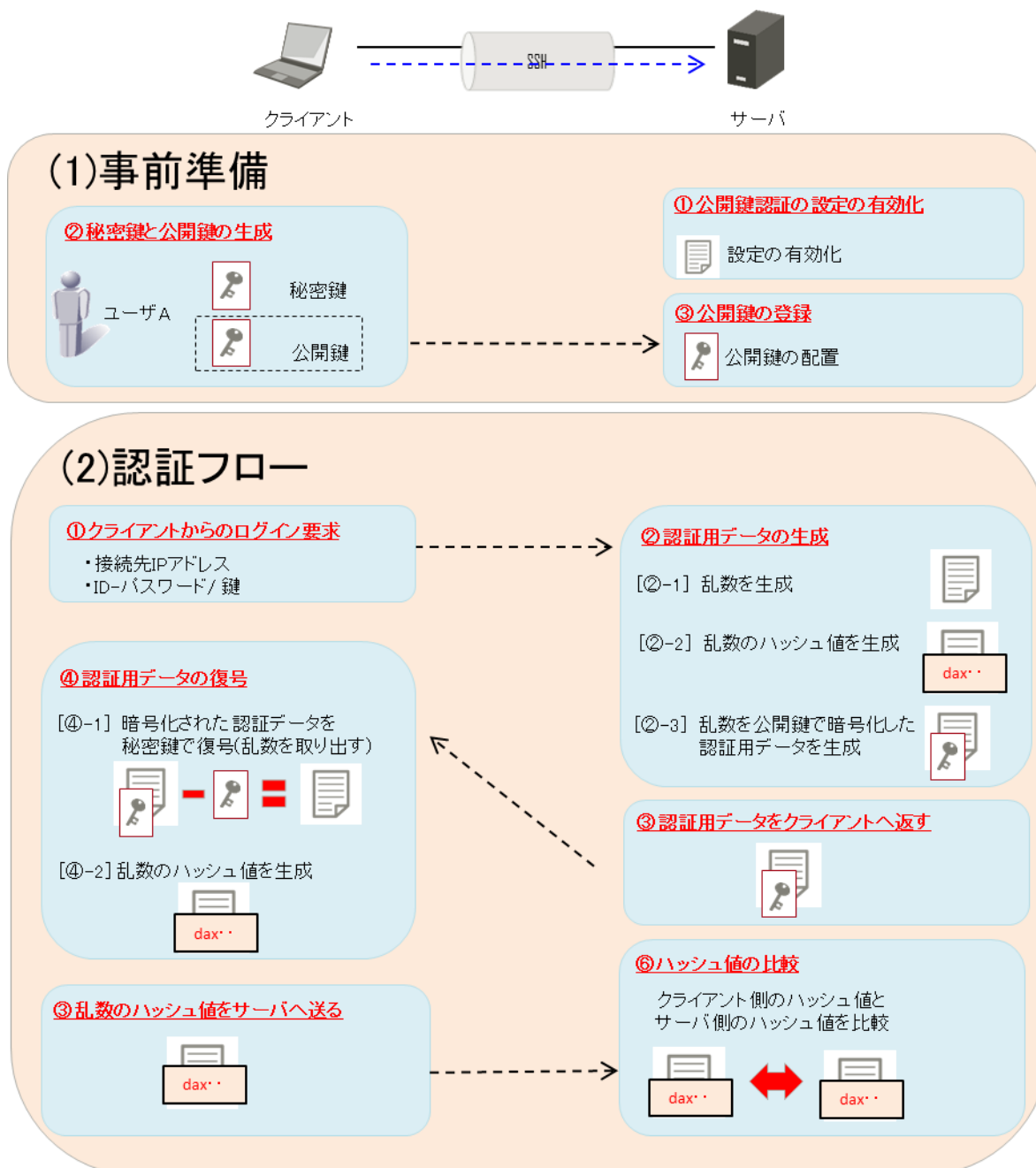
<付録. 1> SSH の公開鍵認証について

2015年09月

<付録.1> SSHの公開鍵認証について

1. SSHの公開鍵認証のシーケンス

SSHの公開鍵認証のシーケンスを説明する。



※ 公開鍵で暗号化された認証データを、復号できるのは秘密鍵となる。

※ ハッシュ値が一致すれば正規のユーザとしてアクセスが許可される。

<付録.1> SSHの公開鍵認証について

2. SSHの公開鍵認証の設定方法

本章では、SSHの公開鍵認証を安全に利用していただくための設定方法を記載する。

(1) 公開鍵と秘密鍵の生成

公開鍵ペア(秘密鍵と公開鍵)は、サーバ側で生成することが可能であるが、ここでは、SSHクライアントであるTera TermとTectia Clientを使用した生成方法を記載する。

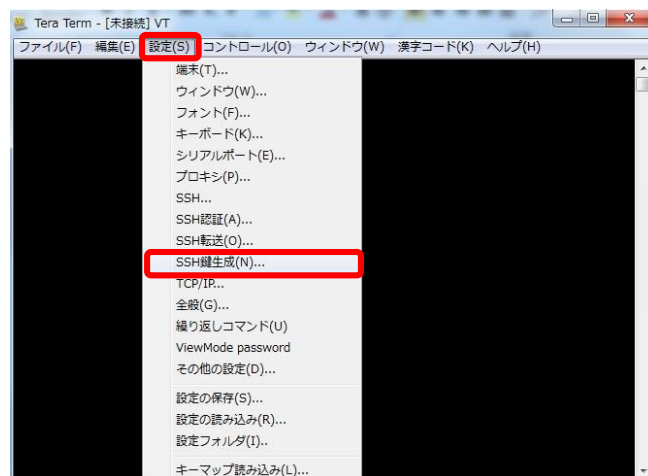
No	ソフトウェア	作成できる鍵の形式
1	TeraTerm	OpenSSH形式
2	Tectia Client	SECSH形式(ssh.com)形式

■TeraTermによる公開鍵ペア(秘密鍵と公開鍵)の生成

- ① TeraTermを起動する。
- ② [新しい接続]ダイアログが表示されるので、「キャンセル」ボタンを押す。

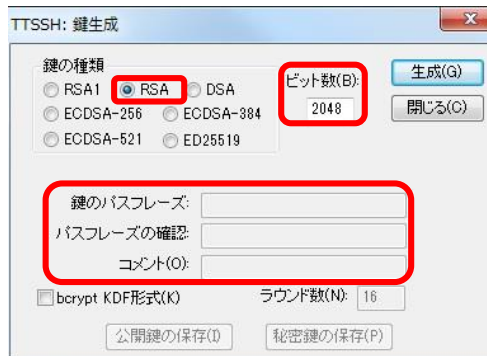


- ③ Tera Termの[設定]→[SSH鍵生成]をクリックする。



<付録. 1> SSHの公開鍵認証について

- ④ [鍵の種類]で作成する[鍵の暗号方式]を選択し、
[生成]ボタンをクリックする。
- [鍵の種類] : RSA
- [ビット数(B)] : 2048



- ⑤ [鍵のパスフレーズ]と[パスフレーズの確認]と[コメント]を入力する。
- [鍵のパスフレーズ] : [任意のパスフレーズ]
- [パスフレーズの確認] : [任意のパスフレーズ]
- [コメント] : [任意のコメント]

※ 鍵のパスフレーズを[無し]にした場合、秘密鍵があるクライアントから、パスフレーズ不要で、サーバへログインできるようになる。このため、秘密鍵の管理は、十分に注意して運用する必要がある。

- ⑥ [公開鍵の保存]ボタンをクリックし、鍵(id_rsa.pub)に任意の名前をつけ、任意の場所に保存する。
- ⑦ [秘密鍵の保存]ボタンをクリックし、鍵(id_rsa)に任意の名前をつけ、任意の場所に保存する。

※鍵のパスフレーズを[無し]にした場合「空のパスフレーズを使用しますか？」と警告画面がでるので、「はい」ボタンをクリックする。

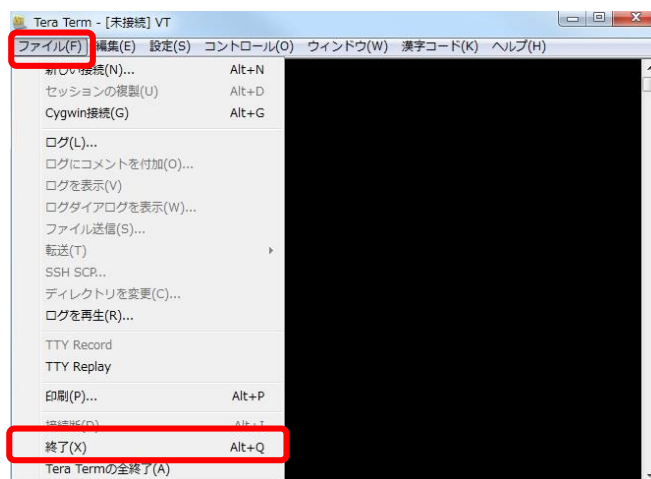


<付録.1> SSHの公開鍵認証について

- ⑧ [閉じる]ボタンをクリックし、[鍵生成]ウィンドウを閉じる。

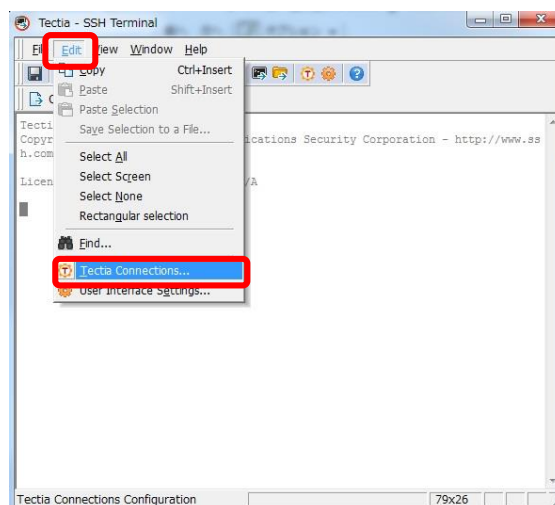


- ⑨ [ファイル]→[終了]をクリックし、[TeraTerm] ウィンドウを閉じる。



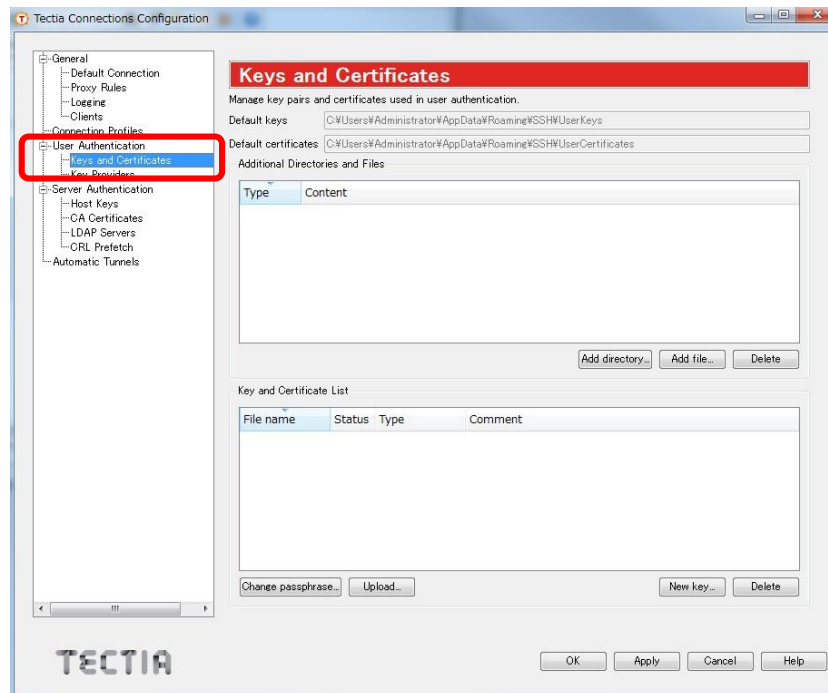
■Tectia Client による公開鍵ペア (秘密鍵と公開鍵)の作成

- ① Tectia Client を起動する。
- ② [Edit]→[Tectia Connections]をクリックする。

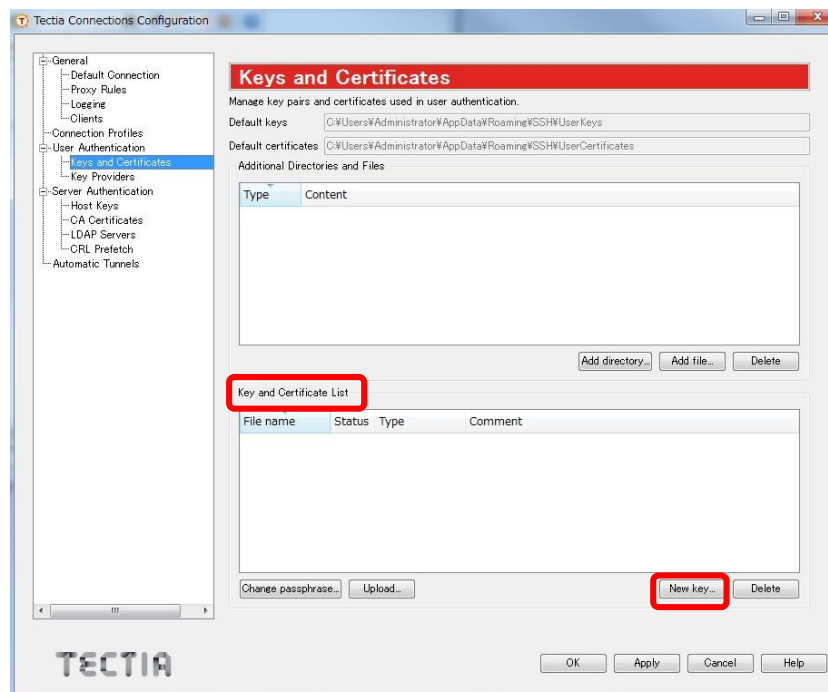


<付録. 1> SSHの公開鍵認証について

- ③ [User Authentication]→[Keys and Certificates]をクリックする。



- ④ [Key and Certificate List] → [New Key]ボタンをクリックする。

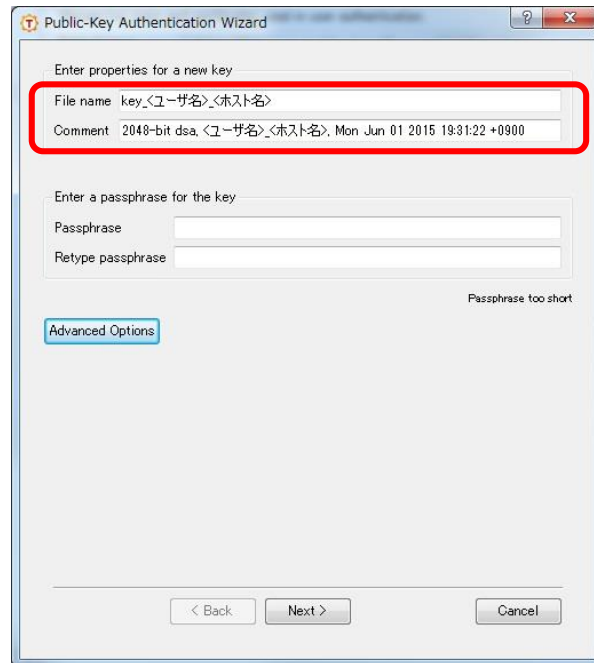


<付録. 1> SSHの公開鍵認証について

- ⑤ [Enter properties for a new key]の[File name]と[Comment]を入力する。

File name : 任意のファイル名 (Key_<ユーザ名>_<ホスト名>)

Comment : 任意のコメント

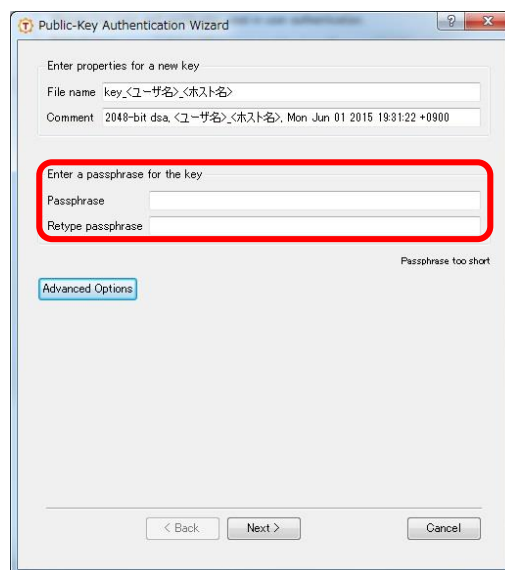


- ⑥ [Enter a passphrase for the key]の[Passphrase]と[Retype Passphrase]を入力する。

Passphrase : [任意のパスフレーズ]

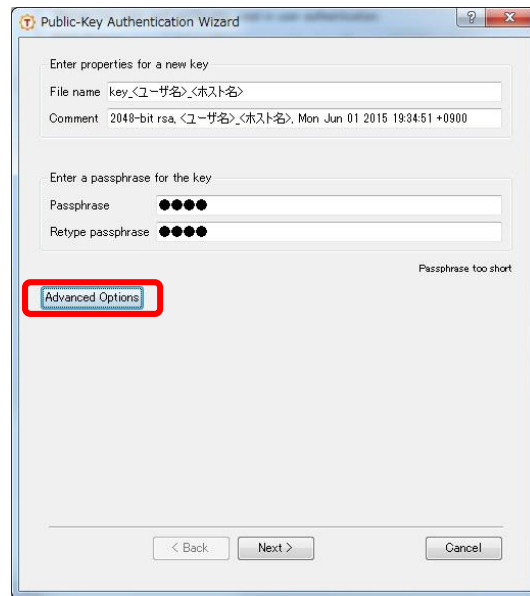
Retype Passphrase : [任意のパスフレーズ]

- ※ 鍵のパスフレーズを[無し]にした場合、秘密鍵があるクライアントから、パスフレーズ不要で、サーバへログインできるようになる。このため、秘密鍵の管理は、十分に注意して運用する必要がある。



<付録. 1> SSHの公開鍵認証について

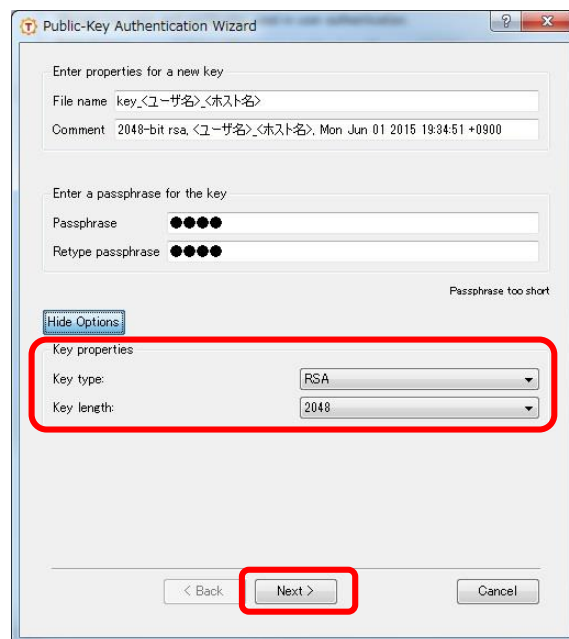
- ⑦ [Advanced Options] ボタンをクリックする。



- ⑧ [Key properties]にて、[Key type]と[Key length]を設定し、[Next]ボタンをクリックする。

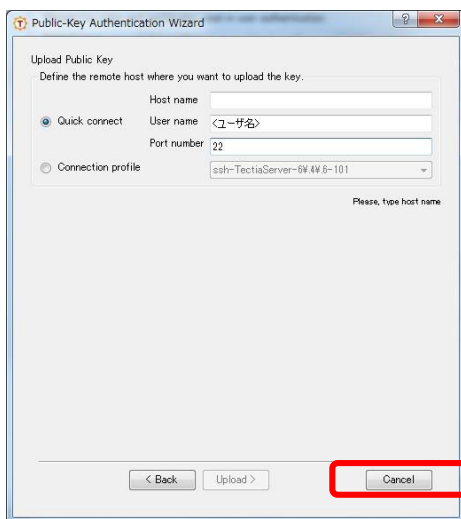
[Key type] : RSA

[Key length] : 2048

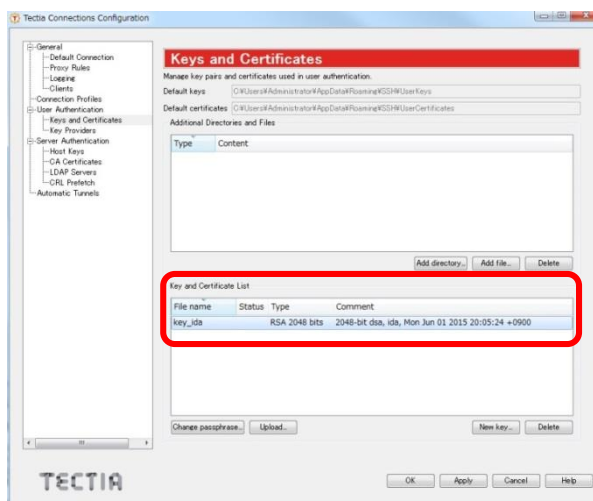


<付録. 1> SSHの公開鍵認証について

- ⑨ [公開鍵 (Public-Key)] をアップロードする [Public-Key Authentication Wizard] 画面が表示される。ここでは、[Cancel] ボタンをクリックする。



- ⑩ [Key and Certificate List] に鍵が追加されたことを確認する。

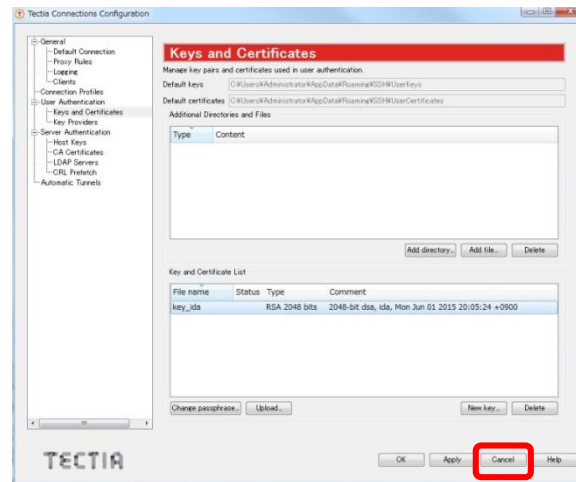


[鍵ファイルの生成場所]

項目	内容
ファイルパス	[Windows] ① %APPDATA%\%SSH%\UserKeys [Unix/Linux] ② \$HOME/.ssh2/authorized_keys
公開鍵	<作成ファイル名>.pub
秘密鍵	<作成ファイル名>

<付録. 1> SSHの公開鍵認証について

⑪ [Cancel]をクリックし、ウィンドウを閉じる



<付録.1> SSHの公開鍵認証について

(2) SSH サーバの設定

SSH サーバはデフォルトの設定で、パスワード認証・公開鍵認証による接続が可能である。

ここでは、SSH のサーバ側の設定において、セキュリティを向上させるために変更することが推奨される設定箇所と設定変更方法について記載する。

① Open SSH Server のセキュリティ強化設定

[コンフィグファイルの配置場所]

No	ソフトウェア	ファイルパス
1	Open SSH	/etc/ssh/sshd_config

[SSH サーバのセキュリティ強化設定項目]

上記のコンフィグファイルに記載されている下記の項目の設定を行う。

No	設定項目	設定方法
1	待受けポート番号の変更	SSH で接続する場合に使用するポート番号はデフォルトで 22 番になっている。外部からの不正アクセスなどの攻撃は、デフォルトの 22 番ポートへ行なわれる可能性が高いため、変更しておくことが推奨される。 [設定変更例] #Port 22 Port 2022 ※ 設定するポート番号は、現在他のサービスなどで使用されていないポート (1025 ～ 49151 番) を使用。
2	root でのログイン禁止	root ユーザで直接ログインできないように設定する。root ユーザはデフォルトで用意されているため、パスワード認証が利用できる場合はパスワードだけを推測して攻撃を受けやすくなる。その為、root ユーザで直接ログインはできないようにしておき、一般ユーザでログイン後に管理者になることが推奨される。 [設定変更例] #PermitRootLogin yes PermitRootLogin no

<付録.1> SSHの公開鍵認証について

3	パスワード認証の不許可	<p>公開鍵認証による SSH 接続が行なえるように設定した場合はパスワード認証ではログインできないようにすることが推奨される。</p> <p>[設定変更例]</p> <pre>#PasswordAuthentication yes PasswordAuthentication no</pre> <p>※ 公開鍵認証の設定を行う前に上記設定を実施した場合、コンソール接続からしかアクセスできなくなるため注意が必要。</p>
4	ログ設定のチューニング	<p>OpenSSH のデフォルト設定では、通信日時/接続元 IP アドレス/ユーザといった情報しかログ出力されない。ログの出力ログレベルをチューニングし、ファイル操作に関するログを取得する。</p> <p>[設定変更例]</p> <p>■ 操作ログの出力レベル</p> <p><input type="checkbox"/> 修正前</p> <pre>#LogLevel INFO</pre> <p><input type="checkbox"/> 修正後</p> <pre>LogLevel VERBOSE</pre> <p>■ SFTP ログの出力レベル</p> <p><input type="checkbox"/> 修正前</p> <pre>Subsystem sftp /usr/libexec/openssh/sftp-server</pre> <p><input type="checkbox"/> 修正後</p> <pre>Subsystem sftp /usr/libexec/openssh/sftp-server -f autoriv -l VERBOSE</pre> <p>※ ただしログが大量に出力されるようになるため、ログローテーションの設計が</p>

<付録.1> SSHの公開鍵認証について

		必要になる。
5	公開鍵の配置場所の変更	<p>一般的に公開鍵は各ユーザのホームディレクトリ配下の .ssh フォルダに配置されることが多く、該当のフォルダの鍵ファイルに意図しない公開鍵を追加されてしまった場合、意図しないアクセスを許可してしまうことになるため、公開鍵の配置場所を変更しておくことが推奨される。</p> <p>[設定変更例]</p> <p><input type="checkbox"/>変更前</p> <pre>#AuthorizedKeysFile .ssh/authorized_keys</pre> <p><input type="checkbox"/>変更後</p> <pre>AuthorizedKeysFile <任意の格納場所>/authorized_keys</pre>

② Tectia SSH Server のセキュリティ強化設定

[コンフィグファイルの配置場所]

No	ソフトウェア	ファイルパス
1	Tectia SSH Server	/etc/ssh2/ ssh-server-config.xml

[SSH サーバのセキュリティ強化設定項目]

上記のコンフィグファイルに記載されている下記の項目の設定を行う。

No	設定項目	設定方法
1	待受けポート番号の変更	<p>SSH で接続する場合に使用するポート番号はデフォルトで 22 番になっている。外部からの不正アクセスなどの攻撃は、デフォルトの 22 番ポートへ行なわれる可能性が高いため、変更しておくことが推奨される。</p> <p>[設定変更例]</p> <p><input type="checkbox"/>変更前</p> <pre><listener id="default" port="22" /></pre> <p><input type="checkbox"/>変更後</p> <pre><listener id="default" port="2022" /></pre> <p>※ 設定するポート番号は、現在他のサービ</p>

<付録.1> SSHの公開鍵認証について

		スなどで使用されていないポート(1025～49151番)を使用。
2	rootでのログイン禁止	<p>root ユーザで直接ログインできないように設定する。root ユーザはデフォルトで用意されているため、パスワード認証が利用できる場合はパスワードだけを推測して攻撃を受けやすくなる。その為、root ユーザで直接ログインはできないようにしておき、一般ユーザでログイン後に管理者になることが推奨される。</p> <p>[設定変更例]</p> <p>下記の追加設定を<authentication-methods xxx>の最下部に追記する。</p> <p><input type="checkbox"/>追加設定</p> <pre><authentication action="deny"> <selector> <user-privileged value="yes" /> </selector> </authentication></pre>
3	パスワード認証の不許可	<p>公開鍵認証によるSSH接続が行なえるように設定された場合はパスワード認証ではログインできないようにすることが推奨される。</p> <p>[設定変更例]</p> <p><input type="checkbox"/>削除項目</p> <p><authentication-methods xxx>にある下記の項目を削除。</p> <pre><auth-password /></pre> <p>※ 公開鍵認証の設定を行う前に上記設定を実施した場合、コンソール接続からしかアクセスできなくなるため注意が必要。</p>
4	公開鍵の配置場所の変更	<p>一般的に公開鍵は各ユーザのホームディレクトリ配下の.sshフォルダに配置されることが多く、該当のフォルダの鍵ファイルに意図しない公開鍵を追加されてしまった場合、意図しないアクセスを許可してしまうことになるため、公開鍵の配置場所を変更しておくことが推奨される。</p>

<付録. 1> SSHの公開鍵認証について

		<p>[設定変更例]</p> <p><input type="checkbox"/>変更前</p> <p><authentication-methods>内の下記を変更する。 <auth-publickey authorized-keys- directory="%D/.ssh2/authorized_keys" /> ... </authentication></p> <p><input type="checkbox"/>変更後</p> <p><auth-publickey authorized-keys-directory="<u><任意のディレクトリ></u>/authorized_keys" /> ... </authentication></p> <p>※[%D] もしくは[%homedir%]はユーザーのホームディレクトリです。</p>
--	--	---

<付録.1> SSHの公開鍵認証について

(3) ユーザ公開鍵の配置

ここではサーバ管理者向けに、すでに下記のようなサーバのセキュア化が実施されている環境を想定したユーザ公開鍵の配置方法を説明する。

[サーバのセキュア化及び設定状況]

- ① OSに公開鍵認証を行うユーザがすでに追加されている環境を想定
- ② 公開鍵のサーバへの配置は、サーバの管理者が実施する環境を想定
- ③ サーバではパスワード認証の不許可の設定がされている環境を想定
- ④ サーバへの root でのログイン禁止の設定がされている環境を想定

■ OpenSSH サーバの公開鍵の配置手順

[デフォルトの公開鍵認証の配置場所]

ファイル	場所	パーミッション
SSH ディレクトリ	<ユーザディレクトリ>/.ssh/	700
公開鍵ファイル	<ユーザディレクトリ>/.ssh/ authorized_keys	600

- ① メンテナンスユーザの公開鍵認証で対象のサーバへログインする。
- ② 公開鍵認証のユーザに切り替える。
\$ su <公開鍵認証を行うユーザ>

※ 公開鍵ディレクトリ/公開鍵ファイルを作成するため、
所有権を該当ユーザに変更するために必要な操作になる。
所有権を後程変更できる場合は不要。
- ③ ユーザのホームディレクトリに移動する。
\$ cd \$HOME
- ④ 公開鍵認証の設定ディレクトリ “.ssh” が作成されているか確認する。
\$ ls -la

※ 初回時(ユーザ作成時)デフォルトでは “.ssh” が作成されないので、
 “.ssh” が作成されていないことを確認する。
- ⑤ 作成されていない場合は “.ssh”ディレクトリを作成する。
\$ mkdir .ssh
- ⑥ “.ssh”ディレクトリのパーミッション(アクセス権)を設定する
\$ chmod 700 .ssh

※ 所有者にだけ読み取り権、書き込み権、実行権(パーミッション“700”)を
設定する。

<付録.1> SSHの公開鍵認証について

- ⑦ “.ssh”ディレクトリに移動する。

```
$ cd .ssh
```

- ⑧ 公開鍵登録ファイルを作成する。

```
$ touch authorized_keys
```

- ⑨ 公開鍵登録ファイルのパーミッション(アクセス権)を設定する。

```
$ chmod 600 authorized_keys
```

※ 所有者にだけ読み取り権、書き込み権(パーミッション“600”)を設定する。

- ⑩ 公開鍵登録ファイルに公開鍵を登録する。

- (a) サーバ上で、公開鍵登録ファイルをテキストエディタで開く。

```
$ vi authorized_keys
```

- (b) クライアント PC 上で公開鍵をテキストエディタで開く。

- (c) この文字列を全て選択してコピーする。

- (d) コピーした文字列をサーバ上の authorized_keys(公開鍵登録ファイル)へ、ペーストする。

※ あらかじめ公開鍵ファイルをアップロードできる場合は、下記のコマンドで、公開鍵を公開鍵登録ファイルに登録することが可能。

```
$ cat <公開鍵ファイル> >> $home.ssh/authorized_keys
```

<付録.1> SSHの公開鍵認証について

■Tectia SSH Server の公開鍵の配置手順

[デフォルトの公開鍵認証の配置場所]

ファイル	場所	パーミッション
SSH ディレクトリ	<ユーザディレクトリ>/ .ssh2/	700
公開鍵格納ディレクトリ	<ユーザディレクトリ>/ .ssh2/ authorized_keys	700
公開鍵ファイル	<任意の公開鍵ファイル名>	600

- ① メンテナンスユーザの公開鍵認証で対象のサーバへログインする。
- ② 公開鍵認証のユーザに切り替える。
\$ su <公開鍵認証を行うユーザ>
(a) 公開鍵ディレクトリ/公開鍵ファイルを作成するため、
所有権を該当ユーザに変更するために必要な操作になる。
所有権を後程変更できる場合は不要。
- ③ ユーザのホームディレクトリに移動する。
\$ cd \$HOME
- ④ 公開鍵認証の設定ディレクトリ “.ssh2” が作成されているか確認する。
\$ ls -la
※ 初回時(ユーザ作成時)デフォルトでは “.ssh2” が作成されないの
で、 “.ssh2” が作成されていないことを確認する。
- ⑤ 作成されていない場合は “.ssh2”ディレクトリを作成する。
\$ mkdir .ssh2
- ⑥ “.ssh2”ディレクトリのパーミッション(アクセス権)を設定する。
\$ chmod 700 .ssh2
※ 所有者にだけ読み取り権、書き込み権、実行権(パーミッション“700”)を
設定する。
- ⑦ “.ssh2”ディレクトリに移動する。
\$ cd .ssh2
- ⑧ 公開鍵の格納ディレクトリを作成する。
\$ mkdir authorized_keys

<付録. 1> SSHの公開鍵認証について

- ⑨ 公開鍵の格納ディレクトリのパーミッション(アクセス権)を設定する。

```
$ chmod 700 authorized_keys
```

※ 所有者にだけ読み取り権、書き込み権、実行権(パーミッション"700")を設定する。

- ⑩ 公開鍵を公開鍵格納ディレクトリに転送する。

```
$ scp <公開鍵ファイル> <ユーザ名>@<サーバ>:<鍵格納ディレクトリ>
```

(4) 公開鍵認証方式での接続方法

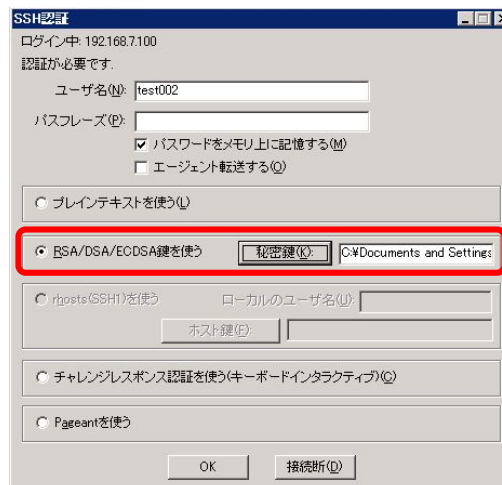
ここではクライアントにおける公開鍵認証での接続方法について説明する。

■TeraTerm での接続方法

- ① TeraTerm を起動する。
- ② [新しい接続]ダイアログが表示されるので、接続先のサーバのホスト名、もしくは IP アドレスを入力し、「OK」ボタンを押す。

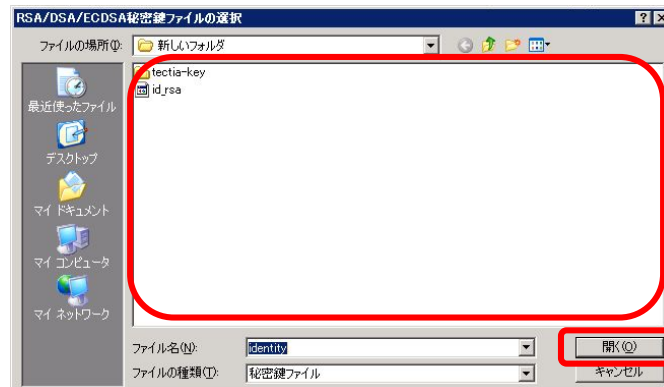


- ③ SSH 認証ダイアログが表示されるので、ユーザ名を入力し[RSA/DSA/ECDSA 鍵を使う]にチェックをつけ、[秘密鍵]ボタンをクリックする。

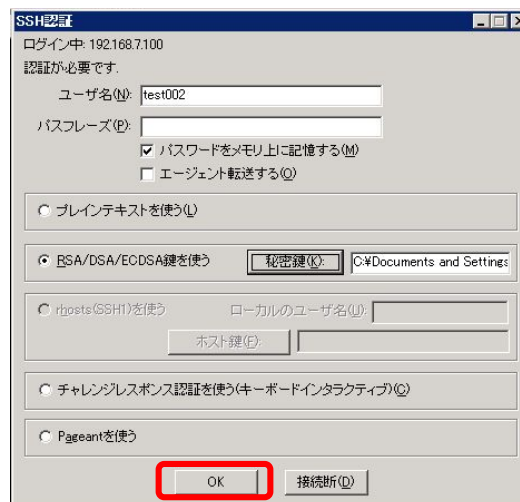


<付録. 1> SSHの公開鍵認証について

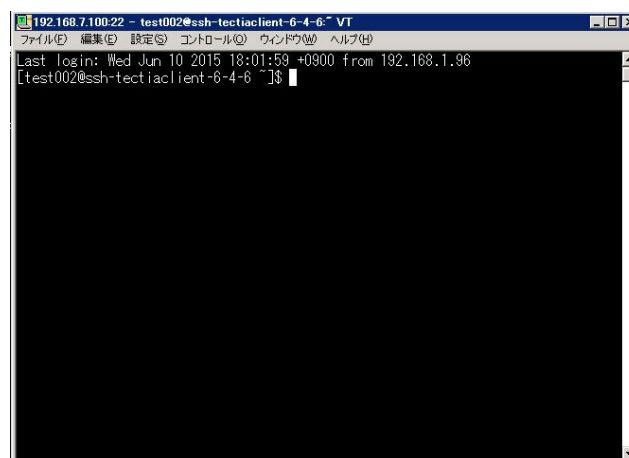
- ④ ファイルの選択画面が表示されるので、③で入力したユーザの秘密鍵を選択し、[開く]ボタンをクリックする。



- ⑤ 下記画面に戻るので、[OK]ボタンをクリックする。



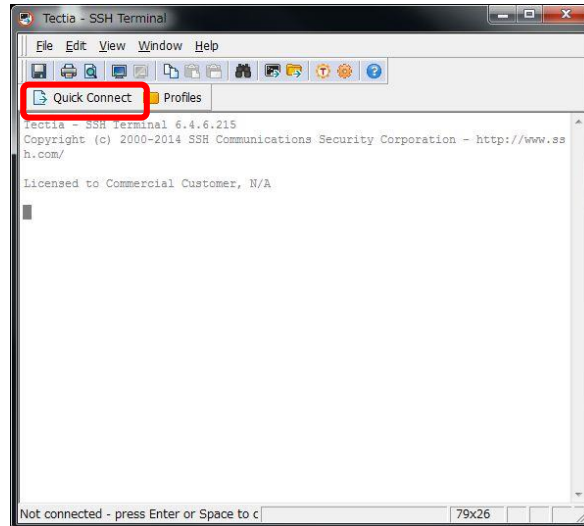
- ⑥ 下記、リモート操作画面が表示される。



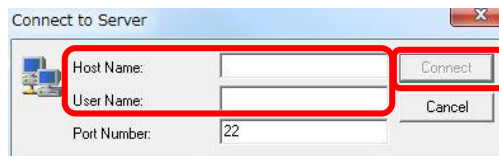
<付録.1> SSHの公開鍵認証について

■Tectia SSH Client での接続方法

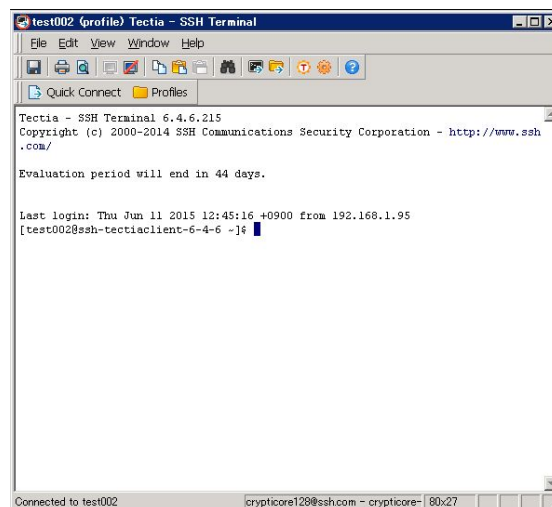
- ① Tectia SSH Client を起動する。
- ② SSH Terminal 画面が表示されるので、[Quick Connect]をクリックする。



- ③ Connect to Server 画面が表示されるので、[Hostname]に[接続先のホスト名もしくは IP アドレス]、[User Name]に[ユーザ名]を入力し、[Connect]を[クリック]する。



- ④ 下記、リモート操作画面が表示される。



















<付録.1> SSHの公開鍵認証について

<補足>

上記では、鍵の作成/配布、サーバでのセキュア化設定について述べてきたが、これらの作業はサーバ台数が多くなると、サーバ毎、ユーザ毎に手動での作業が必要となり非常に手間がかかることが想定される。

Tectia SSH 製品の Universal SSH Key Manager を使うことで、管理 GUI 上から一元的にかつ簡単に各サーバ上での鍵生成/配布といった操作やサーバ設定が行えるため、結果、管理者の作業負担を軽減でき、運用作業の効率化につながる。是非、Tectia 製品の導入をご検討ください。

～ Universal SSH Key Manager の 鍵管理画面 ～

From	To
 root@172.16.32.110	 root@172.16.32.111
 maintenance@172.16.32.111	 root@172.16.32.110
 guest1@172.16.32.110	 dit@172.16.32.111
 Unknown private key@172.16.32.1	 root@172.16.32.110
 sato@172.16.32.110	 root@172.16.32.111
	<u>6 Authorized keys on 1 host:</u>
	 eigyo@172.16.32.110
	 guest1@172.16.32.110
	 kanri@172.16.32.110
 root@172.16.32.111	 root@172.16.32.110
	 support@172.16.32.110

※ 送信元(From) 、送信先(To)、 ユーザ@IP アドレスの形式で分かり易く表示。

以上

ssh® and Tectia® are registered trademarks of SSH Communications Security Corporation in the United States and in certain other jurisdictions.

SSH and Tectia logos and names of SSH products and services are trademarks of SSH Communications Security Corporation and are protected by international copyright laws and treaties.

Logos and names of the products may be registered in certain jurisdictions.

Copyright © 2014–2015 SSH Communications Security Corporation. All rights reserved.