「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？”

2015 年 09 月

目次

1. はじめに

SSH とは「Secure Shell（セキュア・シェル）」の略で、システムのメンテナンス時やサーバ間通信のセキュリティを確保するために、世界中で広く利用されている非常に強力なツールである。しかし、SSH を正しく理解して設計や運用をしていなければ、セキュリティ上の大きなリスクが存在することとなる。

本冊子では、危険な SSH の使われ方とその対策について解説する。

2. SSH とは

インターネットが普及し始めたころ、システムのメンテナンス通信、サーバ間通信は telnet や FTP が主流であった。しかし telnet や FTP は暗号化されていないため、現在では､セキュアなプロトコルとして SSH が利用されている｡SSH は世界中の企業 IT 環境の 90%で使用されている。

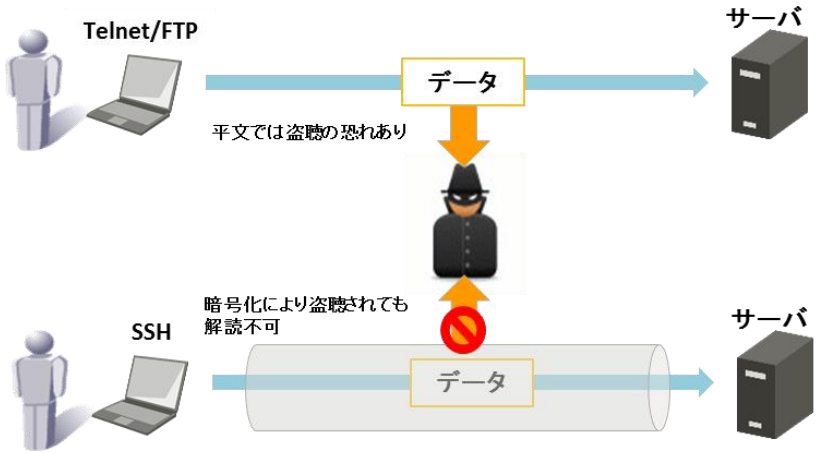


図 2.1. SSH による通信の暗号化

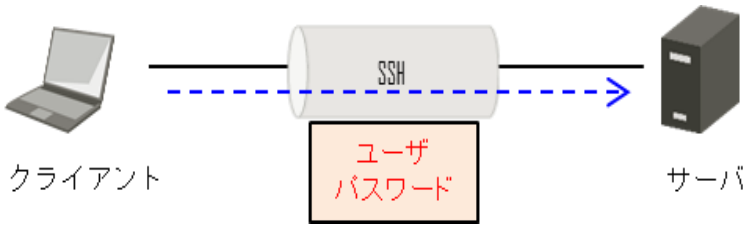
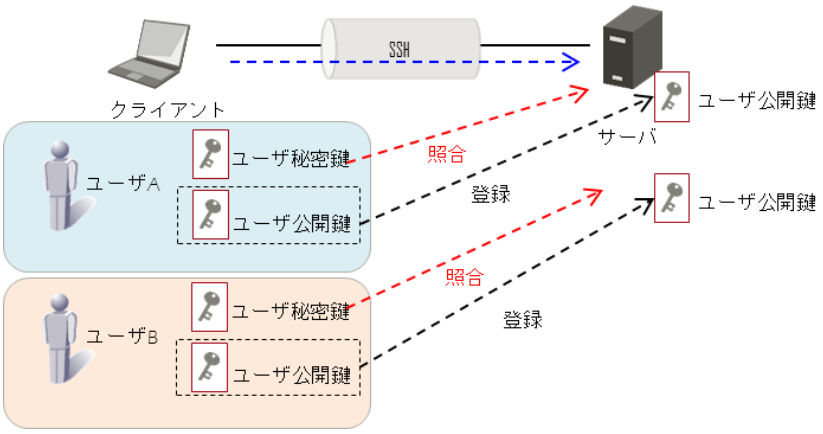
以下に SSH プロトコルの優位点および認証方式について記載する。

＜SSH プロトコルの優位点＞

項目	内容
認証	・パスワード認証以外に、公開鍵認証等の複数の認証方式に対応。 ・ユーザ名及び接続元 IP アドレスによりアクセス制御が可能。 ・一致する鍵ペアを持つユーザのみに接続を制限することが可能。 ・二段階認証(公開鍵+パスフレーズ)を設定することが可能。
通信データの保護	・ID/パスワード等の認証情報も含めて全ての通信データを暗号化可能。 ・エンドツーエンド(アプリケーションレベル)で暗号化されているためデータの盗聴が困難。 ・データのハッシュ値を計算しているため、データの改竄を検知可能。
導入・設定変更が容易	・Linux 系では SSH はプリインストールされているため、ネットワーク経路上で SSH プロトコル（ポート番号 22 番）を許可するだけで容易に導入可能。 ・SSH プロトコルが許可されていれば、既存のネットワーク環境を変更しなくても、任意のアプリケーション通信の暗号化が可能。

表 1-1. SSH プロトコルの優位点

＜SSH プロトコルの認証方式＞

認証方式	概要
パスワード認証	パスワード認証はログインの際にユーザ名とパスワードを使って認証を行う方式である。事前にサーバ側でユーザとパスワードを設定しておく必要があり、一般的にユーザ名とパスワードの入力を対話形式で利用されるケースが多い。またサーバ間の自動化通信(ファイル転送処理等)のスクリプト内にパスワードが埋め込まれた形で利用されているケースもある。 
公開鍵認証	公開鍵認証はユーザ鍵を使って認証を行う方式で、クライアント側の秘密鍵とサーバ側の公開鍵を照合し対となる鍵を持つユーザのみサーバに接続が可能となる。事前にユーザの鍵ペア(秘密鍵/公開鍵)を作成し、承認済の鍵としてサーバ上に公開鍵を配置する必要がある。  秘密鍵はパスフレーズによって保護することも可能で、簡単に2段階認証を行うことができる。また公開鍵認証ではコマンド制限の設定が可能である。

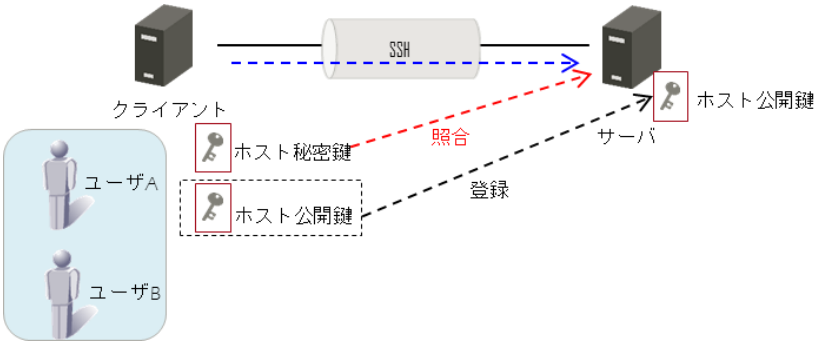
ホストベース認証	ホストベース認証はホスト鍵を使って認証を行う方式で、クライアント側の秘密鍵とサーバ側の公開鍵を照合し対となる鍵を持つホストのみサーバに接続が可能となる。公開鍵認証との違いはユーザレベルではなくホストレベルで認証を行う点である。  ホストレベルでは、複数のユーザが存在してもサーバ単位で接続が許可される。事前に 1 組のホストの鍵ペア(秘密鍵/公開鍵)を作成し、承認済の鍵としてサーバ上に公開鍵を配置する必要がある。

表 2.1. SSH プロトコルの認証方式

3. SSH に潜む落とし穴

SSH を利用していれば経路は暗号化されるので安心しがちであるが、パスワード認証やユーザ ID の使い回し等は非常に危険である。以下に危険な SSH の使い方や SSH の運用における課題について述べる。

① 安全性の低いパスワードの使用

SSH ブルートフォースアタック（ID/パスワード総当たり）といった攻撃や内部犯行による情報漏えい事件が多数発生しており、パスワードのずさんな管理によって発生しているものも多い。安全性の低いパスワードだと簡単に破られてしまう危険性がある。

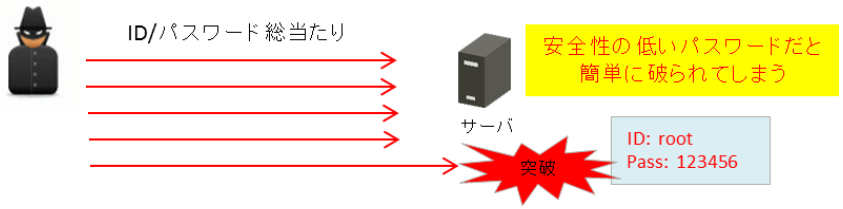


図 3.1. 安全性の低いパスワードの使用

② ID/パスワードの使い回し

システム内の各サーバで共通 ID/パスワードが使われていると、その ID/パスワードが１つ漏洩しただけで、全てのサーバが被害にあってしまう。どのサーバにも簡単にアクセスできてしまうので、情報漏えい等の危険性が高まる。

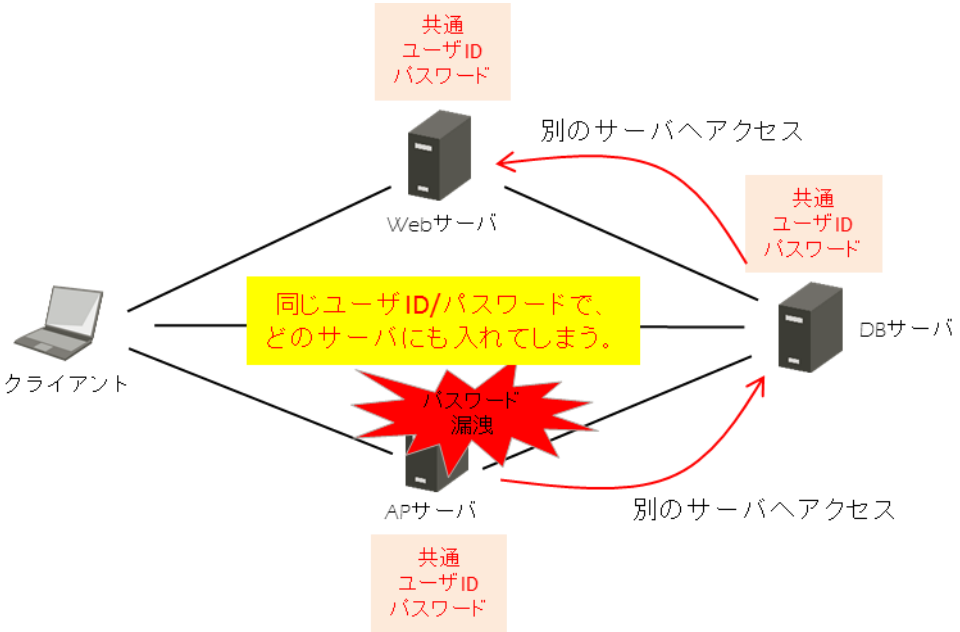


図 3.2. ID/パスワードの使い回し

③ 自動スクリプト内のパスワード埋め込み

自動転送スクリプト等に直接パスワードが書き込まれていると、スクリプトファイルは平文でサーバ上に格納されているため、容易にパスワードの入手が可能となってしまう。

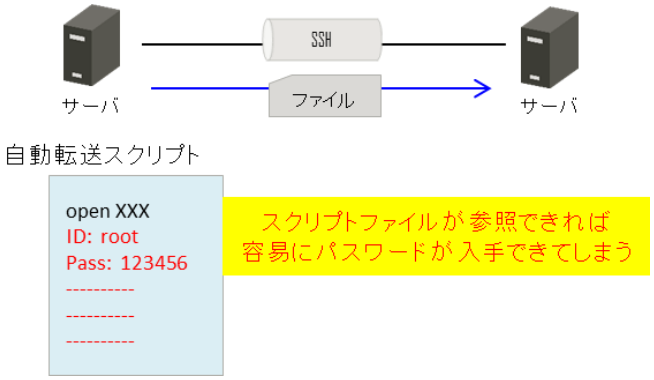


図 3.3. 自動スクリプト内のパスワード埋め込み

④ 公開鍵認証の鍵の使い回し

公開鍵認証では、一つの鍵を複数のユーザで使い回すといった危険な使い方は絶対に避けるべきである。鍵ファイルが１つ漏洩しただけで、全てのサーバが被害にあってしまう。どのサーバにも簡単にアクセスできてしまうので、情報漏えい等の危険性が高まる。

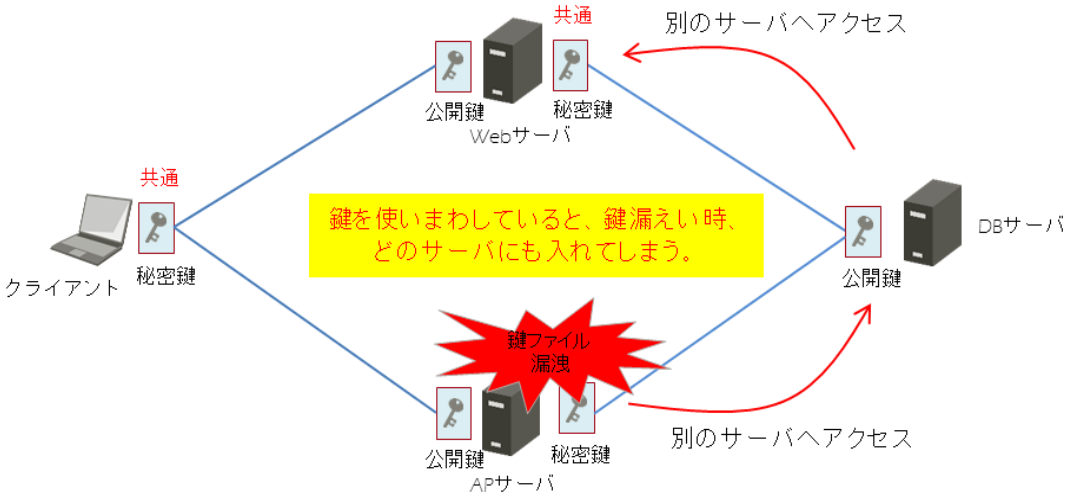


図 3.4. 鍵の使い回し

⑤ 通信履歴/内容の把握がしづらい

OpenSSH ではデフォルトで通信日時/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ため、管理者はどういった操作が行われたかの内容まで含めて把握することができない。管理者にとってサーバ個々のログを確認するには手間がかかりシステム全体における通信内容を簡単に一括で把握することは難しい。また暗号化されているため、不正通信/情報漏えいという観点で通信の中身をチェックすることも困難である。

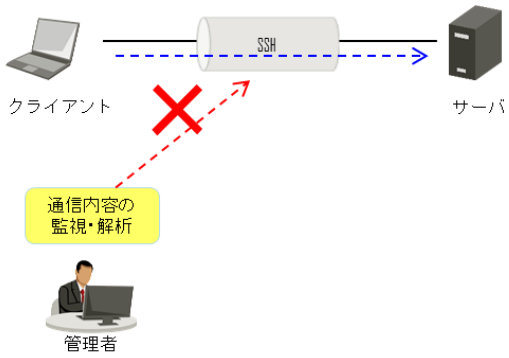


図 3.5. 通信履歴/内容の把握がしづらい

⑥ 公開鍵認証の鍵管理

公開鍵認証による SSH を運用していく上で、管理者はどのサーバに鍵ファイルが配置されているか把握し、不明な鍵や不要な鍵が存在していないかチェック、適切に管理/運用していくことが求められる。

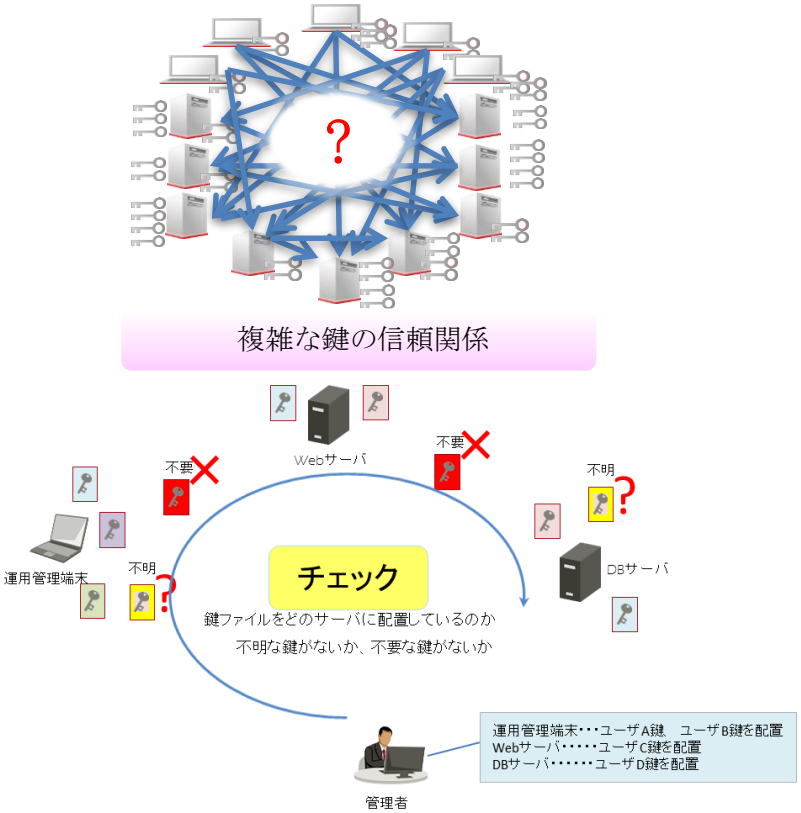


図 3.6. 公開鍵認証の鍵管理

しかし、多数のサーバを運用している環境で、不正な鍵が存在していないかのチェックはサーバ 1 台 1 台にログインして確認する必要があり、非常に手間がかかる。また鍵の更新や削除、鍵ファイル転送といった操作もサーバ 1 台 1 台にログインして操作する運用の場合、ある程度のコストがかかる。以下に試算表を示す。

項目	内容
環境内の SSH 台数	50 台
1 台当たりの鍵設定数	10 鍵
鍵設定にかかる平均時間/1 鍵	2 時間
セキュリティ管理者の平均時間給	6000 円
年間運用コスト	2400 万円

表 3.1. 運用コスト例

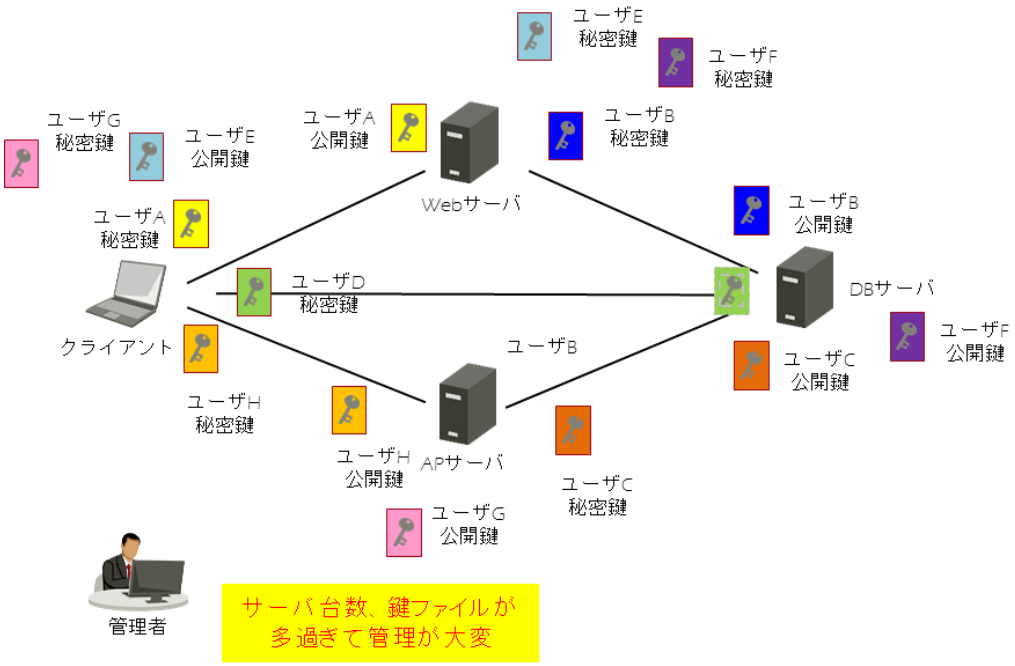


図 3.7. 公開鍵認証の鍵管理

4. SSH での効果的な対策

本章では、“3. SSH に潜む落とし穴”で列挙した課題の対策を述べる。

No	SSH に潜む落とし穴	対策
1	安全性の低い パスワードの使用	■安全性の高いパスワードを使用する ① パスワードは ID と同じにしない ② 8 文字以上の大小英数字および記号を含む文字列にする ③ パスワードの推測が容易なものは避ける ④ 複数のサーバで同じパスワードを使用しない ⑤ パスワードを定期的に変更する ⑥ ログインの試行回数を制限 ※ 1 つ 1 つに安全性の高いパスワードを設定し、 そのすべてを記憶することは非常に困難
2	ID/パスワードの 使い回し	■ユニークなパスワードを利用する ・サーバ別／ユーザ別に異なるパスワードを利用する ※ 1 つ 1 つにユニークなパスワードを設定し、 そのすべてを記憶することは非常に困難である。
3	自動スクリプト内の パスワード埋め込み	■パスワードファイルの中身の文字列を暗号化する ・本来のパスワードを MD5 や DES などで暗号化

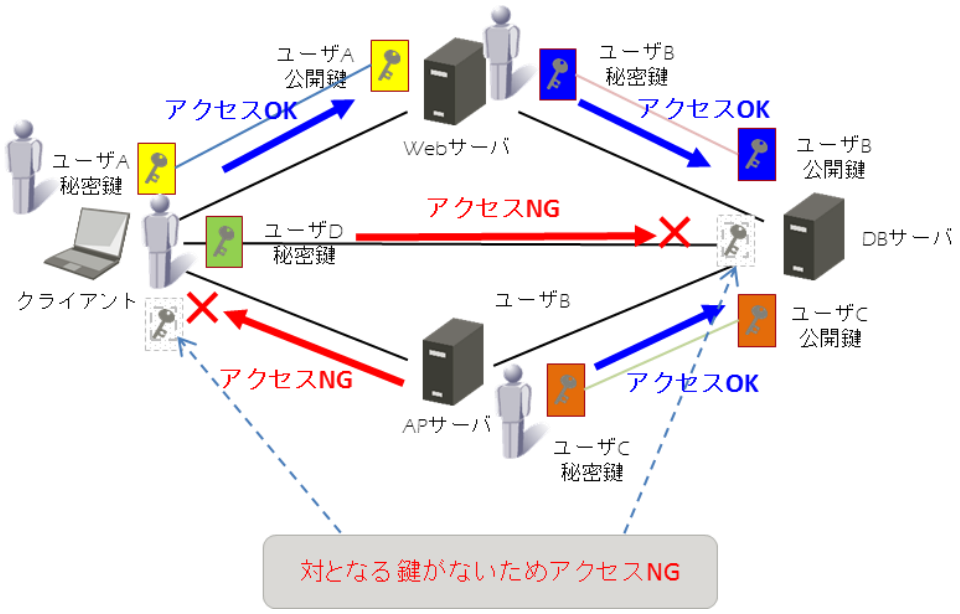
表 4.1. SSH での効果的な対策 その 1

No. 1～3 の推奨対策

■公開鍵認証によるアクセス制御を行う

公開鍵認証を利用することで、ID/パスワード入力をする必要がなくなるため、
パスワード認証で挙げられる課題を解決することが可能である。

〈公開鍵認証によるアクセス制御のイメージ図〉



以下に公開鍵認証のメリットを記載する。

認証方式比較	パスワード認証	公開鍵認証
対ブルートフォース攻撃	弱	強
自動化スクリプトの認証情報の更新	困難	容易
コマンドの実行制限	不可	可

※ 公開鍵認証の仕組みの詳細については、「〈付録．1〉 SSH の公開鍵認証について」を参照。

表 4.2. SSH での効果的な対策 その２

No	SSH に潜む落とし穴	対策
4	公開鍵認証の鍵の使い回し	■ユニークな鍵を使用する クライアント - サーバ、またはサーバ - サーバごとに個々に鍵ペアを作成する。鍵のペアを保持する者同士のみアクセス可能となるため、より強固なアクセス環境をつくることができる。さらには、同じユーザであっても接続先のサーバ毎に鍵ペアを作成することでセキュリティを高めることができる。
5	通信履歴/内容の把握がしづらい	■ログ設定をチューニングする。 ログ設定をチューニングすることで、通信日時/接続元 IP アドレス/ユーザといった情報以外にもファイル操作に関するログを確認することが可能である。 より詳細な証跡取得、一括で監査したい場合には、証跡管理製品の導入を検討する必要がある。
6	公開鍵認証の鍵管理	■鍵の管理台帳を作成し、管理する ユーザ鍵の送信元(秘密鍵)と送信先(公開鍵)の関連性を紐づけ、管理台帳によりユーザ鍵の管理を行う。 ［管理項目例］ ① ユーザ名 ② ホスト名 ③ IP アドレス ④ 鍵ファイル名 ⑤ 暗号化アルゴリズム ⑥ 暗号化強度 ⑦ 有効期限 ⑧ 鍵ファイル配置場所 ⑨ コマンド制限設定の有無 ■不正な鍵がないか、確認する。 不正な鍵が存在していないか定期的にサーバ 1 台 1 台にログインして確認する。 運用コストを削減したい場合には、自動で鍵情報を収集し信頼関係を可視化できる製品の導入を検討する必要がある。

表 4.3. SSH での効果的な対策 その３

5. 最後に

SSH の利用にあたっては、上記で述べてきたとおり、パスワード認証は脆弱であるため公開鍵認証を使った SSH 環境を構築することを推奨する。また ID/パスワード・公開鍵の使い回し等の危険な使い方を避けることで、よりセキュアに運用することができる。ただし、管理者は詳細な証跡取得の必要性についての検討や鍵管理の運用コストを下げるための対策について検討しなければならない。

Tectia SSH 製品シリーズは SSH を発明した SSH Communications Security 社（フィンランド）が開発したもので、SSH の問題点を解決するために総合的なソリューションを提供する製品である。Tectia SSH 製品を導入することで、自動化されたサーバ間通信を含めた SSH 通信における監査証跡の取得/情報漏洩防止、さらには SSH の鍵情報の可視化、鍵の一元管理が可能となり大幅な運用コスト削減を実現できる。

[Tectia SSH 製品シリーズ]

製品	概要
通信環境の管理・監査ソリューション	
Universal SSH Key Manager	SSH クライアント/サーバの鍵情報を収集/可視化、鍵の配布/失効等のライフサイクルを自動管理
CryptoAuditor	リモート管理に利用される通信(SSH/RDP)、自動化されたサーバ間通信の監視、監査証跡を取得
セキュアなファイル転送ソリューション	
Tectia ConnectSecure	既存インフラ、スクリプト、アプリケーションに手を加えず平文ファイル転送を容易にセキュア化
Tectia SSH Client/Server	エンドツーエンドでのセキュアなリモートアクセス、ファイル転送機能を提供。ファイル転送機能は、OpenSSH と比較して 3～4 倍の高速なファイル転送を実現

表 5-1. Tectia SSH 製品シリーズ

以上

SSH 実装勉強会

■執筆および協力者一覧

木口　雅博　株式会社 富士通ソーシアルサイエンスラボラトリ
駒形　真樹　株式会社 富士通ソーシアルサイエンスラボラトリ
坂口　徹　株式会社 富士通ソーシアルサイエンスラボラトリ
光吉　浩之　株式会社 富士通ソーシアルサイエンスラボラトリ
山崎　俊一　株式会社 NTT データ・フィナンシャルコア
梶　一俊　株式会社 ディ アイ ティ
荒井　勇樹　株式会社 ディ アイ ティ
佐藤　隼　株式会社 ディ アイ ティ

■お問い合わせ先

株式会社 ディアイティ　ネットワークセキュリティ事業部
http://www.dit.co.jp/　E-mail：info@dit.co.jp