LinuC レベル2
技術解説セミナー

2.12.2 OpenSSH サーバーの設定と管理

株式会社ゼウス・エンタープライズ
（LPI-Japanプラチナスポンサー）

鯨井貴博(LinuCエヴァンジェリスト)

WHO ARE YOU?(講師紹介)

鯨井貴博
LPI-Japan プラチナスポンサー　株式会社ゼウス・エンタープライズ
LinuCエヴァンジェリスト

Linuxとの出会い

大学時代 Unixの存在を知り、日経Linuxを読み始める。
2000年にVine Linux 2.0で一度挫折を経験。
その悔しさを忘れきれず、2007年 他業種からIT業界に転職しLinuxに再チャレンジ。

SE・商用製品サポート・インストラクター・プロジェクト管理などを経験し、現在に至る。
自分自身が学習で苦労した経験から、初心者を含む受講者に分りやすい講義を行うように心がけている。

また、興味の向くIT技術・オープンソースソフトウェアなどについて、
Opensourcetehブログ（https://www.opensourcetech.tokyo/）で執筆中。
実際に自分でやってみる/使ってみる・開発者本人から話を聞いてみることを大切にしています。

Linus Torvaldsさん(Linux開発者)


Igor Sysoevさん(nginx開発者)


Alexei Vladishevさん(Zabbix開発者)

WHO ARE YOU?(講師紹介)

Open Source Summit Japan 2023

Open Source Summit Japan 2023: ボランティアリーダーの体験記
https://www.opensourcetech.tokyo/entry/20231224/1703429785

アジェンダ

1. LinuCについて
   試験概要と特徴

2. 技術解説
   2.12.2 OpenSSH サーバーの設定と管理
   重要度　　4
   概要　　SSHデーモンの設定と保護ができる。これには、鍵の管理とユーザ用にSSHを設定することも含まれる。
   詳細
   OpenSSH サーバーの設定ファイルとデーモン
   sshd, /etc/ssh/sshd_config
   /etc/ssh/ssh_host_*_key および ssh_host_*_key.pub
   スーパーユーザおよび一般ユーザのログインを制限する。
   PermitRootLogin, PubKeyAuthentication, AllowUsers, PasswordAuthentication

3. Appendix

4. お知らせ

5. Q&A

本日のゴール

- OpenSSHの設定を理解する
- 鍵の管理を理解する
- ログイン制限を理解する

Linux技術者認定「LinuC（リナック)」とは

■LinuCとは

クラウド／DX時代のITエンジニアに求められるシステム構築から運用管理に必要なスキルを証明する技術者認定です。

✓クラウド活用に役立つスキルの習得

- オンプレミス／仮想化・コンテナを問わず様々な環境下でのサーバー構築
- 他社とのコラボレーションの前提となるオープンソースへの理解

✓習得できるスキルが実践的

問題作成にはトップエンジニアも参加するコミュニティ内の意見を取り込むことで、本当に必要な内容を網羅的に盛り込んでいます。

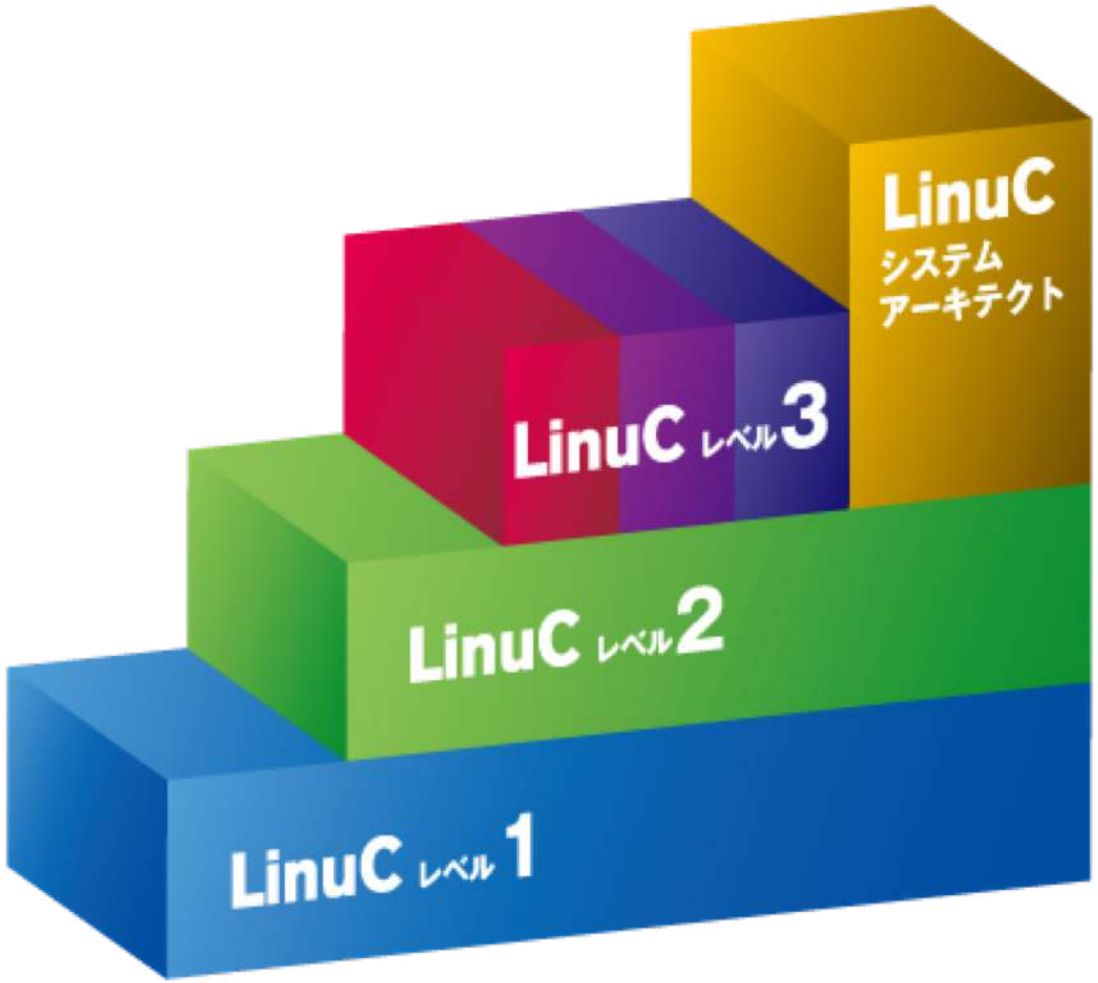
✓上流工程を担うアーキテクトの領域までカバー

システムの運用管理からアーキテクチャ設計までの４つのレベルをひとつずつ習得していくことで、活躍できるエンジニアとして必要なスキルを網羅的に身につけていくことができます

LinuC
Linux技術者認定

LinuCの４つのレベル

LinuCは、サーバーの運用管理からアーキテクト設計まで、システム開発・運用に必要な知識とスキルを体系立てて習得することができます。



LinuC システムアーキテクト
ITプロジェクトを成功に導く上級エンジニア
SA01試験 | SA02試験

LinuC レベル3
高度な技術力を備えた特定分野のスペシャリスト
304試験（仮想化&高可用性）
300試験（混在環境） | 303試験（セキュリティ）

LinuC レベル2
仮想マシン・コンテナを含むLinuxシステム、ネットワークの設計・構築
201試験 | 202試験

LinuC レベル1
物理／仮想Linuxサーバーの操作・運用
101試験 | 102試験

LinuC レベル1

101試験

1.01：Linuxのインストールと仮想マシン・コンテナの利用
- 1.01.1Linuxのインストール、起動、接続、切断と停止
- 1.01.2仮想マシン・コンテナの概念と利用
- 1.01.3ブートプロセスとsystemd
- 1.01.4プロセスの生成、監視、終了
- 1.01.5デスクトップ環境の利用

1.02：ファイル・ディレクトリの操作と管理
- 1.02.1ファイルの所有者とパーミッション
- 1.02.2基本的なファイル管理の実行
- 1.02.3ハードリンクとシンボリックリンク
- 1.02.4ファイルの配置と検索

1.03：GNUとUnixのコマンド
- 1.03.1コマンドラインの操作
- 1.03.2フィルタを使ったテキストストリームの処理
- 1.03.3ストリーム、パイプ、リダイレクトの使用
- 1.03.4正規表現を使用したテキストファイルの検索
- 1.03.5エディタを使った基本的なファイル編集の実行

1.04：リポジトリとパッケージ管理
- 1.04.1apt コマンドによるパッケージ管理
- 1.04.2Debianパッケージ管理
- 1.04.3yumコマンドによるパッケージ管理
- 1.04.4RPMパッケージ管理

1.05：ハードウェア、ディスク、パーティション、ファイルシステム
- 1.05.1ハードウェアの基礎知識と設定
- 1.05.2ハードディスクのレイアウトとパーティション
- 1.05.3ファイルシステムの作成と管理、マウント

102試験

1.06：シェルおよびスクリプト
- 1.06.1シェル環境のカスタマイズ
- 1.06.2シェルスクリプト

1.07：ネットワークの基礎
- 1.07.1インターネットプロトコルの基礎
- 1.07.2基本的なネットワーク構成
- 1.07.3基本的なネットワークの問題解決
- 1.07.4クライアント側のDNS設定

1.08：システム管理
- 1.08.1アカウント管理
- 1.08.2ジョブスケジューリング
- 1.08.3ローカライゼーションと国際化

1.09：重要なシステムサービス
- 1.09.1システム時刻の管理
- 1.09.2システムのログ
- 1.09.3メール配送エージェント(MTA)の基本

1.10：セキュリティ
- 1.10.1セキュリティ管理業務の実施
- 1.10.2ホストのセキュリティ設定
- 1.10.3暗号化によるデータの保護
- 1.10.4クラウドセキュリティの基礎

1.11：オープンソースの文化
- 1.11.1オープンソースの概念とライセンス
- 1.11.2オープンソースのコミュニティとエコシステム

https://linuc.org/linuc1/range/101.html
https://linuc.org/linuc1/range/102.html

LinuC レベル2

201試験

- 2.01：システムの起動とLinuxカーネル
  - 2.01.1 ブートプロセスとGRUB
  - 2.01.2 システム起動のカスタマイズ
  - 2.01.3 Linux カーネルの構成要素
  - 2.01.4 Linuxカーネルのコンパイル
  - 2.01.5 カーネル実行時における管理とトラブルシューティング
- 2.02：ファイルシステムとストレージ管理
  - 2.02.1 ファイルシステムの設定とマウント
  - 2.02.2 ファイルシステムの管理
  - 2.02.3 論理ボリュームマネージャの設定と管理
- 2.03：ネットワーク構成
  - 2.03.1 基本的なネットワーク構成
  - 2.03.2 高度なネットワーク構成
  - 2.03.3 ネットワークの問題解決
- 2.04：システムの保守と運用管理
  - 2.04.1 makeによるソースコードからのビルドとインストール
  - 2.04.2 バックアップとリストア
  - 2.04.3 ユーザへの通知
  - 2.04.4 リソース使用状況の把握
  - 2.04.5 死活監視、リソース監視、運用監視ツール
  - 2.04.6 システム構成ツール
- 2.05：仮想化サーバー
  - 2.05.1 仮想マシンの仕組みとKVM
  - 2.05.2 仮想マシンの作成と管理
- 2.06：コンテナ
  - 2.06.1 コンテナの仕組み
  - 2.06.2 Dockerコンテナとコンテナイメージの管理

202試験

- 2.07：ネットワーククライアントの管理
  - 2.07.1 DHCPサーバーの設定と管理
  - 2.07.2 PAM認証
  - 2.07.3 LDAPクライアントの利用方法
  - 2.07.4 OpenLDAPサーバーの設定
- 2.08：ドメインネームサーバー
  - 2.08.1 BINDの設定と管理
  - 2.08.2 ゾーン情報の管理
  - 2.08.3 セキュアなDNSサーバーの実現
- 2.09：HTTPサーバーとプロキシサーバー
  - 2.09.1 Apache HTTPサーバーの設定と管理
  - 2.09.2 OpenSSLとHTTPSの設定
  - 2.09.3 nginxの設定と管理
  - 2.09.4 Squidの設定と管理
- 2.10：電子メールサービス
  - 2.10.1 Postfixの設定と管理
  - 2.10.2 Dovecotの設定と管理
- 2.11：ファイル共有サービス
  - 2.11.1 Sambaの設定と管理
  - 2.11.2 NFSサーバーの設定と管理
- 2.12：システムのセキュリティ
  - 2.12.1 iptables や firewalld によるパケットフィルタリング
  - 2.12.2 OpenSSH サーバーの設定と管理
  - 2.12.3 OpenVPNの設定と管理
  - 2.12.4 セキュリティ業務
- 2.13：システムアーキテクチャ
  - 2.13.1 高可用システムの実現方式
  - 2.13.2 キャパシティプランニングとスケーラビリティの確保
  - 2.13.3 クラウドサービス上のシステム構成
  - 2.13.4 典型的なシステムアーキテクチャ

https://linuc.org/linuc2/range/201.html
https://linuc.org/linuc2/range/202.html

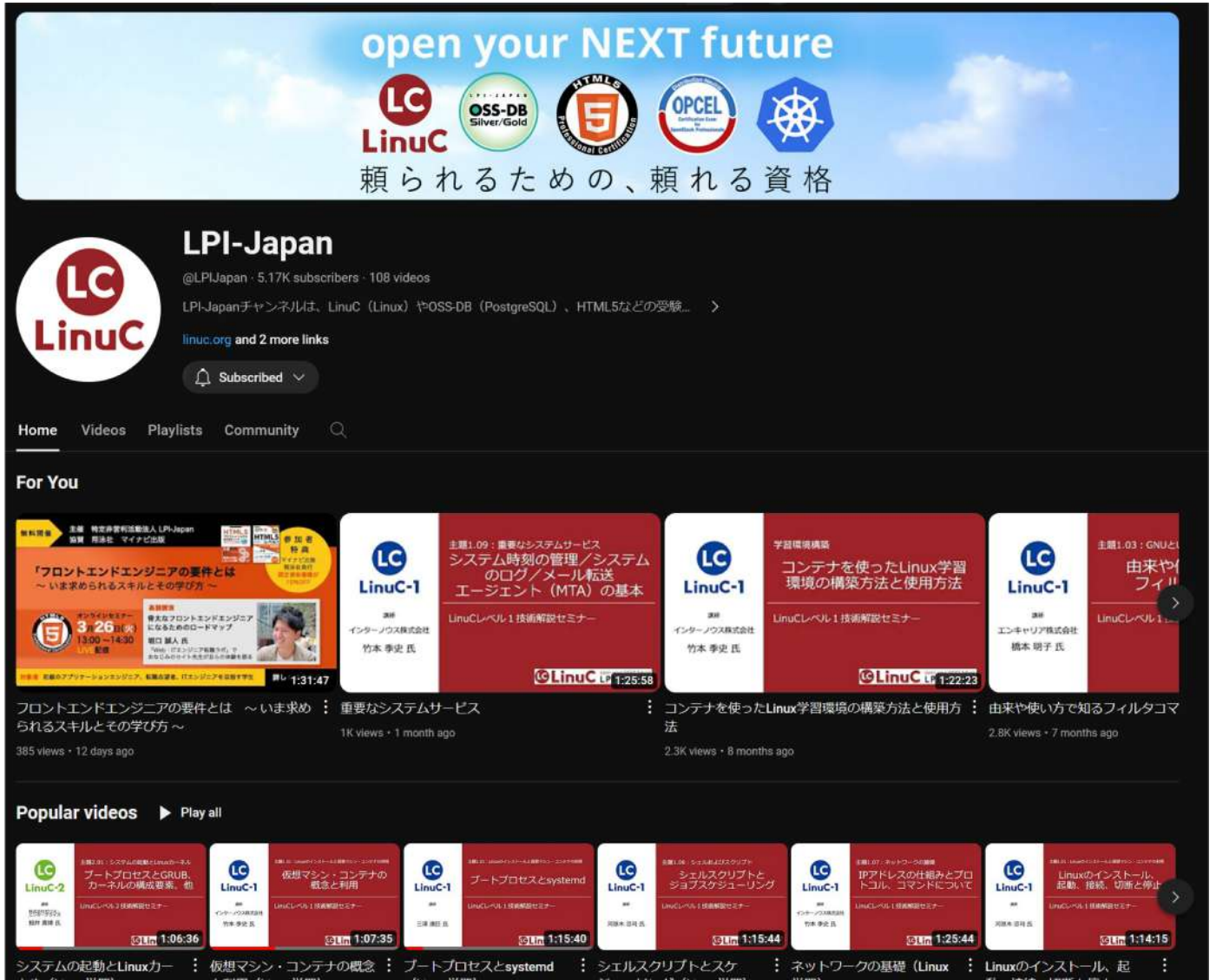
LinuC 学習のコツ

①出題範囲の内容について調べてみる
　　公式ドキュメント・技術書など
②実際に操作してみる
　　これが大事！
③学習の補助教材などを利用する
　・メールマガジン
　・標準教科書
　・過去のセミナー資料
　　詳細は、https://lpi.or.jp/learning/

LinuC 学習のコツ

④過去セミナーの動画

https://www.youtube.com/user/LPIJapan



LinuC 学習のコツ

学習の具体的な進め方(2~3か月程度)

学習開始(0%)

試験範囲の確認(LinuC HP)

●月頃、受験しようという目標を立てる

↓

LinuC認定教材の購入・1週目読込 https://lpi.or.jp/linuc1/book.shtml https://lpi.or.jp/linuc2/book.shtml

↓

LinuC認定教材・Webサイトを参考に、実機操作(サーバ構築やコマンド操作)を試す

※操作やトラブルシュートで力が身に付く！

1ヶ月

↓

LinuC認定教材 2週目読込

●月●日頃、受験しようと決める

↓

問題集やメルマガサンプル問題で理解力確認 https://linuc.org/study/samples/

※理解不足箇所の洗い出し

↓

LinuC認定教材 3週目

※弱点の補強

2ヶ月

↓

仕上げ(80%)

受験申込

受験日の変更も可能なので安心

↓

問題を8~9割以上、正解となるまで繰り返し解く

苦手な部分を重点的に復習

2.5ヶ月

↓

完全に理解した！(100%)

受験

・受験まで継続して学習すること
・繰り返し学習し、理解度/問題正解率を高めた状態で受験すること

2.12.2 OpenSSH サーバーの設定と管理

試験としての重要度(出題割合)は高く、
現場においても頻繁に利用されるため しっかりとした理解が必要

2.12.2 OpenSSH サーバーの設定と管理

重要度　4

概要　SSHデーモンの設定と保護ができる。これには、鍵の管理とユーザ用にSSHを設定することも含まれる。

詳細

OpenSSH サーバーの設定ファイルとデーモン

sshd, /etc/ssh/sshd_config

/etc/ssh/ssh_host_*_key および ssh_host_*_key.pub

スーパーユーザおよび一般ユーザのログインを制限する。

PermitRootLogin, PubKeyAuthentication, AllowUsers, PasswordAuthentication

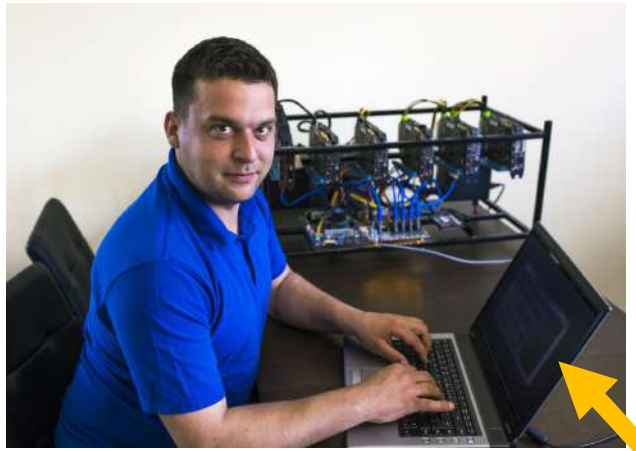
SSHとは

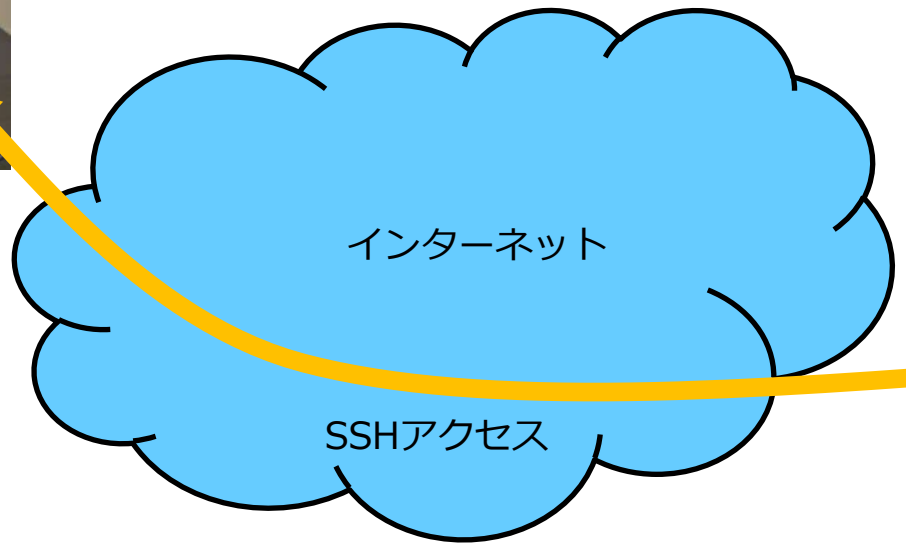
技術解説(SSHとは)

Secure SHellの略
何がSecureなのか？ →通信の暗号化
以前は、telnet(非暗号化 = 平文でのやりとり)であった

技術解説(SSHによるアクセス)

安全なリモートアクセスを提供
秘密鍵/公開鍵のペアによる認証
サーバへの通信暗号化





OpenSSHを使ってみる

技術解説(OpenSSHとは)

OpenSSHは、SSHに対応したソフトウェア
SSHプロトコルを使用してログイン、ファイル転送、リモートコマンドの実行が可能
オープンソースソフトウェアとして公開されており、多くのOSで使用されている
サーバデーモンは、sshd

https://www.openssh.com/

技術解説(OpenSSH環境の用意)

サーバプログラムとして、openssh-serverをインストール
通信元(クライアント)として、
sshコマンドやそれを実装したクライアントソフトウェアを利用する

```
ubuntu@ubuntu: $ dpkg -l | grep ssh
ii  libssh-4:amd64                      0.9.6-2ubuntu0.22.04.3          amd64        tiny C SSH library (OpenSSL flavor)
ii  openssh-client                      1:8.9p1-3ubuntu0.6              amd64        secure shell (SSH) client, for secure access to remote machines
ii  openssh-server                      1:8.9p1-3ubuntu0.6              amd64        secure shell (SSH) server, for secure access from remote machines
ii  openssh-sftp-server                 1:8.9p1-3ubuntu0.6              amd64        secure shell (SSH) sftp server module, for SFTP access from remote machi
nes
ii  ssh-import-id                       5.11-0ubuntu1                   all          securely retrieve an SSH public key and install it locally
```

技術解説(OpenSSHの用意)

SSHクライアントソフトウェア

- sshコマンド(CLI)
- TeraTerm
- rlogin
- Putty
- scpコマンド・winscp ※sshを使ったファイル転送

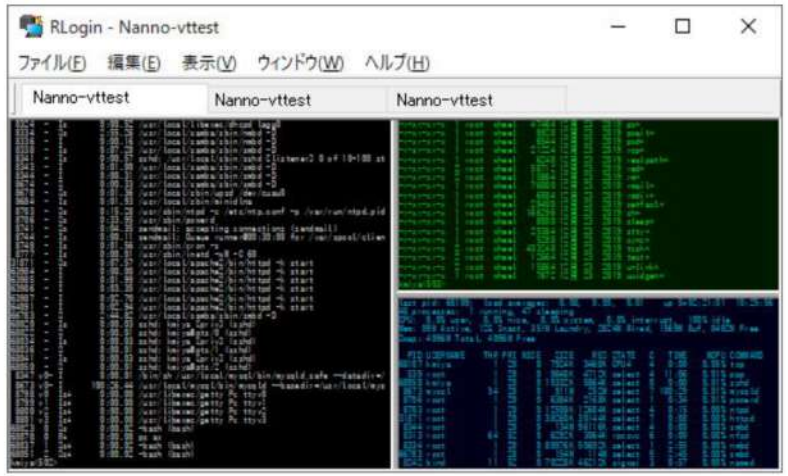
RLogin (2.29.2) 2024/03/21

RLoginは、Windows上で動作するターミナルソフトです

プロトコルはrlogin,telnet,ssh(バージョン１と２)の3種類に対応し遠隔でのサーバーメンテナンスを考えて安全な暗号化通信をサポートしています

漢字コードは、EUC,SJIS,UTF-8などに対応しISO-2022によるバンク切り替えで様々な文字コードが表示できます

画面制御としてxtermに準じたエスケープシーケンスなどに対応しANSIやvt100コンソールとして使用する事ができます



技術解説(sshコマンドの使い方)

ssh オプション username@SSHサーバのIPアドレス

オプション

-4：IPv4のみを使用

-6：IPv6のみを使用

-p：ポート番号指定

```
C:\>ssh ubuntu@192.168.1.115
ubuntu@192.168.1.115's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-102-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro
Last login: Mon Apr 15 06:40:44 2024 from 192.168.1.122
ubuntu@ubuntu:~$
```

技術解説(scpコマンドの使い方)

scp オプション username@SSHサーバのIPアドレス:ファイルのパス ファイル置場所

※サーバ上にあるファイルをローカルにコピーする例

オプション

-P：ポート番号指定

```
PS C:\Users\kujir\ssh> scp -P 10022 ubuntu@192.168.1.115:/home/ubuntu/.ssh/id_ed25519 .
ubuntu@192.168.1.115's password:
id_ed25519                                                    100%  399   137.2KB/s   00:00
```

OpenSSHの設定

技術解説(OpenSSH設定ファイルの概要)

OpenSSHのサーバー設定は、 /etc/ssh/sshd_config ファイルで設定する

```
ubuntu@ubuntu:~$ cat /etc/ssh/sshd_config

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO
```

技術解説(OpenSSH設定ファイルの概要)

ディレクティブ	設定内容	設定例
Port	OpenSSHが公開するポート番号	22、10022
Protocol	サポートするバージョン	2
ListenAddress	接続を受け付けるアドレス指定	0.0.0.0 ※全てのアドレス
HostKey	秘密鍵の指定	/etc/ssh/ssh_host_ed25519_key
LOGLEVEL	ログの出力レベル指定	INFO・DEBUG・ERROR
PermitRootLogin	rootユーザのログイン可否	yes・no・without-password・forced-commands-only
MaxAuthTries	1接続当たりの認証の試行回数	6
MaxSessions	接続ごとに許可されるセッション数	10
AuthorizedKeysFile	ユーザ認証に使われる公開鍵指定	.ssh/authorized_keys .ssh/authorized_keys2
UsePAM	PAMによる認証の許可	yes・no
Include	外部設定ファイルの参照先	/etc/ssh/sshd_config.d/*.conf
PasswordAuthentication	パスワード認証の許可	yes・no
AllowUsers	許可するユーザを指定	ユーザ名
PubKeyAUthentication	公開鍵認証の許可	yes・no

技術解説(sshdデーモンの操作)

起動

systemctl start sshd

停止

systemctl stop sshd

再起動 ※設定変更後など

systemctl restart sshd

ステータス確認

systemctl status sshd

```
ubuntu@ubuntu: $ systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-04-15 06:40:19 UTC; 29s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 719 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 787 (sshd)
      Tasks: 1 (limit: 4557)
     Memory: 5.3M
        CPU: 379ms
     CGroup: /system.slice/ssh.service
             └─787 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Apr 15 06:40:18 ubuntu systemd[1]: Starting OpenBSD Secure Shell server...
Apr 15 06:40:19 ubuntu sshd[787]: Server listening on 0.0.0.0 port 22.
Apr 15 06:40:19 ubuntu sshd[787]: Server listening on :: port 22.
Apr 15 06:40:19 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
Apr 15 06:40:43 ubuntu sshd[1176]: Accepted password for ubuntu from 192.168.1.122 port 49369 ssh2
Apr 15 06:40:43 ubuntu sshd[1176]: pam_unix(sshd:session): session opened for user ubuntu(uid=1000) by
```

技術解説(OpenSSHの鍵の種類)

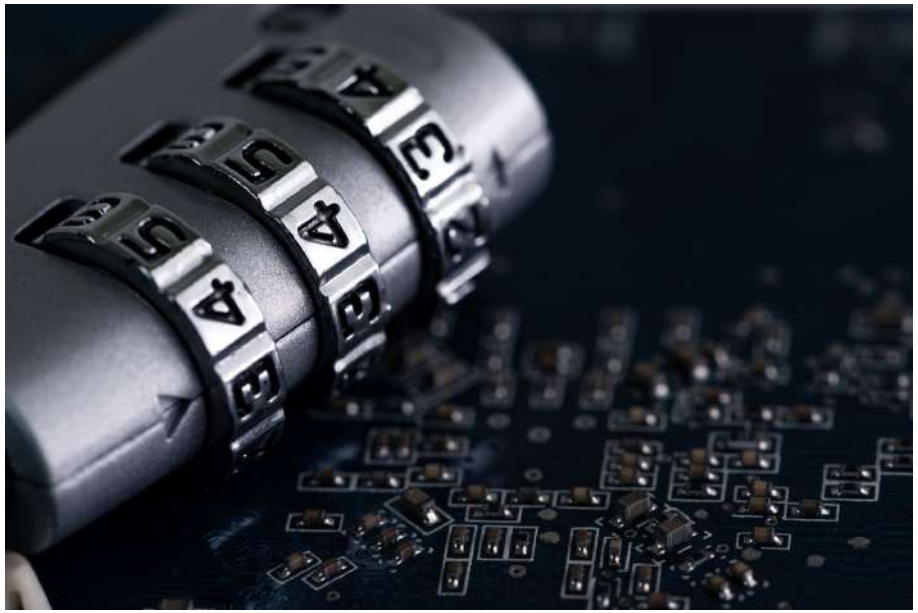
DSA: 桁数の大きい半素数の素因数分解が非常に難しいという性質を利用
RSA: 離散対数問題の困難性に基づく
ECDSA: 楕円曲線暗号を利用
ED25519: エドワーズ曲線を利用 ※デフォルト
ECDSA-SK：ECDSA + セキュリティキーを使用
ED25519-SK：ED25519 + セキュリティキーを使用

セキュリティキーとは？ (ECDSA-SK・ED25519-SK)
　USBデバイスによる指紋認証などハードウェア認証を併用するもの

共通鍵暗号方式と公開鍵暗号方式

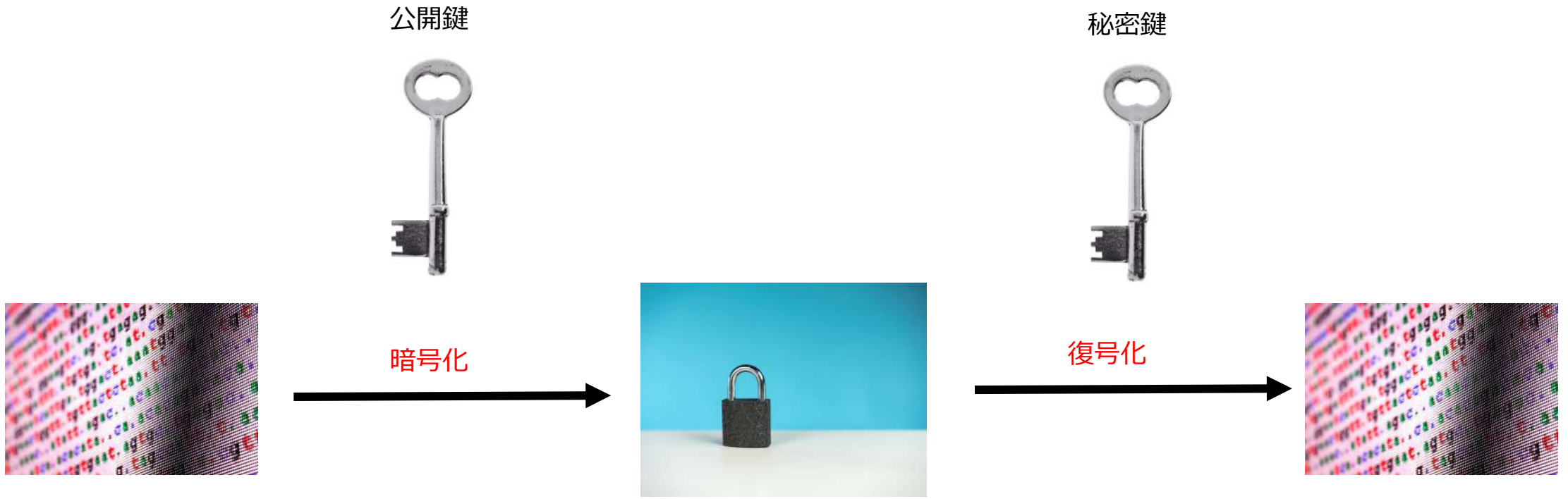
技術解説(共通鍵暗号方式とは)

共通鍵暗号方式とは、暗号化と復号化に同じ鍵を使用する方式
受信者と送信者は、共有の鍵を所有している必要がある
データの暗号化には、共通鍵アルゴリズムが使用される
共通鍵の強度が重要



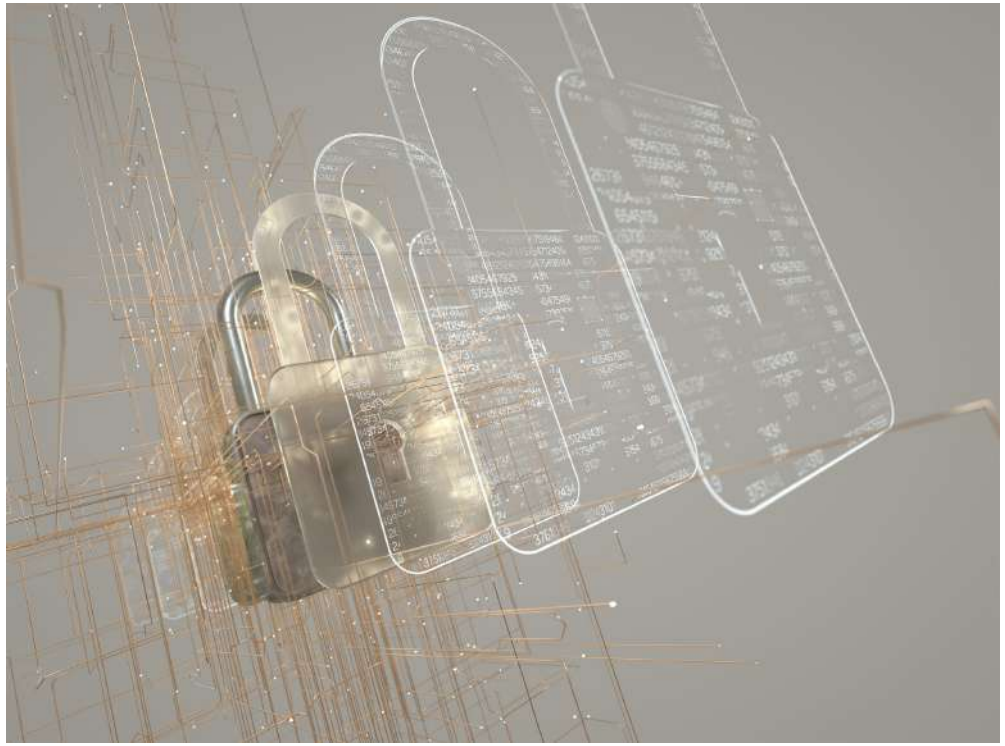
技術解説(公開鍵暗号方式とは)

暗号化と復号化に異なる鍵を使用する
公開鍵は誰でも使用できる一方、秘密鍵は非公開で管理する



技術解説(公開鍵暗号方式のメリット・デメリット)

暗号化データと公開鍵が流出しても、秘密鍵がなければ復号出来ない
秘密鍵の管理が重要
データの送信者と受信者が多い場合、鍵の管理が複雑になる



技術解説(キーペアの作成方法)

ssh-keygen -t 鍵の種類[dsa/rsa/ecrsa/ed25519]

```
ubuntu@ubuntu:~/.ssh$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:655rmPD5yLLmfS/g8ciScSFsKYkkzuD6stfHb8L32fE ubuntu@ubuntu
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|o.               |
|B. o .           |
|.+o = .          |
|.  o . .S        |
|.   o +  .       |
| . . @ O.   .    |
|. o *o&oB. o o   |
|.+ oo=+XB=+ . E  |
+----[SHA256]-----+

ubuntu@ubuntu:~/.ssh$ ls
authorized_keys  id_dsa.pub  id_ecdsa.pub  id_ed25519.pub  id_rsa.pub
id_dsa           id_ecdsa    id_ed25519    id_rsa
```

https://linux.die.net/man/1/ssh-keygen

技術解説(作成したキーペアを使ったsshアクセス)

①キーペアの作成

②設定(sshd_config)を変更　※sshdへの反映もお忘れなく

PasswordAuthentication no

PubKeyAuthentication yes

③公開鍵(id_ed25519.pub)の内容をauthorized_keysにコピー

```
ubuntu@ubuntu:~/.ssh$ cat id_ed25519.pub
ssh-ed25519
  AAAAC3NzaC1lZDI1NTE5AAAAICj12/EeJt5AG2RZXITcC8XwO1VQRcZsCyUmwRAhsKyT
  ubuntu@ubuntu
ubuntu@ubuntu:~/.ssh$ cat id_ed25519.pub >> authorized_keys
```

④秘密鍵(id_ed25519)をクライアントへ渡す

⑤クライアントからサーバへssh実施

技術解説(脆弱性に要注意)

緊急度が高い脆弱性が確認された場合には、対応が必要

JVN iPedia 脆弱性対策情報データベース

JVN iPedia English Version

脆弱性対策情報データベース検索

検索キーワード： openssh 検索 検索の使い方

類義語： ☑

ベンダ名/製品名検索

ベンダ名：

製品： - ベンダを選択してください -

公表日： 年 月～ 年 月

最終更新日： 年 月～ 年 月

深刻度(CVSSv3)： □緊急：(9.0～10.0) □重要：(7.0～8.9) □警告：(4.0～6.9) □注意：(0.1～3.9) □なし：(0)

深刻度(CVSSv2)： □危険：(7.0～10.0) □警告：(4.0～6.9) □注意：(0.0～3.9)

CWE：

共通脆弱性タイプ一覧（CWE）とは

※「ベンダ名/製品名検索」ボタンは、Microsoft Edge(IEモード)でのみご利用いただけます。

114件中1～100件表示中 1 2 →

ID	タイトル	CVSSv3	CVSSv2	公表日	最終更新日
JVNDB-2023-023721	openssh の openssh 等複数ベンダの製品における脆弱性	7.0	-	2023/12/24	2024/01/29
JVNDB-2023-020905 (JVNVU#99836374)	OpenBSD の OpenSSH 等複数ベンダの製品におけるデータの整合性検証不備に関する脆弱性	5.9	-	2023/12/18	2024/01/17
JVNDB-2023-020641 (JVNVU#98271228)	OpenBSD の OpenSSH 等複数ベンダの製品における引用されない検索パスまたは要素に関する脆弱性	9.8	-	2023/07/20	2024/01/17
JVNDB-2023-020261	OpenBSD の OpenSSH における脆弱性	5.5	-	2023/12/18	2024/01/16
JVNDB-2023-020260	OpenBSD の OpenSSH 等複数ベンダの製品における OS コマンドインジェクションの脆弱性	6.5	-	2023/12/18	2024/01/16
JVNDB-2023-005684 (JVNVU#98271228)	OpenBSD の OpenSSH 他複数ベンダの製品における脆弱性	9.8	-	2023/03/17	2023/12/21

https://jvndb.jvn.jp/

技術解説(脆弱性に要注意)

公開された脆弱性の内容やその対策をチェック

「OpenSSH」に脆弱性、アップデートがリリース

「OpenSSH」に脆弱性が明らかとなった。特定の条件が重なるとリモートよりコードの実行が可能になるとしており、「OpenSSH」の開発チームは、脆弱性を修正したアップデートを提供している。

過去に修正された「CVE-2016-10009」の修正が不十分だったことに由来する脆弱性「CVE-2023-38408」が明らかとなったもの。Qualysの研究チームが発見、報告した。

SSHエージェント転送を利用している場合に影響があり、「PKCS#11」のサポートによってディストリビューションにおいて共有ライブラリが保存されているディレクトリを読み込むことができることに起因している。

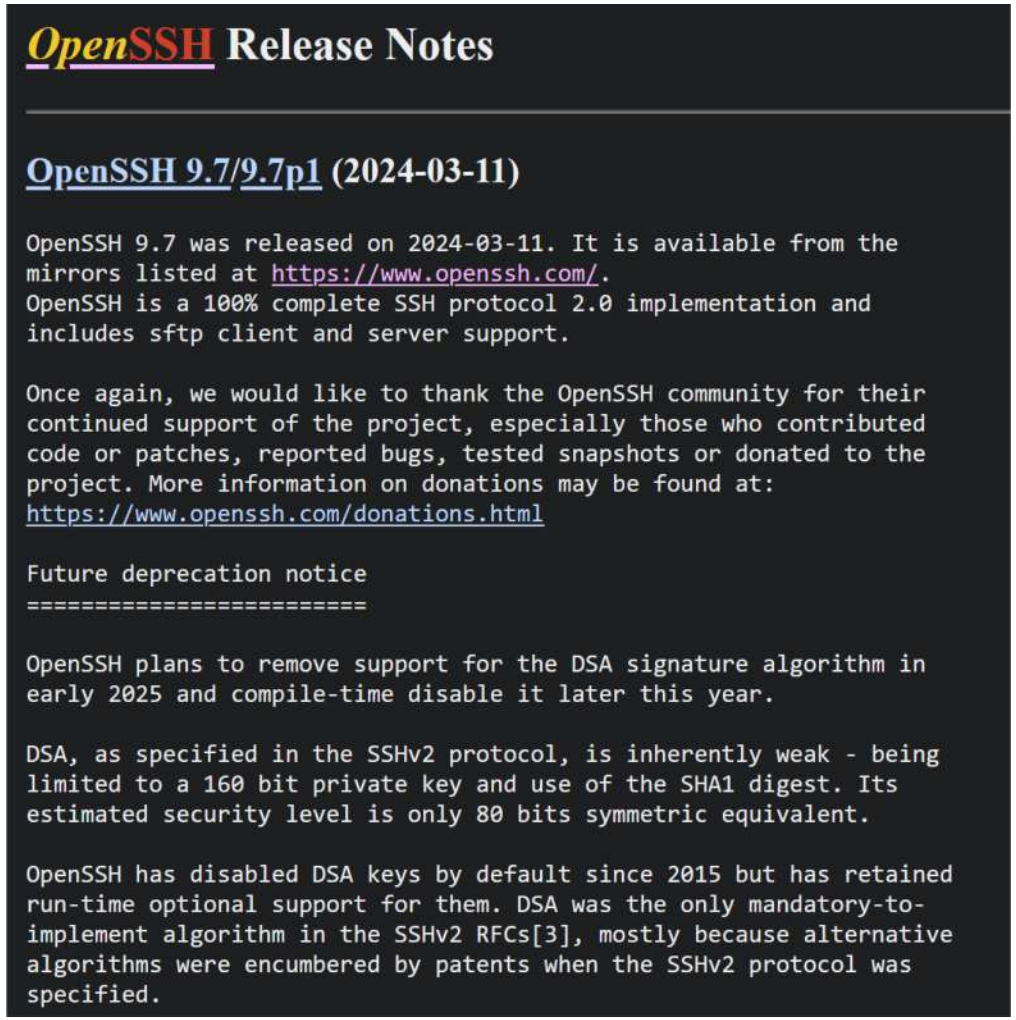
システム内で利用するパッケージに依存するが、複数の特定ライブラリが同ディレクトリ内に存在する場合、リモートより任意のコマンドを実行することが可能としており、同社では、一部パッケージを追加導入したUbuntuの特定バージョンで動作する実証コード（PoC）の作成にも成功したとしている。

同社では、現地時間7月6日にOpenSSHの開発チームへ初期のパッチを提供。その後連携しつつ対応を進めていた。開発チームは、現地時間7月19日に同脆弱性を修正したアップデート「OpenSSH 9.3p2」をリリース。許可リストを指定するなど緩和策についてもアナウンスしている。

（Security NEXT - 2023/07/26 ）

https://www.security-next.com/148143

本家(openssh)サイトの更新(リリース)情報もチェック



OpenSSH Release Notes

OpenSSH 9.7/9.7p1 (2024-03-11)

OpenSSH 9.7 was released on 2024-03-11. It is available from the mirrors listed at https://www.openssh.com/.
OpenSSH is a 100% complete SSH protocol 2.0 implementation and includes sftp client and server support.

Once again, we would like to thank the OpenSSH community for their continued support of the project, especially those who contributed code or patches, reported bugs, tested snapshots or donated to the project. More information on donations may be found at:
https://www.openssh.com/donations.html

Future deprecation notice
==========

OpenSSH plans to remove support for the DSA signature algorithm in early 2025 and compile-time disable it later this year.

DSA, as specified in the SSHv2 protocol, is inherently weak - being limited to a 160 bit private key and use of the SHA1 digest. Its estimated security level is only 80 bits symmetric equivalent.

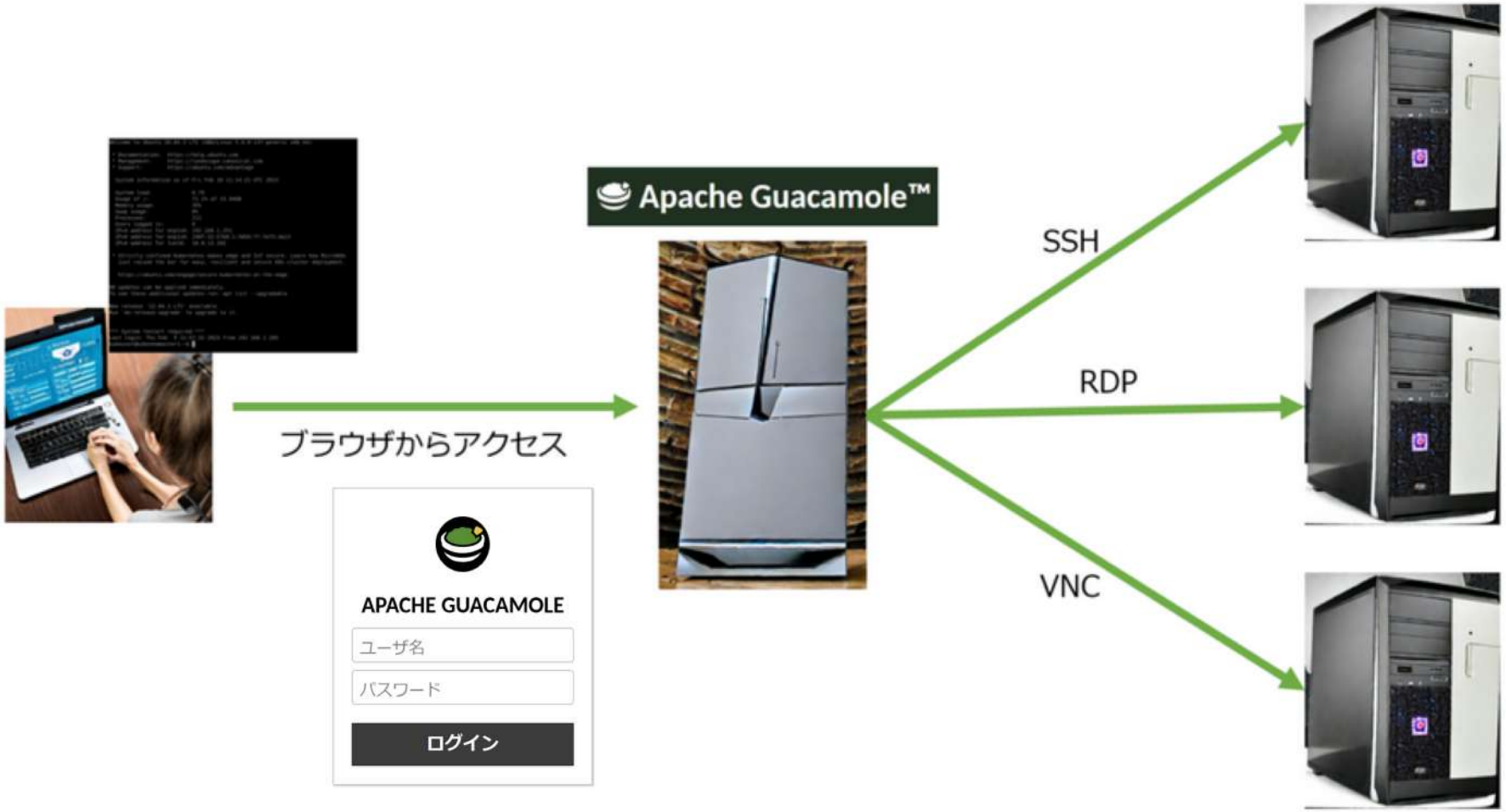
OpenSSH has disabled DSA keys by default since 2015 but has retained run-time optional support for them. DSA was the only mandatory-to-implement algorithm in the SSHv2 RFCs[3], mostly because alternative algorithms were encumbered by patents when the SSHv2 protocol was specified.

https://www.openssh.com/releasenotes.html

Appendix

Appendix(踏み台サーバを使ったsshアクセス)

リモートアクセス(SSH/RDPなど)の踏み台サーバ Apache Guacamoleを使ってみる



https://www.opensourcetech.tokyo/entry/20230210/1676032369

Appendix

LinuCレベル2 202試験
主題2.12.2の例題と解説

https://linuc.org/study/samples/4964/

LinuCレベル2 202試験の例題と解説

2.12.2 OpenSSH サーバーの設定と管理

いいね！ 4　シェアする　X ポスト　B! 0

LinuCレベル2 202試験の出題範囲から「2.12.2 OpenSSH サーバーの設定と管理」についての例題を解いてみます。
SSHでのログイン制限について学んでいきましょう。

Linucレベル2 202試験 出題範囲

例題

/etc/ssh/sshd_configに以下の設定がされている。

```
PermitRootLogin yes
AllowUsers admin
PasswordAuthentication no
PubkeyAuthentication yes
```

この時の動作として正しいものを一つ選択してください。

1. adminユーザのみがパスワード認証でログインすることができる
2. adminユーザとrootユーザがパスワード認証でログインすることができる
3. adminユーザのみが公開鍵認証でログインすることができる
4. adminユーザとrootユーザが公開鍵認証でログインすることができる

※この例題は実際の試験問題とは異なります。

お知らせ

ゼウス・エンタープライズの提供サービス

Network Engineering Service

高い専門スキルを有するエンジニア集団だから可能な質の高いソリューション

ゼウス・エンタープライズは、時代変革の要となるネットワーク・セキュリティ分野に特化したエンジニア集団として、顧客のニーズや課題に迅速かつ確実に応え、満足度の高いIT支援サービスを提供しています。情報通信・官公庁・金融・製造などの様々なクライアント先にて TCP/IPスタックの機器や、Linuxにおける豊富な経験と高度な技術を活用し、ネットワークやセキュリティ分野のパフォーマンスを最大限に引き出します。
主な業務としては、小規模LANから大規模WANまでのネットワーク構築や運用支援。各種アプリケーションの実行基盤やデータベースなど業務サーバーの構築や運用支援。また、オンプレ環境やクラウド環境、ハイブリッドクラウドの環境においても、セキュリティを重視した構築や運用支援を提供しています。
そして、当社は活躍する社員一人ひとりの能力を昇華させるべく、「ゼウスITトレーニングセンター」という教育機関を併設しており、ネットワークやLinuxを中心に、時代のトレンドに沿ったインフラ教育を行っています。
日々変革を遂げるIT業界に伴い、研修にて社員のスキルを底上げし、ネットワーク・セキュリティに特化したスペシャリスト集団として、クライアントの課題解決に貢献いたします。

リナックス・ネットワークに強いITスクール

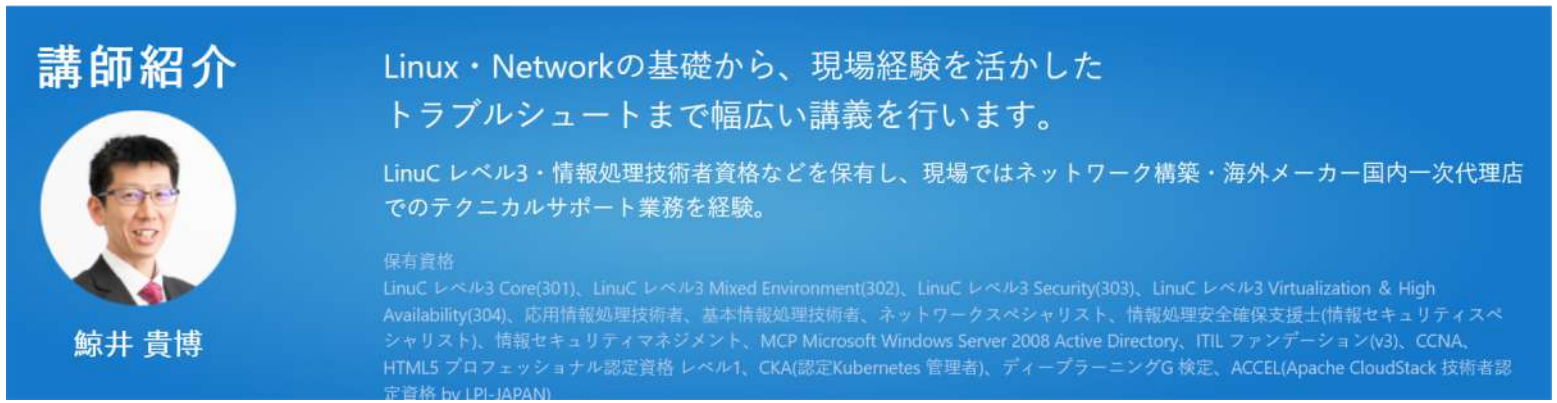
IT Training

未経験者を戦力に育て上げた独自のカリキュラムに定評があるITキャリアスクールです

LPI-Japanのアカデミック認定校であるITキャリアスクール「ゼウスITトレーニングセンター」を運営し、リナックスとネットワークに強いエンジニアを育成します。当社社員の研修カリキュラムを基にしているため、プロの技術者だけでなく未経験者までも現場ですぐに活躍できる人材に育てます。セキュリティ分野から開発分野まで、クライアントの要望に合わせて講座を選択できるよう、幅広いコースを展開しています。

https://www.zeus-enterprise.co.jp/solution/service/
https://www.it-training.tokyo/

Kubernets研修、やってます！



https://www.it-training.tokyo/kubernetes/

お知らせ

技術を楽しみながら、一緒に働ける仲間を募集中！

整った職場環境、充実した制度で
「一生働ける会社」を実現します

ゼウス・エンタープライズは、全社員の終身雇用を目指し、働きやすい社内環境作りに力を入れています。「ゼウスITトレーニングセンター」を研修拠点として、経験の有無を問わず、着実な技術の習得に導きます。また、独自の福利厚生制度を導入し、社員の健康的・文化的な生活を支援します。「安心できる環境で長く働きたい」「確かなスキルを身につけて活躍したい！」「ワークライフバランスを大切にしたい！」という方のエントリーをお待ちしています。

Pick up 現在募集中の職種
①ネットワークエンジニア 新卒 キャリア
②バイリンガルコーディネータ 新卒 キャリア

新卒採用応募事項 ENTRY →
キャリア採用応募事項 ENTRY →

会社説明会 →

https://www.zeus-enterprise.co.jp/recruit/

本日のまとめ

- OpenSSHの設定を理解する
- 鍵の管理を理解する
- ログイン制限を理解する

OpenSSHの設定ファイル(sshd_config)のディレクティブとその意味を知る

キーペアの作成方法を理解し・鍵の配布などは慎重に

セキュリティは大事！

Q & A

Thank you for joining my seminar!

https://www.opensourcetech.tokyo/

https://twitter.com/matt_zeus

https://www.facebook.com/takahiro.kujirai.1