

～クラウドサービス時代を支えるOSS/Linux人材育成～

Skill Brain

スキルブレイン株式会社

LPI-JAPAN

Linux Professional Institute Japan

LPIC303技術解説無料セミナー



LPI-Japanアカデミック認定校
スキルブレイン株式会社
インストラクター 三浦 一志



■ 主題

- 主題325: 暗号化
- 主題326: ホストセキュリティ
- 主題327: アクセス制御
- 主題328: ネットワークセキュリティ



■ 303試験がVer2.0に改訂

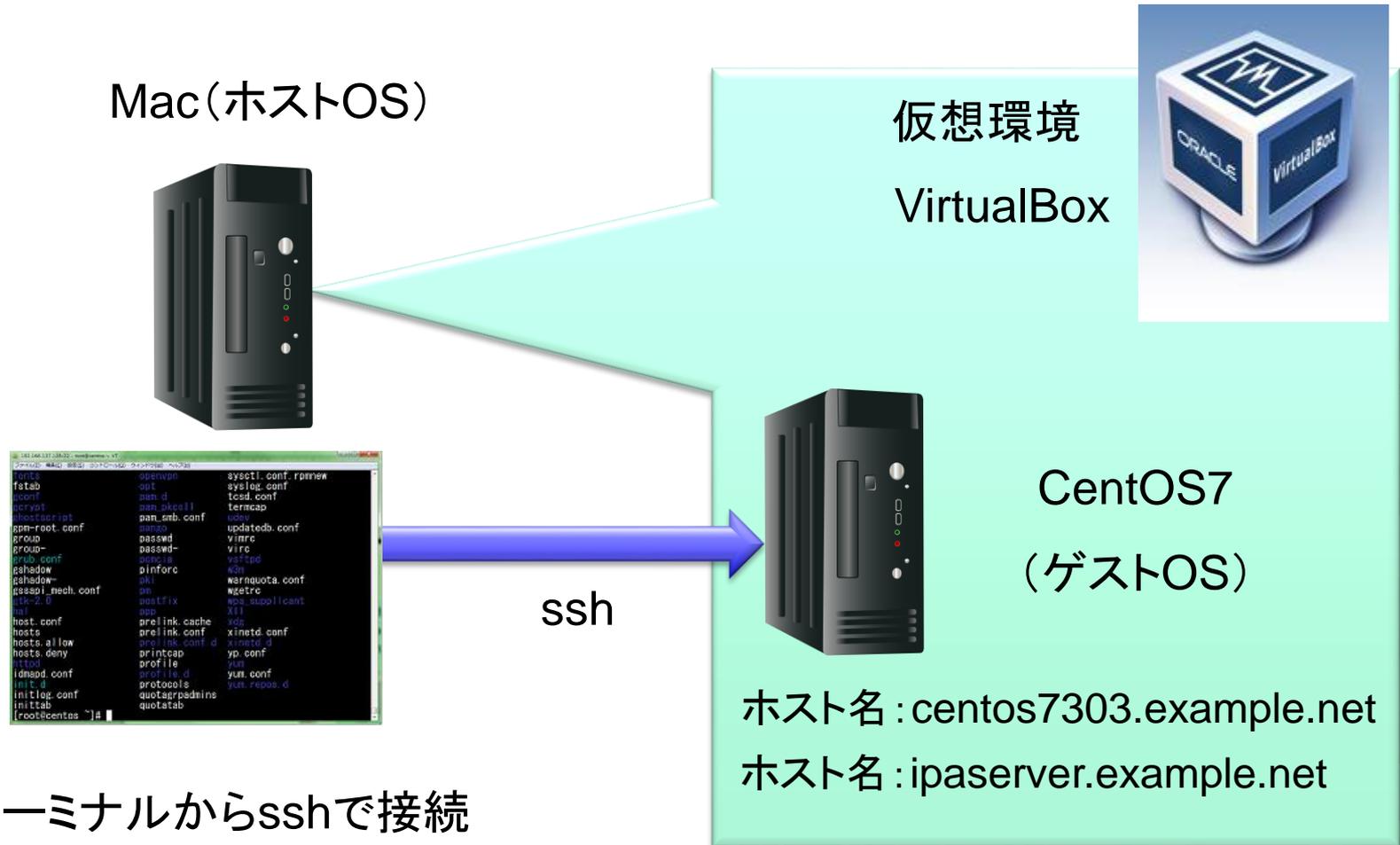
- 2016年3月1日から開始

■ 主な変更点

- OpenSSL証明書およびX.509証明書の拡張
- DNSのセキュリティについて。DNSSECとDANEに関する知識
- 暗号化ファイルシステムeCryptfsの追加
- 侵入検知および監視にOpenVASを追加
- パケットフィルタリングの範囲を拡張。IPV6、etables、nftablesおよびnft
- ホストの構成管理(Puppet)およびPGPは範囲外に



- 各主題のポイントとなる部分を紹介
- 仮想環境を利用し、デモで確認を行う



ターミナルからsshで接続



- 325.1 X.509 証明書と公開鍵の基礎 (重要度: 5)
- 325.2 暗号化、署名および認証のX.509 証明書 (重要度: 4)
- 325.3 暗号化ファイルシステム (重要度: 3)
- 325.4 DNS と暗号化 (重要度: 5)



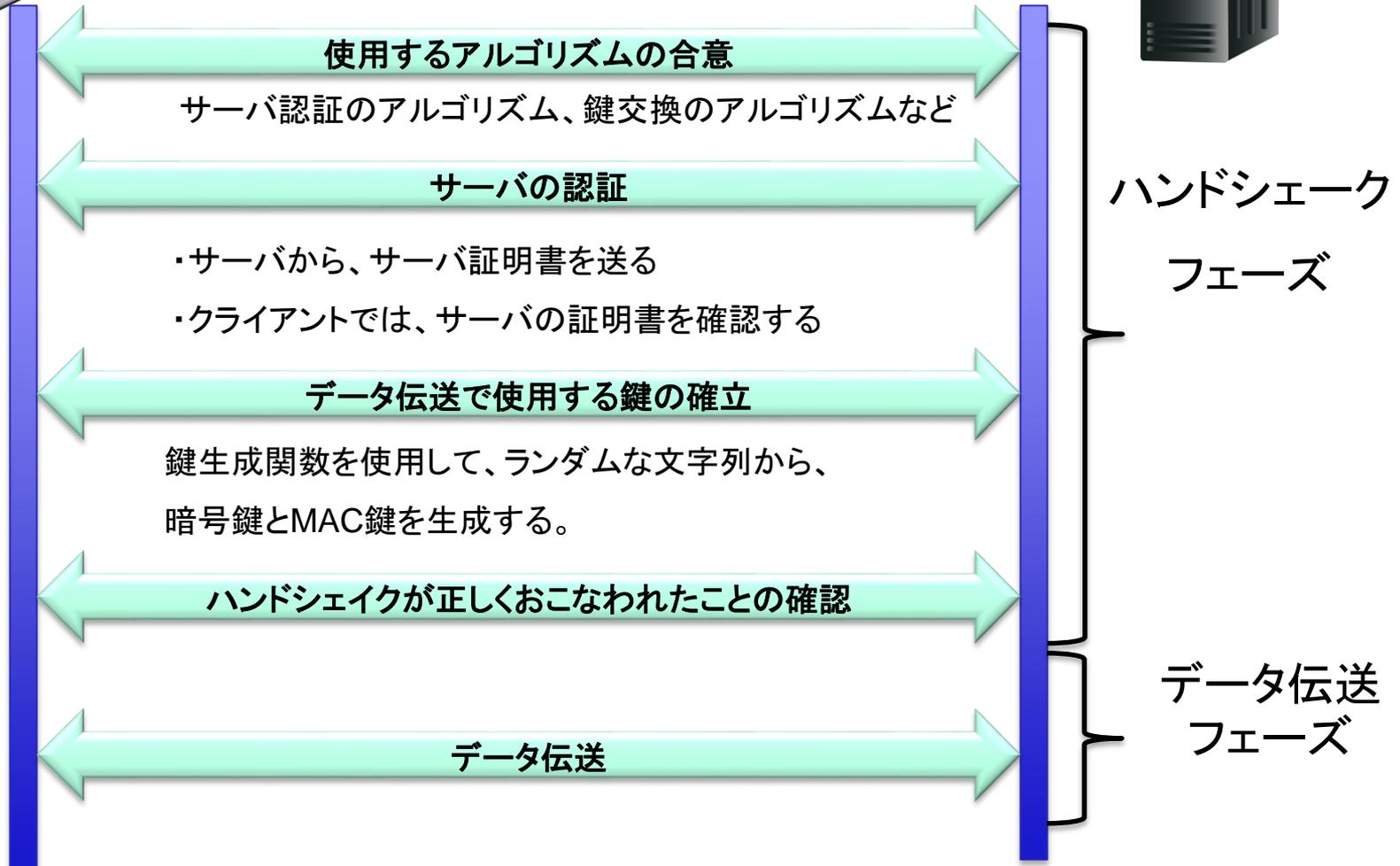
- **SSL (Secure Sockets Layer)**はセキュリティーを要求される通信を行うためのプロトコル
 - IETFではTLS (Transport Layer Security) でインターネット標準とされている
- **主な機能**
 - 通信相手の認証
 - 通信内容の暗号化
 - 改竄の検出など
- **OpenSSL**
 - SSLプロトコル・TLSプロトコルの、オープンソースで開発・提供されるソフトウェア
 - サポート: SSL 2.0、3.0、TLS 1.0、1.1、1.2、DTLS 1.0、1.2
 - 暗号方式: DES、RC2、RC4、RC5、SEED、IDEA、AESなど
 - ハッシュ関数: MD5、MD2、SHA-1、SHA-2、MDC-2
 - 公開鍵暗号方式: RSA暗号、DSA、Diffie-Hellman鍵共有



クライアント



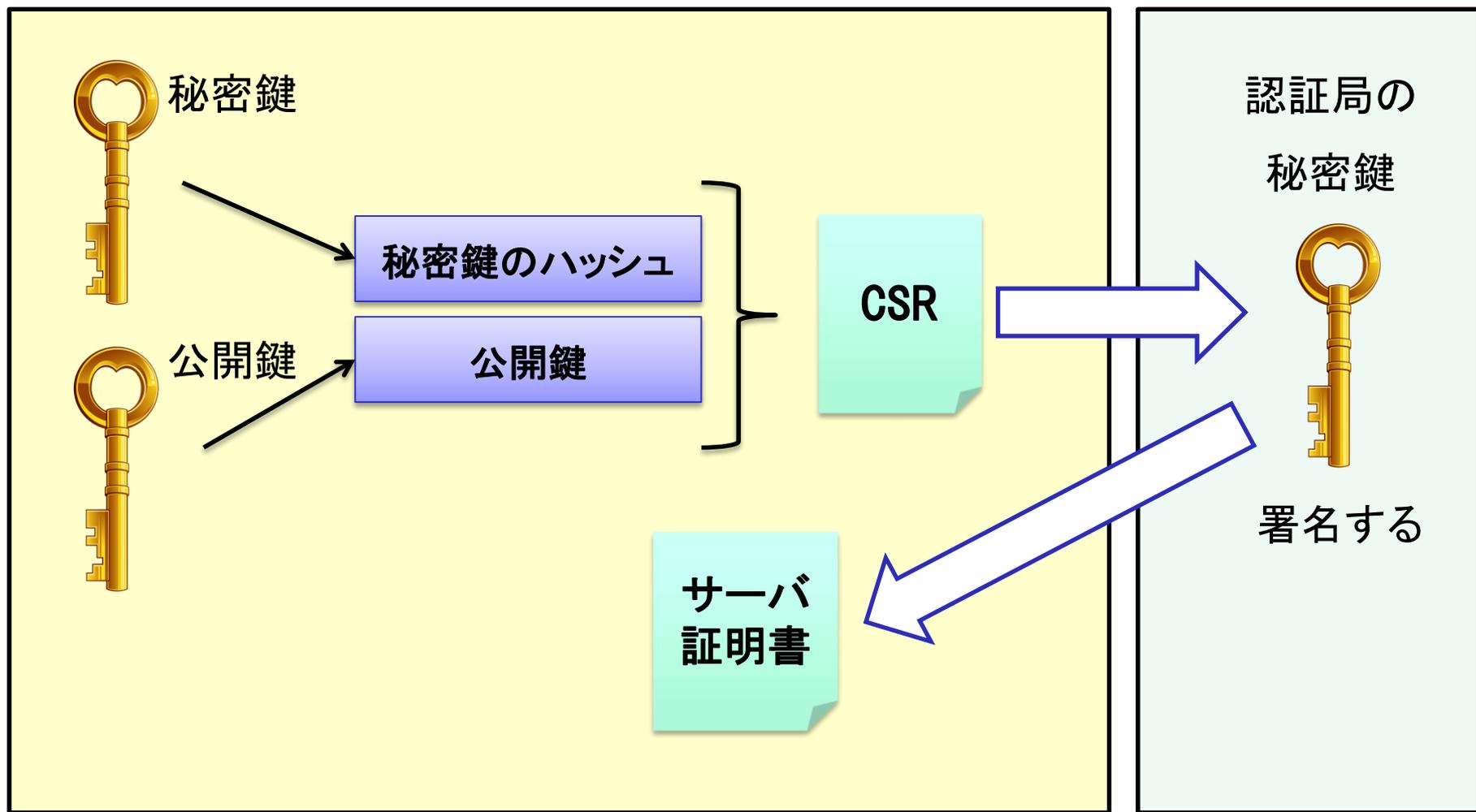
サーバ





サーバ

認証局





- /etc/pki/tls/openssl.cnfを確認

```
basicConstraints=CA:FALSE
```

- 秘密鍵の生成 (RSA形式)

```
# openssl genrsa -des3 -out privkey.key 2048
```

- 秘密鍵から公開鍵を生成する

```
# openssl rsa -in privkey.key -pubout > pubkey.key
```

- 署名リクエスト CSR (Certificate Signing Request) の作成

```
# openssl req -new -key privkey.key -out newreq.pem -days 365
```

- サーバ証明書の作成

```
# openssl ca -out newcert.pem -infiles newreq.pem
```

- サーバ証明書の失効

```
# openssl ca -revoke newcert.pem
```



■ Apacheにサーバ証明書を組み込む

/etc/httpd/conf.d/ssl.conf

SSLEngine on 有効化
SSLProtocol all -SSLv2 -SSLv3 接続可能なバージョンを指定
SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5:!SEED:!IDEA 接続可能な暗号スイート
SSLCertificateFile /etc/httpd/conf.d/newcert.pem サーバ証明書のパス
SSLCertificateKeyFile /etc/httpd/conf.d/privkey.key 秘密鍵のパス

※暗号スイートの前の「!」は除くという意味



暗号リスト	説明
aNULL	認証を提供しない暗号スイート
HIGH MEDIUM	共通鍵暗号方式の鍵長が128ビットより大きい
IDEA	共通鍵暗号方式にIDEAを使用する
MD5	ハッシュ関数にMD5を使用する
SEED	共通鍵暗号方式にSEEDを使用する
SHA1	ハッシュ関数にSHA1を使用する

■ 利用可能な暗号スイートを確認する

```
# openssl ciphers -v 'HIGH:MEDIUM:!aNULL:!SEED:!IDEA:!SHA1'
```



■ Webサーバのテスト

Webサーバとして動作させる

```
# openssl s_server -cert cert.pem -key private.key -WWW
```

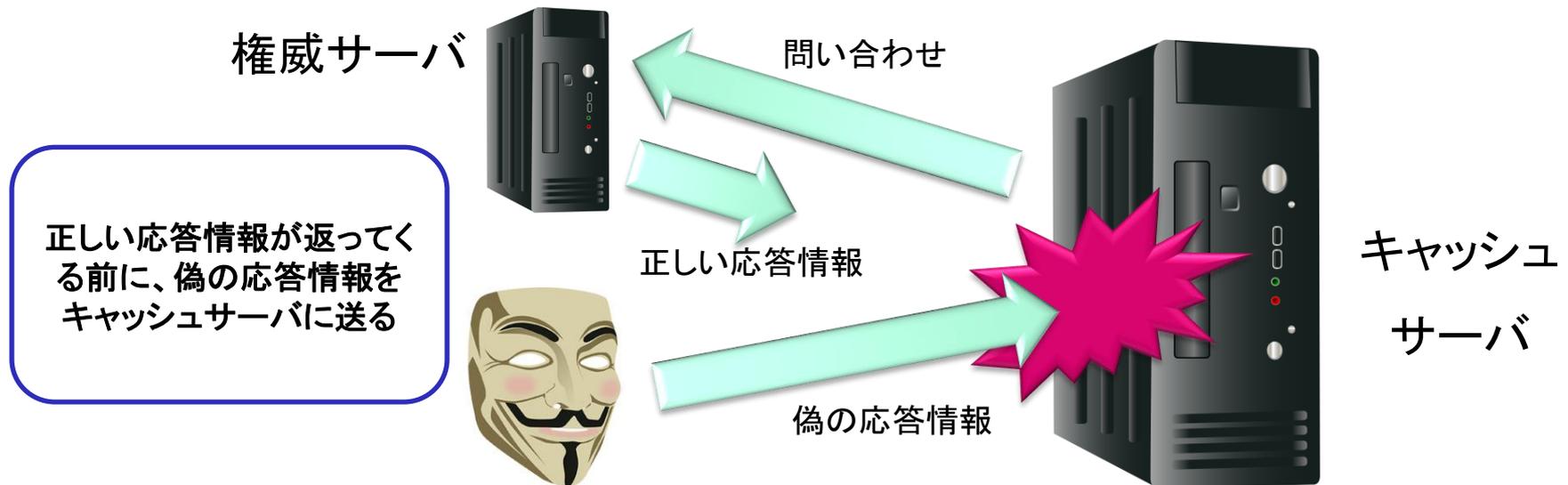
(ポート番号の指定がないと4433を使用する)

■ クライアントテスト:サーバのポート443に接続する

```
# openssl s_client -connect centos.example.net:443
```



- DNS通信はUDPを利用しており、「問い合わせ」と「応答」の1セッションで終了する
- 識別には「送信元IPアドレス」「ポート番号」「クエリ名」「ID」を使用している
- 送信元IPアドレスの詐称がしやすく、IDさえわかれば偽装した応答パケットをキャッシュサーバに送ることができる
- 結果として、キャッシュポイズニング(毒入れ)の影響を受けやすい





■ DNSSEC (DNS Security Extensions)とは

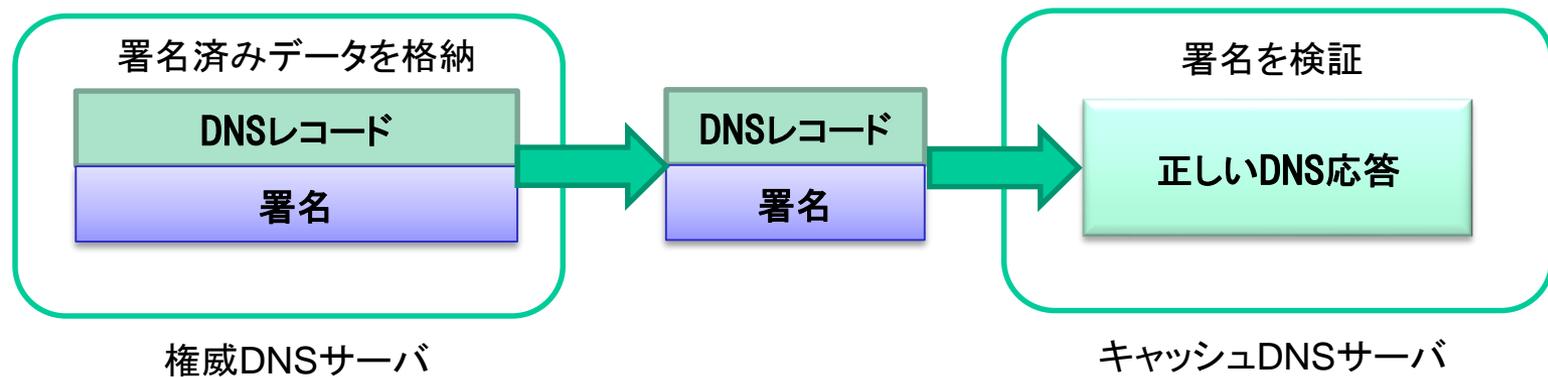
DNS応答が正しいものかどうか検証することで、DNSのセキュリティを向上させる仕組み

■ 出自の認証

DNSの応答が、ドメイン名の正当な管理者が作成したものであること

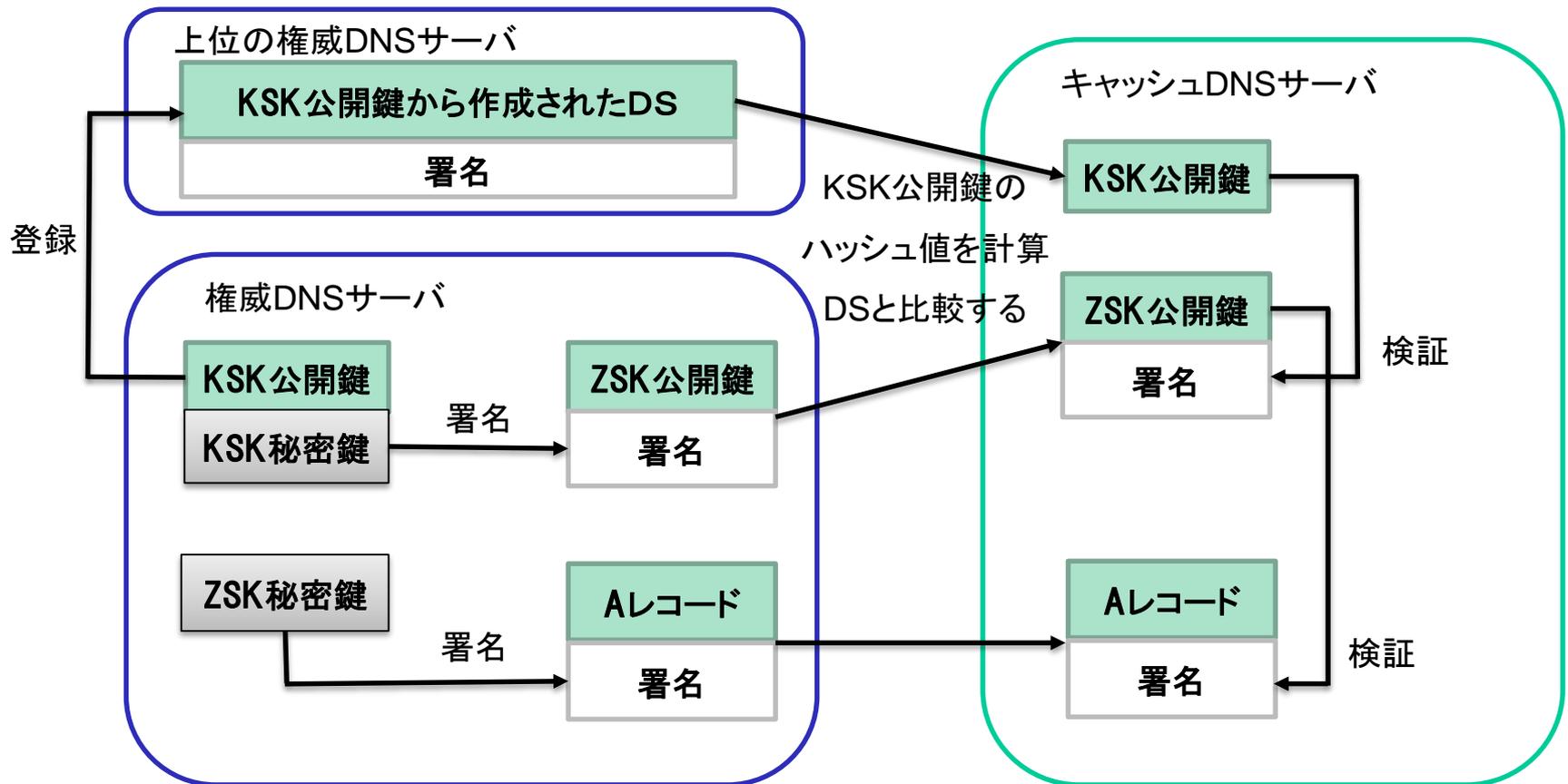
■ 完全性の保証

DNSの応答において、DNSレコードの改変や欠落が無いこと





- KSK (Key Signing Key) ゾーンの公開鍵に署名する鍵
- ZSK (Zone Signing Key) ゾーンに署名するための鍵
- DS (Delegation Signer) 上位の権威DNSサーバに登録する情報

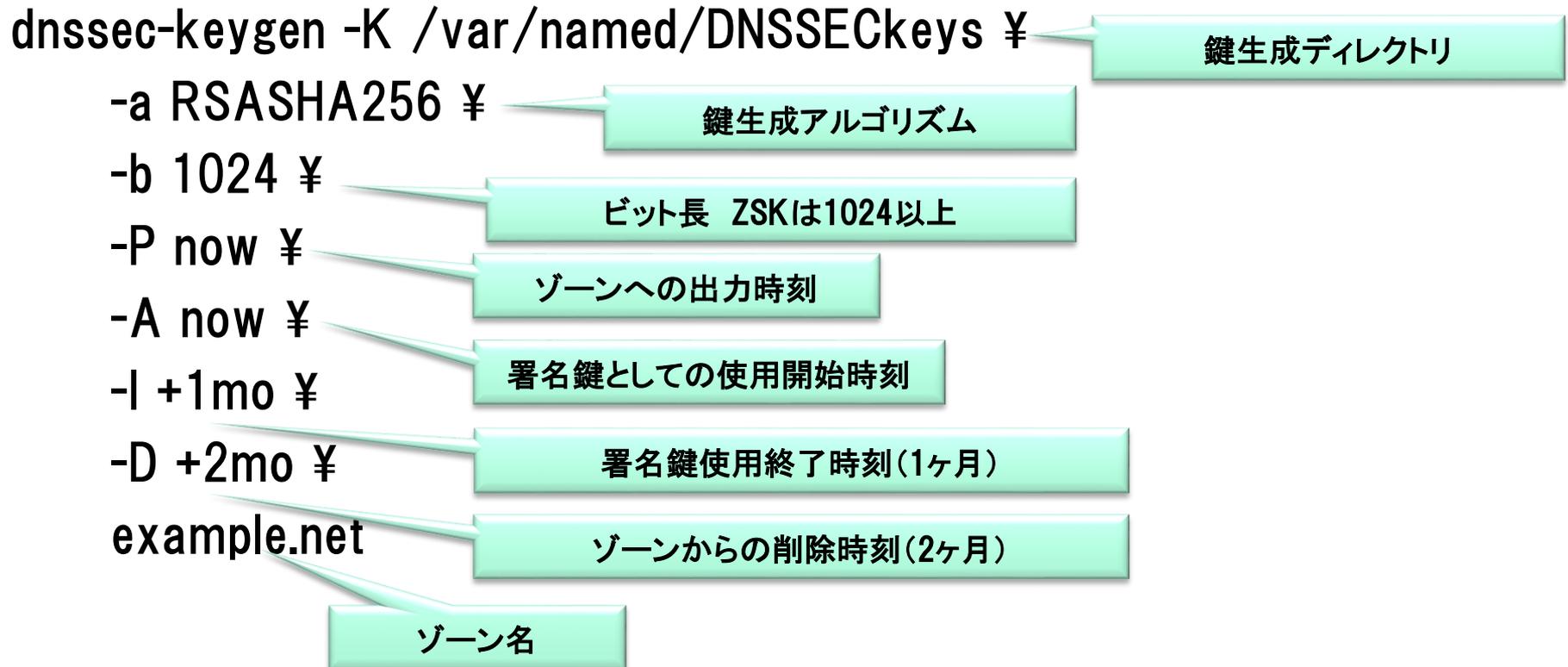




■ dnssec-keygenコマンド

- KSKとZSKの鍵を作成するコマンド

■ ZSKの鍵を作成する





■ KSKの鍵を作成する

```
dnssec-keygen -K /var/named/DNSSECkeys ¥
```

```
-a RSASHA256 ¥
```

```
-b 2048 ¥
```

```
-f KSK ¥
```

```
-P now ¥
```

```
-A now ¥
```

```
-l +13mo ¥
```

```
example.net
```

KSKのときは2048以上

KSKのときは指定する

-P -A は省略可能 デフォルトはnow

ZSK、KSKの公開鍵と秘密鍵ができるが、ファイル名だけだと区別がつかない。そのため、鍵のファイル内を確認する。

256 3 8

256:ZSK 257:KSK

3:プロトコルフィールド

8:アルゴリズム (RSASHA256)



■ dnssec-signzone

- ゾーンに対する署名を行うコマンド

`dnssec-signzone -S ¥` ———— スマート署名を利用する

`-K /var/named/DNSSECkeys ¥` ———— 鍵のあるディレクトリを指定する

`-d /var/named/DNSSECkeys ¥` ———— DSレコードファイルの格納場所

`-H 3 ¥` ———— ハッシュの繰り返し回数

`-3 'd0ec' ¥` ———— NSEC3の使用を指定し、ソルトを16進数で指定する

`-N unixtime ¥` ———— SOAのシリアル番号を指定する

`-o example.net ¥` ———— ゾーン名の指定

`/var/named/chroot/var/named/example.net.zone` ———— ゾーンファイル

example.net.zone.signedという署名ファイルが出来上がる
DSレコードをもつ、dsset-example.net.が出来上がる



レコード	説明
DNSKEY	KSKとZSKの公開鍵
DS	子ゾーンのKSKを親ゾーンで承認していることを示す
RRSIG	各DNSレコードへの署名を示す
NSEC	存在しないことを示すためのレコード
NSEC3	NSECレコードをたどるとゾーンデータを入手できてしまうので、ドメイン名をハッシュ関数でハッシュ化したもの。
NSEC3PARAM	権威DNSサーバ側が、NSEC3の生成を行うために必要なレコード

- ・DNSSECを使用したゾーンを公開する場合
自身のDSLレコードを上位の権威DNSサーバに登録、
公開してもらう必要がある。
(`dsset-example.net.`を使用する)



■権威サーバの設定

- named.confの設定

```
options {  
    dnssec-enable yes;  
}
```

- ゾーンファイルを変更する

```
zone "example.net" {  
    type master;  
    file "example.net.zone.signed";  
};
```

- namedプロセスに設定を読み込ませる
rndc reload



■キャッシュサーバの設定

- named.confの設定

```
options {  
    dnssec-enable yes;  
    dnssec-validation yes;  
}
```



■ 信頼の連鎖(トラストアンカー)とは

「信頼できる人が信頼できる人を紹介すると、その人も信頼できる」
ルートDNSからDSレコードを生成して設定する

■ ルートゾーンの公開鍵を入手

```
$ dig . DNSKEY | grep -w 257 > root-anchors.key
```

■ 公開鍵からDSレコードを生成する

```
$ dnssec-dsfromkey -a SHA-256 root-anchors.key
```

■ named.confに公開鍵を設定する

```
managed-keys {  
    "." initial-key 257 3 8  
    "AwEAAa・・・(省略)";  
};
```

ルートゾーンの公開鍵が正しいものであるかハッシュ値を計算して検証する必要があるが・・・
今回は割愛。



■ digコマンドでdnssecの設定を確認する

```
$ dig @127.0.0.1 centos.example.net +dnssec
```

- digコマンドのflagsに「ad」(Authentic Data)があるか確認する
- AnswerセクションにRRSIGレコードが追加されている

■ dnssecを無効にした問い合わせ

```
$ dig @127.0.0.1 centos.example.net +nodnssec
```

- digコマンドのflagsに「ad」(Authentic Data)がない
- AnswerセクションにRRSIGレコードがない
- 通常のDNSと同じ問い合わせ結果が表示されることを確認する



- 326.1 ホストの堅牢化 (重要度: 3)
- 326.2 ホストの侵入検知 (重要度: 4)
- 326.3 ユーザの管理と認証 (重要度: 5)
- 326.4 FreeIPA のインストレーションとSambaの統合 (重要度: 4)



■ FreeIPA (Identity, Policy, and Audit Server) の概要

以下のことが一元的に管理することができる。

- LinuxユーザーとLinuxグループ
- パスワードポリシー
- HBAC (Host-Based Access Control : ホストベースアクセスコントロール)
- sudo
- SELinuxユーザーとLinuxユーザーのマッピング
- DNS
- ssh公開鍵
- SSO (Single Sign On : シングルサインオン)

■ FreeIPAの構成

- LDAP: 389 Directoryサーバを使用
- 認証とシングルサインオン: MIT Kerberos
- PKI公開鍵基盤 (Dogtag Certificate System)
- DNSサーバ(bind)

WebによるGUIインターフェースも
提供されている



■前提

- 固定のIPアドレスを設定している
- ホスト名(FQDN)の名前解決、また逆引きができること
(/etc/hostsに設定するのも可)
- DNSの設定が自動で書き換わるので、DNSを設定している場合はバックアップを取っておく
- 443ポートを使用するので、mod_sslを使用していると設定できない

■インストール

```
# yum install ipa-server ipa-server-dns bind bind-dyndb-ldap
```

■セットアップ

```
# ipa-server-install --setup-dns
```



■DNSサーバの設定を上書きするかどうか

Existing BIND configuration detected, overwrite? [no]: yes

■ホスト名の確認

Server host name [centos.example.net]:

■ドメイン名の設定

Please confirm the domain name [example.net]:

■IPアドレスの設定

Please provide the IP address to be used for this host name: 192.168.56.150

■レルム名

Please provide a realm name [EXAMPLE.NET]:

■Directory Managerのパスワードを設定する

Directory Manager password:

■IPA adminのパスワードを設定する

IPA admin password:



■ DNS forwardersの設定をするかどうか

Do you want to configure DNS forwarders? [yes]:

■ DNS forwarder を設定する場合、forwarder の IP を指定

Enter IP address for a DNS forwarder: 8.8.8.8

■ 逆引きゾーンを設定するか

Do you want to configure the reverse zone? [yes]:

■ 逆引きゾーンを設定する場合のゾーン名

Please specify the reverse zone name [56.168.192.in-addr.arpa.]:

■ 設定を確認

Continue to configure the system with these values? [no]: yes



- ユーザを追加したときのデフォルトシェルをbashに変更

```
# ipa config-mod --defaultshell=/bin/bash
```

- ユーザの追加

```
# ipa user-add ipauser --first=ipa --last=user --password
```

- ユーザの確認

```
# ipa user-find ipauser
```

- グループの追加

```
# ipa group-add ipagroup --desc "example.net group"
```

- グループにユーザを追加

```
# ipa group-add-member ipagroup --users=ipauser
```



■ IPAクライアントのインストール

```
# yum -y install ipa-client
```

■ /etc/resolv.confでnameserverをIPAサーバにしておく

■ IPAクライアントの設定

```
# ipa-client-install
```

■ 質問される項目

- 設定を確認して yes で進む

```
Continue to configure the system with these values? [no]: yes
```

- 「admin」で入力

```
User authorized to enroll computers: admin
```

■ ログイン時にホームディレクトリを自動作成

```
# authconfig --enablemkhomedir --update
```



CentOS Linux 7 (Core)

Kernel 3.10.0-123.20.1.el7.x86_64 on an x86_64

centos login: ipauser

Password:

Password expired. Change your password now.

Current Password:

New password:

Retype new password:

Creating home directory for ipauser.

初回はパスワードの
変更を求められる



サブコマンド	説明
user-add	ユーザの追加 <code>ipa user-add ipauser --first ipa --last user</code>
user-del	ユーザの削除 <code>ipa user-del ipauser</code>
user-find	ユーザの検索 <code>ipa user-find ipauser</code>
group-add	グループの追加 <code>ipa group-add ipagroup --desc "this is ipa group"</code>
group-add-member	グループにメンバーを追加する <code>ipa group-add-member ipagroup --users=admin,ipauser</code>
group-del	グループの削除 <code>ipa group-del ipagroup</code>
group-find	グループの検索 <code>ipa group-find ipagroup</code>



- 327.1 任意アクセス制御 (重要度: 3)
- 327.2 強制アクセス制御 (重要度: 4)
- 327.3 ネットワークファイルシステム (重要度: 3)



- Linuxのカーネルに強制アクセス制御 機能を付加する
 - 強制アクセス制御 MAC (Mandatory Access Control)
- SELinuxを使用しないLinuxのアクセス権は・・・
 - ファイルやディレクトリのパーミッションに基づいて行われる
 - rootはこのパーミッションを無視してアクセスが可能
 - root権限が乗っ取られると、致命的な被害を受ける
 - ファイルによるパーミッションの設定は任意アクセス制御と呼ばれる
任意アクセス制御 (DAC: Discretionary Access Control)
- SELinuxでは以下のようなことが可能
 - HTTP、FTPといったプロセスごとにアクセス制限をかけるType Enforcement (TE)
 - rootも含む全てのユーザに関して制限をかけるロールベースアクセス制御 (RBAC)



■TE (Type Enforcement)

- 全てのプロセスに対して「ドメイン」と呼ばれるラベルを付加する。
- リソース(ファイルやディレクトリ)に対しても同じく「タイプ」と呼ばれるラベルを付与する。
- 各リソースには「アクセス・ベクタ」が割り当てられる。
- アクセス・ベクタとは「読み込み」、「書き込み」といったリソースに対して行える操作の種類
- 各ドメインとタイプに対して許可されるアクセス・ベクタを、セキュリティーポリシーとして設定可能

■RBAC (Role-based access control)

- 「ロール」と呼ばれるいくつかのドメインを束ねたものを設定し、それをユーザに付与する仕組み

例

- sysadm_r システム管理者ロール
- webmaster_r Webサーバのコンテンツ管理者のためのロール



- SELinuxを有効にすると、リソースやプロセスにコンテキストが付与される
- コンテキストには、以下の識別子がある
 - ユーザ識別子
 - ロール識別子
 - タイプ識別子
 - MLS (Multi Level Security)

■ リソース (ファイル) のコンテキスト

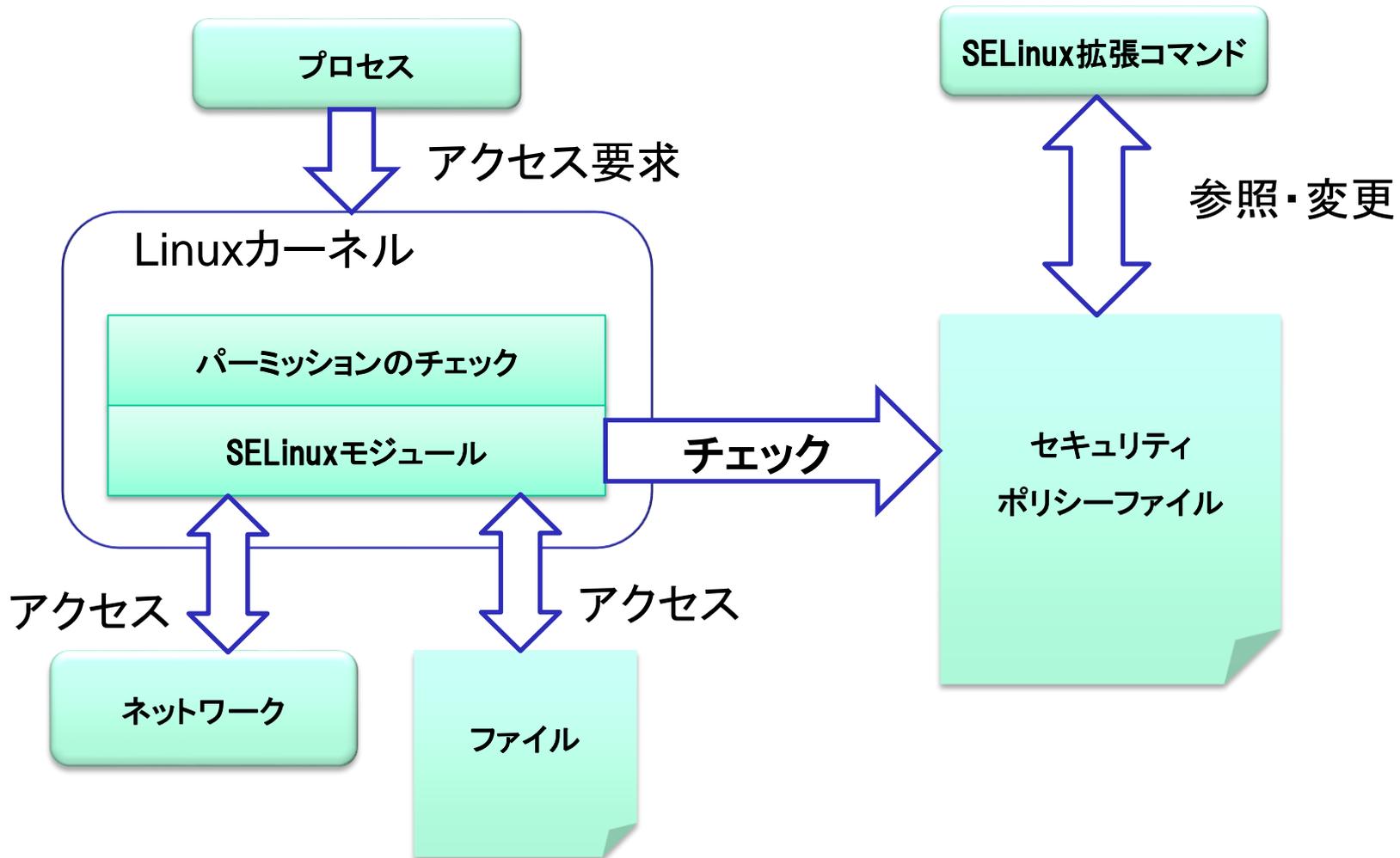
`-rw-rw-r--. centuser centuser unconfined_u:object_r:user_home_t:s0 testfile`

ユーザ識別子

ロール識別子

タイプ識別子

MLS





■ ドメイン遷移とは

- 通常は子プロセスは親プロセスと同じドメインで動作する
- 設定により親プロセスとは違うドメインで子プロセスを実行すること

■ httpdプロセスの実行



■ ドメイン遷移の役割

- プロセスに権限(ドメイン)を割り当てることができる
- 不要な権限の昇格が避けられる

SUIDによる一般ユーザからrootへの昇格など



■SELinuxが有効か確認する

- getenforce
- ステート

「Enforcing」 → SELinuxが有効になっている(強制モード)

「Permissive」 → SELinuxは有効になっている(許容モード)

「Disabled」 → SELinuxは無効になっている

■SELinuxを設定する

- コマンドで設定 `setenforce`
- "setenforce 0" → Permissiveモードで動作する
- "setenforce 1" → Enforcingモードで動作する
- 設定ファイルで設定 `/etc/selinux/config`

`SELINUX=enforcing`

`SELINUX=permissive`

`SELINUX=disable`



■コンテキストを確認する

- ファイルのコンテキストを確認する

```
$ ls -lZ
```

- プロセスのコンテキストを確認する

```
$ ps axZ
```

- ユーザのコンテキストを確認する

```
$ id -Z
```



- アクセス制御を行うルールはポリシーによって決められている
- CentOSのデフォルトポリシーは「targeted」が設定されている
 - targeted : システム上で攻撃対象となる可能性の高いプロセスに対してアクセス制御を適用するポリシー
 - minimum : ポリシーファイル構成は Targeted と全く同じだが、アクセス制御が適用されるプロセスは最小限に絞られている。
 - mls : マルチレベルセキュリティ
- ポリシーは自作することもできるが、ルールが非常に複雑である
- targetedポリシーをもとに、カスタマイズする方法が容易
- ポリシーの変更は/etc/selinux/configで行う



■ semangeは以下のことができる

- SELinuxの有効／無効
- リソースに対するセキュリティコンテキストの変更
- ユーザに対するセキュリティコンテキストの割当て
- ネットワークに対するセキュリティコンテキストの割当て
- booleanの設定

semanageで制御できる項目はmanのマニュアルなどで調べてください。



- SELinuxのポリシーを変更するのは複雑である
- ポリシーは変更せずに、ある特定の機能だけを有効にしたり無効にする機能がある
- 現在のboolean値を確認する

```
# semanage boolean -l もしくは # getsebool -a
```

- boolean値を変更する

```
# setsebool -P allow_ftp_full_access on
```

(再起動後も設定値を維持する場合は-Pオプションを使用する)

- 設定値が変更されたか確認する

```
# getsebool allow_ftp_full_access
```



- 328.1 ネットワークの堅牢化 (重要度: 4)
- 328.2 ネットワークの侵入検知 (重要度: 4)
- 328.3 パケットフィルタ (重要度: 5)
- 328.4 仮想プライベートネットワーク (VPN) (重要度: 4)



■ 侵入検知ソフトウェア

■ Snortの特徴

- IPネットワーク上でのリアルタイムの解析
- GPLライセンス
- パケットスニファ／パケットロガーとしても使用できる
- 豊富なプリプロセッサが用意されている
 - portscan: ポートスキャンの検出を行う
 - frag2: IPフラグメントの再構築を行う
 - stream4: TCPストリームの再構築とステートフルな解析を行う
 - telnet_decode: Telnetの制御文字を正規化する
- さまざまな形式でアラートを出力することができる
- 公式サイト: <https://www.snort.org/>



■ ソースからインストールを行う

- ソースからのインストールは複雑なため、Linuxセキュリティ標準教科書を参照してください

■ ソースを展開したディレクトリから設定ファイルをコピーする

```
# cp snort-2.9.7.3/rpm/snort.sysconfig /etc/sysconfig/snort
```

■ /etc/sysconfig/snortを編集する

```
INTERFACE=eth1
```

```
USER=snort
```

```
GROUP=snort
```

```
LOGDIR=/var/log/snort
```

Listenするネットワークインターフェースを指定する



■ 起動用スクリプトをコピーする

```
# cp /home/centuser/work/snort-2.9.7.3/rpm/snortd /etc/init.d/  
# chmod 755 /etc/init.d/snortd
```

■ コミュニティ版ルールの配置

```
# mkdir -p /etc/snort/rules  
# chown -R snort.snort /etc/snort  
# wget https://www.snort.org/rules/community.tar.gz  
# tar -xvfz community.tar.gz -C /etc/snort/rules
```

■ ライブラリが利用するディレクトリの作成

```
# mkdir /usr/local/lib/snort_dynamicrules
```

■ ログ領域の作成

```
# mkdir /var/log/snort ; chown -R snort.snort /var/log/snort
```



- 設定ファイル `/etc/snort/snort.conf`

- ネットワークの設定を行う

```
ipvar HOME_NET 192.168.56.0/24
```

```
ipvar EXTERNAL_NET any
```

- 以下の変数のパスをカレントディレクトリからのパスとする(../を./に変更する)

```
var RULE_PATH ./rules
```

```
var SO_RULE_PATH ./so_rules
```

```
var PREPROC_RULE_PATH ./preproc_rules
```

```
var WHITE_LIST_PATH ./rules
```

```
var BLACK_LIST_PATH ./rules
```



■読み込むルールの設定を記述

`include $RULE_PATH/local.rules` → 追加する

`include $RULE_PATH/community.rules` → 追加する

`#これ以下ルールはすべてコメントにする`

`#include $RULE_PATH/app-detect.rules`

`#include $RULE_PATH/attack-responses.rules`

...

■独自ルールを作成する

`# vi /etc/snort/rules/local.rules`

`alert icmp any any -> any any (msg: "ICMP Packet detected"; sid:999999;)`



■ プロミスクラスモードの設定

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth1
```

```
PROMISC=yes
```

ファイルの最後に記述する

- (VirtulaBoxを使用している場合は、ネットワークの設定でプロミスクラスモードを許可にする)

■ Snortを起動する

```
/etc/init.d/snortd start
```

■ Snortが起動しない場合は・・・

- /var/log/messagesにエラーメッセージが出力されていないか確認

■ 動作テスト

- 他のホストからpingを実行してみる
- /var/log/snort/alertにログが出力されていればOK



■ シグネチャのルールは、ルールヘッダとルールボディから成る

■ 書式

<ルールアクション> <プロトコル>	}	ルールヘッダ
<IPアドレス> <ポート番号> <方向演算子>		
<IPアドレス> <ポート番号>	}	ルールボディ
<(オプション...)>		

ルールアクション	説明
activate	ルールに該当するパケットが存在する場合、警告を出す (dynamicアクションを呼び出す)
alert	ルールに該当するパケットを記録し、警告を出す
dynamic	activateアクションから呼び出され、該当するパケットを記録する
log	ルールに該当するパケットを記録する
pass	ルールに該当するパケットを無視する



例1

```
alert tcp any any -> any 80 (msg: "http request GET" ; content:"GET"; http_method; sid:1000000)
```

例2

```
alert tcp any any -> any 80 (msg: "http request URI" ; content:"/index.html"; http_uri; sid:1000001)
```

msg: → ログに出力するメッセージ

content: → パケットのペイロード部にマッチする文字列を指定する

httpd_method → HTTPリクエストのメソッドでマッチするもの

http_uri → HTTPリクエストのURIでマッチするもの

sid: → シグネチャのIDを指定する。独自ルールは1, 000, 000以上



徹底攻略LPIC Level3 303教科書+問題集 [Version 2.0]対応 徹底攻略シリーズ

常泉 茂雄 (著), 菖蒲 淳司 (著)

出版社: 翔泳社

350ページ

価格3,672円

ISBN-10: 4295000515

ISBN-13: 978-4295000518



Linuxセキュリティ標準教科書 (Ver1.0.0)

詳しくは下記URLで

<http://www.lpi.or.jp/linuxtext/security.shtml>

発行: エルピーアイジャパン



実践DNS DNSSEC時代のDNSの設定と運用

民田 雅人 (著), 森下 泰宏 (著), 坂口 智哉 (著), 株式会社日本レジストリサービス (JPRS) (監修)

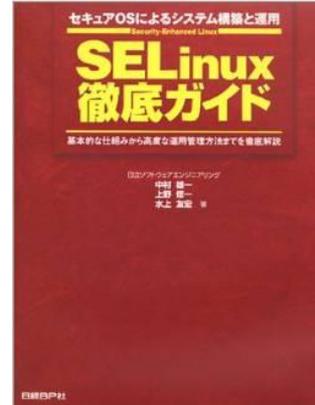
出版社: KADOKAWA / アスキー・メディアワークス (2014/2/20)

328ページ

価格3,024円

ISBN-10: 4048700731

ISBN-13: 978-4048700733



SELinux徹底ガイド—セキュアOSによるシステム構築と運用 基本的な仕組みから高度な運用管理方法までを徹底解説

中村 雄一 (著), 水上 友宏 (著), 上野 修一 (著), & 3 その他

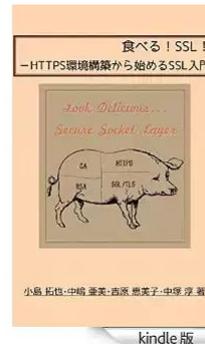
出版社: 日経BP社

318 ページ

価格4913円

ISBN-10: 4822221113

ISBN-13: 978-4822221119



食べる！SSL！—HTTPS環境構築から始めるSSL入門 [Kindle版]

小島 拓也 (著), 中嶋 亜美 (著), 吉原 恵美子 (著), 中塚 淳 (著)

119 ページ

価格320円



質疑応答についてはお気軽にお声掛けください。

ご清聴ありがとうございました。



Skill Brain スキルブレイン株式会社

<http://www.skillbrain.co.jp>



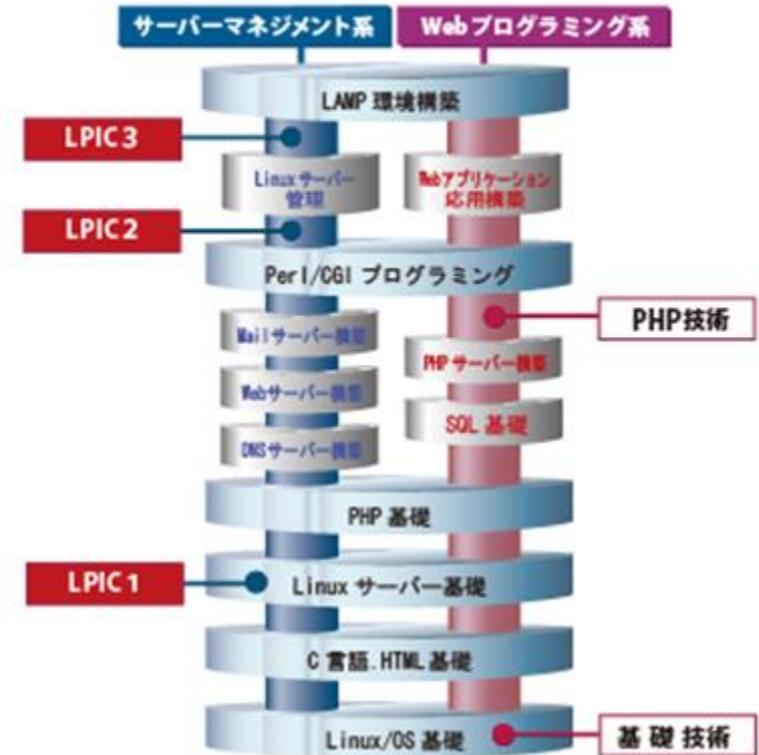
info@skillbrain.co.jp



■OSS/Linux エンジニア研修

- Linux基礎
 - Linuxサーバー構築実践
 - Linuxサーバーセキュリティ構築実践
 - Linuxシステム運用・トラブルシューティング
 - CentOS7 アップデート
 - LPIC(レベル1・2・3)試験対策
 - OSS-DB試験対策
 - HTML5試験対策

 - ITIL®ファウンデーション試験対策
 - ITIL®運用／管理
 - ITIL®エキスパート／インターミディエイト
 - PRINCE2®
 - PMP® PMBOK®
 - セキュリティ研修
- その他企業様ごとにセミオーダー研修を承ります。





三浦 一志

サーバ管理者として8年以上の実務経験を積み、講師としても15年以上のキャリアを持つ。法人向けにLPIC研修・Linuxサーバ構築・セキュリティ研修やITIL研修を主として担当。ITIL認定講師 情報セキュリティスペシャリスト

【担当講習】

・Linux/UNIX ・LPIC試験対策 ・セキュリティ ・Java ・PHP ・OSS-DB ・HTML5



河原木 忠司

Linux、Windowsを使ったインフラ環境の構築・運用、セキュアなインターネットサーバーの構築など、企業・官公庁向けの技術研修を担当。

【担当講習】

・Linux ・Windows ・LPIC試験対策 ・セキュリティ ・仮想化



木村 祐

大手メーカーにて生産管理、ベンチャー支援企業にてビル型ISP事業、ベンチャー系システムマネジメント企業にてシステム監視、運用事業に従事、その後大手電機メーカー情報会社にてデータセンターマネージャ ITIL推進プロジェクトを推進。年間30講座以上を担当し300名以上の合格者を輩出。

【担当講習】

・ITIL®ファウンデーション ・ITIL®エキスパート ・ITIL®Lプラクティショナー ・ITIL運用/管理



谷戸 信幸

大手航空会社のIT部門にて、オンラインリアルタイムで運用される旅客予約システムのシステム基盤のSEとしてシステム維持管理、可用性、キャパシティ管理などを担当。その後データ通信ネットワークの開発、展開、維持管理に従事。IT系子会社を経て現職。

【担当講習】

・PMP® ・PMBOK® ・PRINCE2® ・ITIL®エキスパート