株式会社ケイ・シー・シー

Linux Professional Institute Japan LPI-JAPAN

LPICレベル3　303技術解説無料セミナー

2014/7/5

株式会社ケイ・シー・シー
西日本センターユニット　ITラーニングセンター
村田　一雄

講師プロフィール

株式会社ケイ・シー・シー

■会社概要

株式会社ケイ・シー・シー

http://www.kcc.co.jp/

■講師紹介

ソリューションセンターユニット　ITラーニングセンター所属

Linuxをメインにネットワーク・セキュリティ・XML・資格取得講座など
様々な技術研修を担当

Linux Professional Institute

アウトライン

1. LPIC レベル3　試験概要
   - LPIC試験概要

2. 技術解説項目
   - 主題321　アクセス制御
   - 主題324　ネットワークセキュリティ

LPICレベル3　試験概要

LPI 303 Security Examの出題範囲

- 主題320: 暗号化
- 主題321: アクセス制御
- 主題322: アプリケーションセキュリティ
- 主題323: 操作のセキュリティ
- 主題324: ネットワークセキュリティ

本日のサマリ

ホストセキュリティ

- PAM（Pluggable Authentication Module）
- ACL（Access Control List）

ネットワークセキュリティ

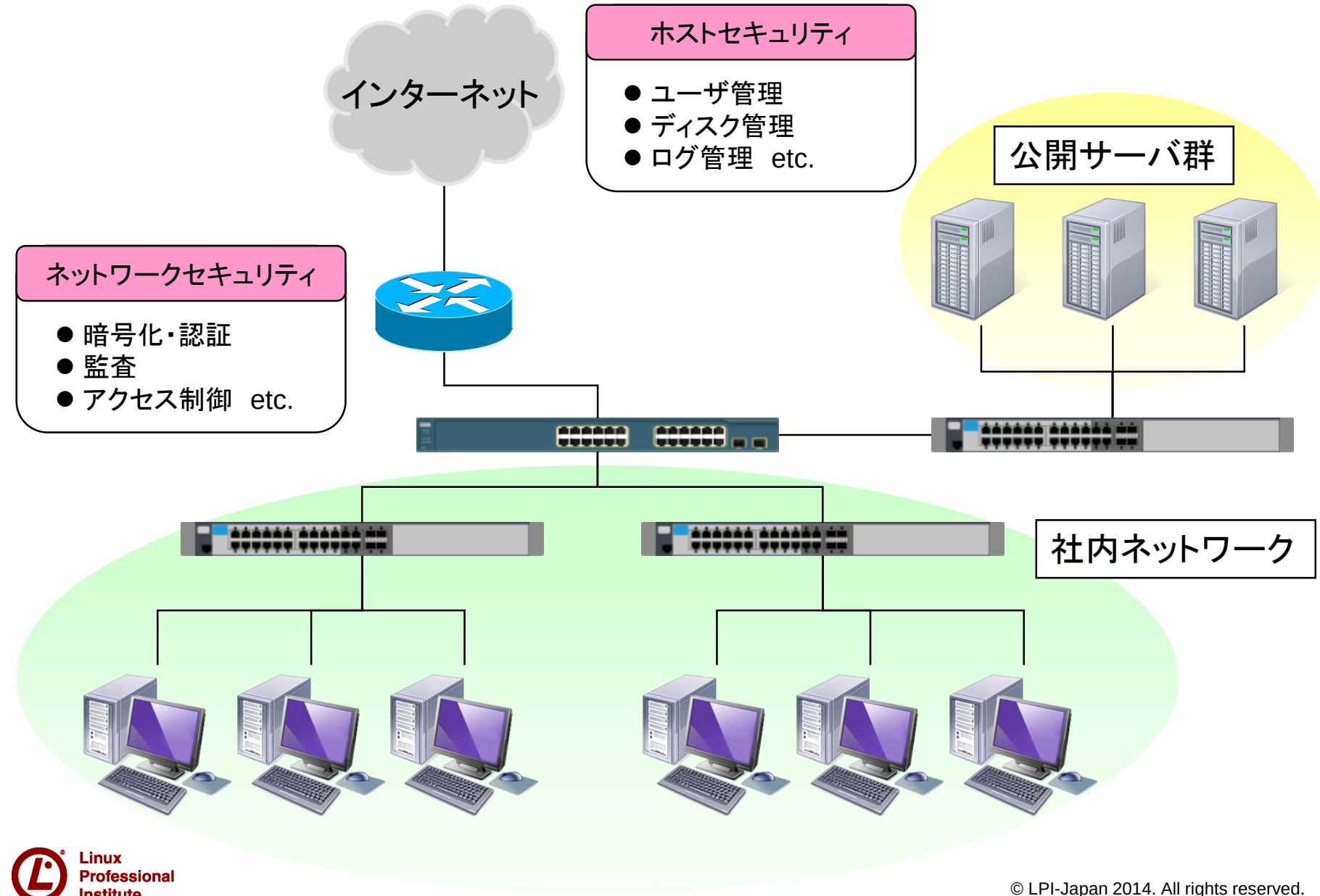
- Snort（ネットワーク型IDS）
- Tripwire（ホスト型IDS）
- Nessus（セキュリティスキャナ）
- Nmap（ポートスキャン）
- Wireshark, tshark, tcpdump（パケットキャプチャ）

株式会社ケイ・シー・シー

セキュリティ概要

Linux Professional Institute

ホストセキュリティとネットワークセキュリティ



株式会社ケイ・シー・シー

LPICレベル3 303　技術解説

主題321　アクセス制御

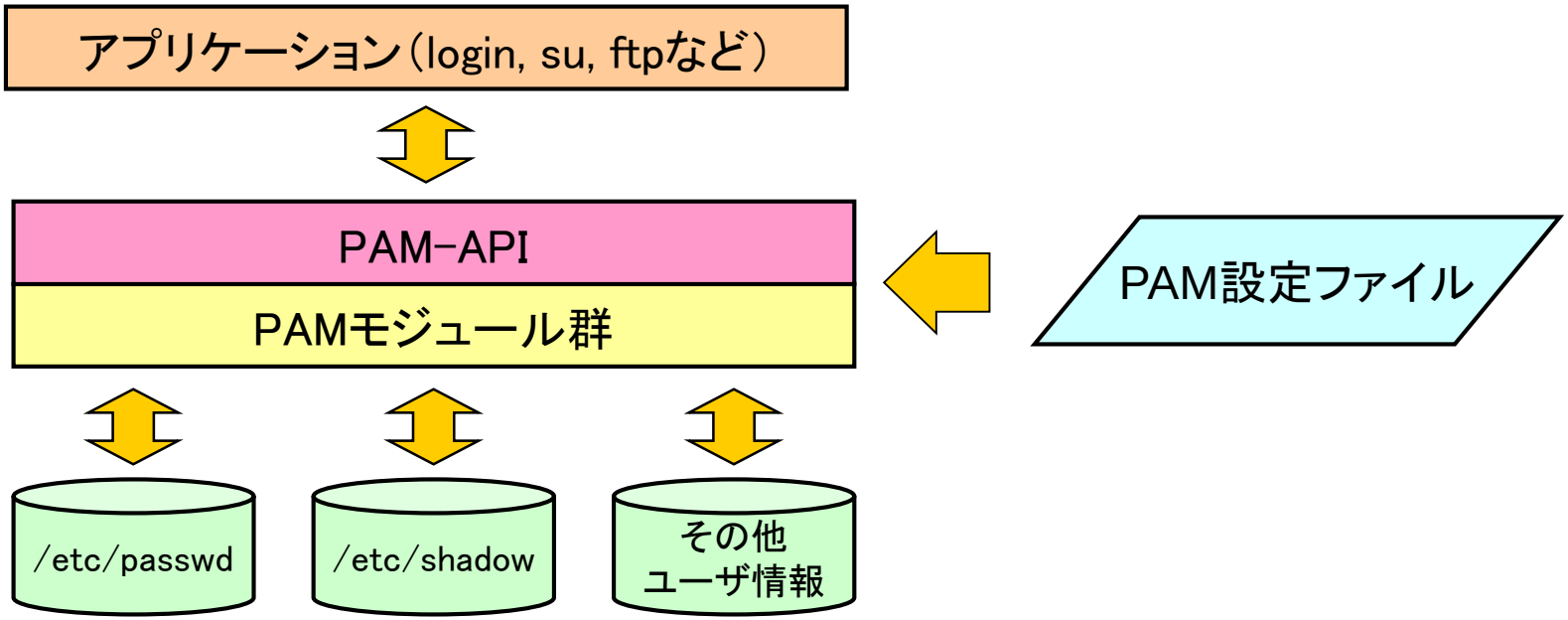
321.1 ホストベースのアクセス制御	重要度2
321.2 拡張属性とACL	重要度5

Linux Professional Institute

PAMによるアクセス制御

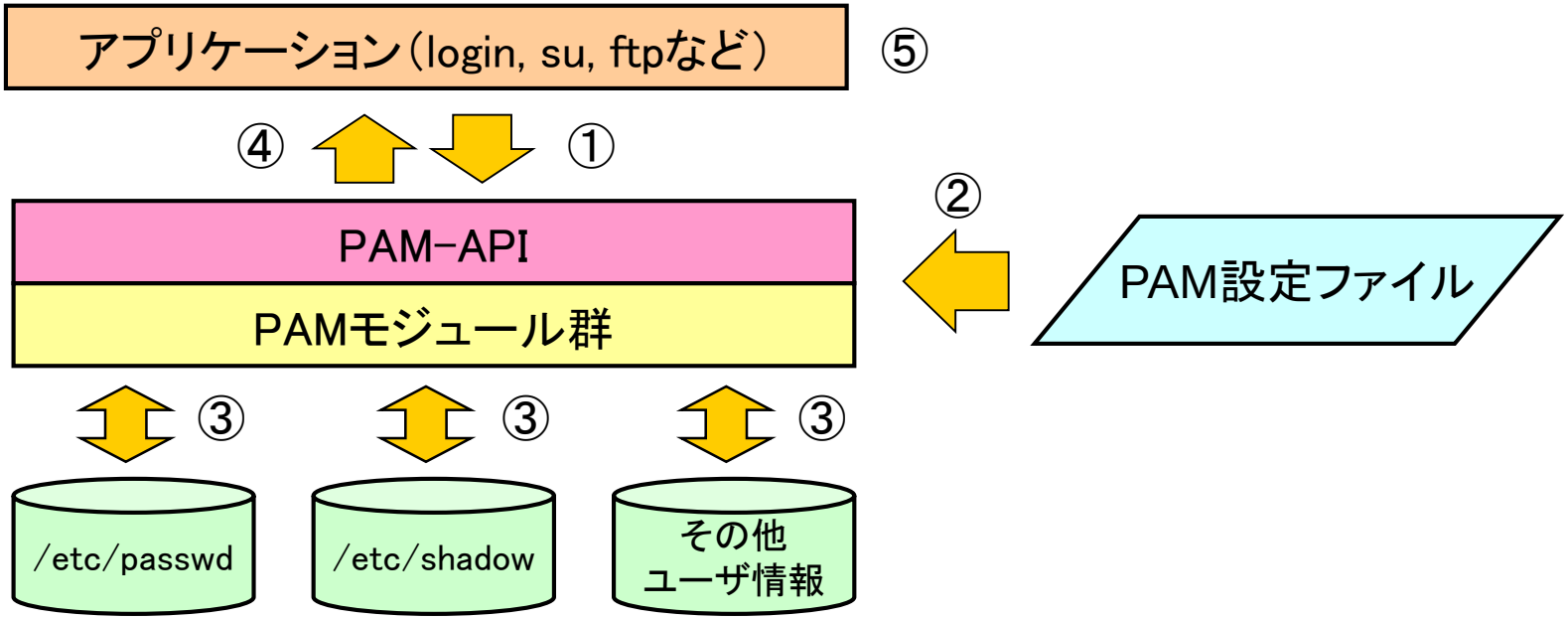
PAM（Pluggable Authentication Module）とは

- 各アプリケーションに対して認証機能を提供する仕組み
- 認証機能はモジュールで提供される
- 各アプリケーションは認証部分の実装を必要としない
- 認証が必要な場合はモジュールを呼び出して実行する



PAM認証の仕組み

- PAMによる認証手順
  - ① アプリケーション認証が行われるタイミングでPAMを呼び出す
  - ② 各アプリケーション用のPAM設定ファイルを読み込む
  - ③ 設定ファイルに記述されたPAMモジュールが呼び出され実行される
  - ④ 認証の結果をアプリケーションに返す
  - ⑤ 認証結果に応じてサービスが提供される



PAMを利用するアプリケーション

- PAMを利用する主なアプリケーション
  - login （CUIログインを提供）
  - su （ユーザの切り替え）
  - ftp （ファイル転送）

- lddコマンド
  - アプリケーションごとに必要とされる共有ライブラリを表示する
  - PAMを使用するライブラリには通常「pam」が含まれる

```
# ldd /bin/login
      linux-gate.so.1 => (0x00d5a000)
      libpam.so.0 => /lib/libpam.so.0 (0x00cae000)
      libpam_misc.so.0 => /lib/libpam_misc.so.0 (0x00c96000)
      libcrypt.so.1 => /lib/libcrypt.so.1 (0x00d23000)
      libdl.so.2 => /lib/libdl.so.2 (0x00110000)
      libc.so.6 => /lib/libc.so.6 (0x00301000)
      libaudit.so.0 => /lib/libaudit.so.0 (0x00115000)
      /lib/ld-linux.so.2 (0x00ce4000)
```

PAM設定ファイル

- /etc/pam.dディレクトリ
  - PAMを利用するアプリケーションごとの認証設定が格納されるディレクトリ
  - 対応するアプリケーションがない場合、/etc/pam.d/otherの設定が読み込まれ、全ての認証処理に失敗を返す

```
# ls /etc/pam.d
atd                 kshell              su
authconfig-gtk      neat                su-l
authconfig-tui      newrole             sudo
chfn                other               sudo-i
chsh                passwd              system-auth
config-util         pirut               system-auth-ac
cpufreq-selector    pm-hibernate        system-cdinstall-helper
crond               pm-powersave        system-config-authentication
cups                pm-suspand          system-config-date

...(以下略)
```

PAMモジュール

- PAMモジュールの格納場所
  - PAMモジュールは/lib/securityディレクトリに格納される
  - PAMモジュールは通常「pam_xxx.so」という名前が付けられる

```
# ls /lib/security
pam_access.so                pam_limits.so                pam_smbpass.so
pam_ccreds.so                pam_listfile.so              pam_stack.so
pam_chroot.so                pam_localuser.so             pam_stress.so
pam_console.so               pam_loginuid.so              pam_succeed_if.so
pam_cracklib.so              pam_mail.so                  pam_tally.so
pam_debug.so                 pam_mkhomedir.so             pam_tally2.so
pam_deny.so                  pam_motd.so                  pam_time.so
pam_echo.so                  pam_namespace.so             pam_timestamp.so
pam_ecryptfs.so              pam_nologin.so               pam_tty_audit.so
pam_env.so                   pam_oddjob_mkhomedir.so      pam_umask.so
pam_exec.so                  pam_passwdqc.so              pam_unix.so
pam_filter/                  pam_permit.so                pam_unix_acct.so
pam_filter.so                pam_permit.so                pam_unix_auth.so
pam_ftp.so                   pam_postgresok.so            pam_unix_passwd.so
pam_group.so                 pam_rhosts.so                pam_unix_session.so
pam_issue.so                 pam_rhosts_auth.so           pam_userdb.so

...（以下略）
```

（参考）主なPAMモジュール

pam_console.so	root権限の必要なコマンドの一部を一般ユーザに利用可能にする
pam_cracklib.so	パスワード作成時のチェックを行う（辞書チェック、文字数など）
pam_deny.so	全てのアクセスを拒否する
pam_env.so	ユーザログイン時に環境変数を設定する
pam_keyinit.so	呼び出したプロセスがデフォルトのセッションキーリング以外のキーリングを持っていることを保証する
pam_limits.so	ユーザログイン時にリソース制限を可能にする
pam_loginuid.so	認証されたプロセスにloginuidプロセスの属性を設定する
pam_nologin.so	/etc/nologinファイルがある場合、一般ユーザのログインを拒否する
pam_permit.so	全てのアクセスを許可する
pam_rootok.so	rootユーザであればアクセスを許可する
pam_securetty.so	/etc/securettyに記述された端末以外からのrootログインを拒否する
pam_selinux.so	デフォルトのセキュリティコンテキストを設定する
pam_succeed_if.so	アカウントの文字列をテストする
pam_unix.so	標準的なUNIX認証を行う
pam_wheel.so	suコマンドでrootに変更できる一般ユーザを制限する
pam_xauth.so	cookieなどの情報をユーザ間で転送する

PAM設定ファイルの記述形式

PAMの設定フォーマット

【書式】 タイプ　コントロール　モジュールパス　モジュール引数

タイプ	PAM認証の種別をグループ化して指定する
コントロール	認証に失敗した際の処理方法を指定する
モジュールパス	認証に使用するモジュールとその格納場所を指定する
モジュール引数	モジュールに与える引数を指定する

```
auth        sufficient  pam_rootok.so
auth        include     system-auth
account     sufficient  pam_succeed_if.so  uid = 0 user_uid quiet
account     include     system-auth
password    include     system-auth
session     include     system-auth
session     optional    pam_xauth.so
```

タイプ　コントロール　モジュールパス　モジュール引数

PAM設定ファイルの書式(1)

- タイプ
  - 4種類のタイプが指定されている
  - 1つのタイプに複数の認証処理をスタック記述することができるため、複数の異なる認証結果を積み重ねた上で、最終的に認証可否の結果を通知することができる

auth	ユーザが本人であることを証明する パスワードや他の識別手段を用いて確認する
account	非認証ベースのアカウント管理を行う アクセス時刻・最大ユーザ数・場所・アカウントの有効期限などに 基づいてアクセス制限/許可を判断する
password	ユーザに関連付けられたパスワードを更新する
session	サービスの開始前・終了時に処理を行う ユーザとのデータ交換の開始・終了に関するログ採取や ディレクトリのマウントなどが含まれる

PAM設定ファイルの書式(2)

- コントロール
  - 記述方法には以下の2種類の方法がある
    - キーワードによる記述方法
    - 「リターンコード」と「アクション」のペアによる記述方法([ ]で括る)
- キーワードによる記述方法

required	認証に失敗した時に認証を拒否する ただし、同じタイプ中に別のモジュールチェックが残っている場合は、すべてのチェックを実施した後に認証結果を返す
requisite	認証に失敗した時に即座に認証を拒否する
sufficient	認証に必要な条件が十分満たされているかを判定する ● sufficient行の認証が失敗した場合は無視して、すべてのrequiredおよびrequisite行の認証が成功した場合に成功を返す ● sufficient行の認証が成功した場合、それ以前にあるrequiredおよびrequisite行の認証が成功していれば、成功を返す ● 上記以外の場合は失敗を返す
optional	認証の成否に関係のない処理を記述する
include	別のPAM設定ファイルにある処理を実施する

PAM設定ファイルの書式(3)

- 「リターンコード」と「アクション」のペアによる主な記述方法
  - [リターンコード=アクション]の形式で記述する
  - 複数指定する場合、空白で区切って記述する
- 主なリターンコード(PAMモジュールの戻り値)

user_unknown	ユーザが不明
new_authtok_reqd	新しい認証情報が必要
success	成功
ignore	認証モジュールを無視
default	リターンコードがない場合の暗黙のリターンコード

- 主なアクション(処理内容)

ok	この時点でのスタック全体の処理ステータスを成功とする
ignore	このモジュールの実行結果を無視する
done	モジュールが成功が返すと即座にスタックから抜ける
bad	スタック全体の処理ステータスを失敗とする
die	スタック全体を失敗とみなし、即座にアプリケーションへ結果を返す

PAM設定ファイルの書式(4)

- (参考)キーワードの「リターンコード」と「アクション」による記述

required	[success=ok new_authtok_reqd=ok ignore=ignore default=bad]
requisite	[success=ok new_authtok_reqd=ok ignore=ignore default=die]
sufficient	[success=done new_authtok_reqd=done default=ignore]
optional	[success=ok new_authtok_reqd=ok default=ignore]

PAMによる認証設定(1)

loginプログラムの設定例

```
auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so
auth       include    system-auth

account    required   pam_nologin.so
account    include    system-auth

password   include    system-auth

# pam_selinux.so close should be the first session rule
session    required   pam_selinux.so close
session    include    system-auth
session    required   pam_loginuid.so
session    optional   pam_console.so
# pam_selinux.so open should only be followed by sessions to be executed in the
user context
session    required   pam_selinux.so open
session    optional   pam_keyinit.so force revoke
```

system-authファイルを参照

PAMによる認証設定(2)

suプログラムの設定例

```
auth       sufficient    pam_rootok.so
# Uncomment the following line to implicitly trust users in the "wheel" group.
#auth       sufficient    pam_wheel.so trust use_uid
# Uncomment the following line to require a user to be in the "wheel" group.
#auth       sufficient    pam_wheel.so use_uid
auth       include       system-auth

account    sufficient    pam_succeed_if.so uid = 0 user_uid quiet
account    include       system-auth

password   include       system-auth

session    include       system-auth
session    optional      pam_xauth.so
```

rootであればアクセスを許可

system-authファイルを参照

PAMによる認証設定(2)

system-authの設定例

環境変数を設定

ベーシック認証を実施
(Nullパスワード許可・ID/PW入力)

全て拒否を返す

入力された文字列を評価し、
uidが500以上であれば成功を返す

```
auth            required        pam_env.so
auth            sufficient      pam_unix.so nullok try_first_pass
auth            requisite       pam_succeed_if.so uid >= 500 quiet
auth            required        pam_deny.so

account         required        pam_unix.so
account         sufficient      pam_succeed_if.so uid < 500 quiet
account         required        pam_permit.so

password        requisite       pam_cracklib.so try_first_pass retry=3
password        sufficient      pam_unix.so md5 shadow nullok try_first_pass
use_authtok
password        required        pam_deny.so

session         optional        pam_keyinit.so revoke
session         required        pam_limits.so
session         [success=1 default=ignore] pam_succeed_if.so service in crond
quiet use_uid
session         required        pam_unix.so
```

ACLによるアクセス制御

ACL（Access Control List）とは

- ファイルやディレクトリに対して、標準的なアクセス権に加えて、柔軟なアクセス制御を可能とする仕組み
- カーネル2.6より標準実装
- ext2, ext3, ReiserFS, JFS, XFSなどのファイルシステムでサポート
- ファイルシステム単位でアクセス制御が可能

制御対象	アクセス権
所有ユーザ	読み込み（r）
所有グループ	書き込み（w）
その他	実行（x）

ACLの実装

制御対象	アクセス権
所有ユーザ	読み込み（r）
所有グループ	書き込み（w）
その他	実行（x）
特定ユーザ	
特定グループ	
ACLマスク	

ACLによるアクセス制御

- ACLの設定方法
  - ① マウントオプションの指定
  - ② ACLの設定
- マウントオプションの設定
  - /etc/fstabのマウントオプションに「acl」を追加

```
LABEL=/          /                ext3      defaults          1 1
LABEL=/home      /home            ext3      defaults,acl      1 2
LABEL=/var       /var             ext3      defaults          1 2
...(以下略)
```

  - ファイルシステムの再マウント

```
# mount -o remount /home
# mount
/dev/sda6 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda7 on /home type ext3 (rw,acl)
/dev/sda3 on /var type ext3 (rw)
...(以下略)
```

ACLによるアクセス制御

- ACLの設定
  - アクセスACL
    - 制御対象に付与される特定のアクセス権
  - デフォルトACL
    - あるディレクトリ内に作成されたファイルがディレクトリから継承するアクセス権

- ACLエントリ

ACL_USER_OBJ	u::rwx	ファイル所有ユーザに対するアクセス権
ACL_USER	u:ユーザ名:rwx	特定ユーザに対するアクセス権
ACL_GROUP_OBJ	g::rwx	ファイル所有グループに対するアクセス権
ACL_GROUP	g:グループ名:rwx	特定グループに対するアクセス権
ACL_MASK	m::rwx	ACL_USER, ACL_GROUP, ACL_GROUP_OBJでの最大のアクセス権
ACL_OTHER	o::rwx	上記に当てはまらないその他のユーザに対するアクセス権

ACL設定コマンド(1)

ACLの設定(setfaclコマンド)

【書式】 setfacl [オプション] ACLエントリ ファイル名

-b	全ての拡張ACLエントリを削除する
-k	デフォルトACLを削除する
-n	実効権マスクを再計算しない
-d	全ての操作をデフォルトACLに適用する
--restore=ファイル名	「getfacl -R」によって作成されたバックアップを復旧する
--test	結果のACLのみをテスト表示する
-R	全てのファイルとディレクトリに対して再帰的に操作を適用する

-m ACLエントリ	ファイル、ディレクトリのアクセス権を変更する
-x ACLエントリ	ACLエントリを削除する

ACL設定コマンド(2)

ACLの確認(getfaclコマンド)

【書式】 getfacl [オプション] ファイル名

-d	デフォルトアクセス制御リストを表示する
--all-effective	全ての実効権のコメントを表示する
--no-effective	実効権のコメントを表示しない
--skip-base	基本ACLエントリしか持たないファイルをスキップする
-R	全てのファイルとディレクトリのACLを再帰的に一覧表示する

ACLの設定例

```
$ ls -l aclfile
-rw------- 1 mickey mickey 0 1月 28 13:30 aclfile

$ getfacl aclfile
# file: aclfile
# owner: mickey
# group: mickey
user::rw-
group::---
other::---

$ setfacl -m u:minnie:r aclfile
$ ls -l aclfile
-rw-r----+ 1 mickey mickey 0 1月 28 13:30 aclfile

$ getfacl aclfile
# file: aclfile
# owner: mickey
# group: mickey
user::rw-
user:minnie:r--
group::---
mask::r--
other::---
```

ACL設定の確認

通常のパーミッションと同じ

minnieユーザにファイルへの読み込みを許可

ACL設定が存在

ACLマスク

- ACLマスク
  - ACL_USER, ACL_GROUP_OBJ, ACL_GROUP 型のエントリで 許可される最大のアクセス権
  - ACLの設定は保持しながら一時的にアクセス制限を変更できる

```
$ setfacl -m m::--- aclfile

$ getfacl -all-effective aclfile
# file: aclfile
# owner: mickey
# group: mickey
user::rw-
user:minnie:r--                 #effective:---
group::---                      #effective:---
mask::---
other::---
```

実効権マスクの値が実際のアクセス権となる

minnieユーザへのACL設定は保持されている

ACL情報を持つファイルのコピー・移動

- cpコマンド
  - 「-p」オプションまたは「-a」オプションでパーミッションを保持する

```
$ cp -p aclfile aclfile2
$ ls -l aclfile2
-rw-------+ 1 mickey mickey 0 1月 28 13:35 aclfile2
$ getfacl aclfile2
# file: aclfile
# owner: mickey
# group: mickey
user::rw-
user:minnie:r--                 #effective:---
group::---                      #effective:---
mask::---
other::---
```

- mvコマンド
  - mvコマンドは常にパーミッションを保持する

```
$ mv aclfile dir1
$ ls -l dir1/aclfile
-rw-------+ 1 mickey mickey 0 1月 28 13:35 aclfile2
```

ACL情報を持つファイルのバックアップ

- ACL情報を持つファイルのバックアップ
  - tarコマンドやdumpコマンドではACL情報をバックアップできない
  - 以下のいずれかの方法を使用する
    - ファイル拡張属性を含めたバックアップに対応した「star」の使用
    - getfacl, setfaclでACL情報をファイルとは別に保存

starを使用したバックアップ・リストア実行例

```
# cd /home
# star cf /tmp/home.star . -H=exustar -acl
star: 4 blocks + 0 bytes (total of 40960 bytes = 40.00k).

# mkdir /restore
# cd /restore
# star xf /tmp/home.star -acl
star: current `./' newer.
star: 4 blocks + 0 bytes (total of 40960 bytes = 40.00k).
# ls -l mickey
合計 8
-rw-------+ 1 mickey mickey    0 1月 28 13:35 aclfile2
drwxrwxrwx  2 mickey mickey 4096 1月 28 13:40 dir1/
```

株式会社クイ・シー・シー

LPICレベル3 303　技術解説

主題324　ネットワークセキュリティ

324.1 侵入検出　重要度4
324.2 ネットワークセキュリティスキャン機能　重要度5

Linux Professional Institute

ネットワークベースIDS –Snort–

Snortとは

- GPLライセンスで配布されるオープンソースIDS
- 豊富なプリプロセッサ
  - portscan, frag2, stream4など
- 豊富な出力プラグイン
  - テキスト、バイナリ、XML、DBへの出力に対応

ネットワークベースIDS　–Snort–

■Snortのインストールに必要なソフトウェア

- libpcap　パケットキャプチャライブラリ
- PCRE　Perl互換の正規表現をC言語で実装したライブラリ
- libdnet　一般的なネットワークAPI
- barnyard2　Snortの出力ファイルを出力プラグインへ転送するツール
- daq　データをキャプチャするために必要なライブラリ

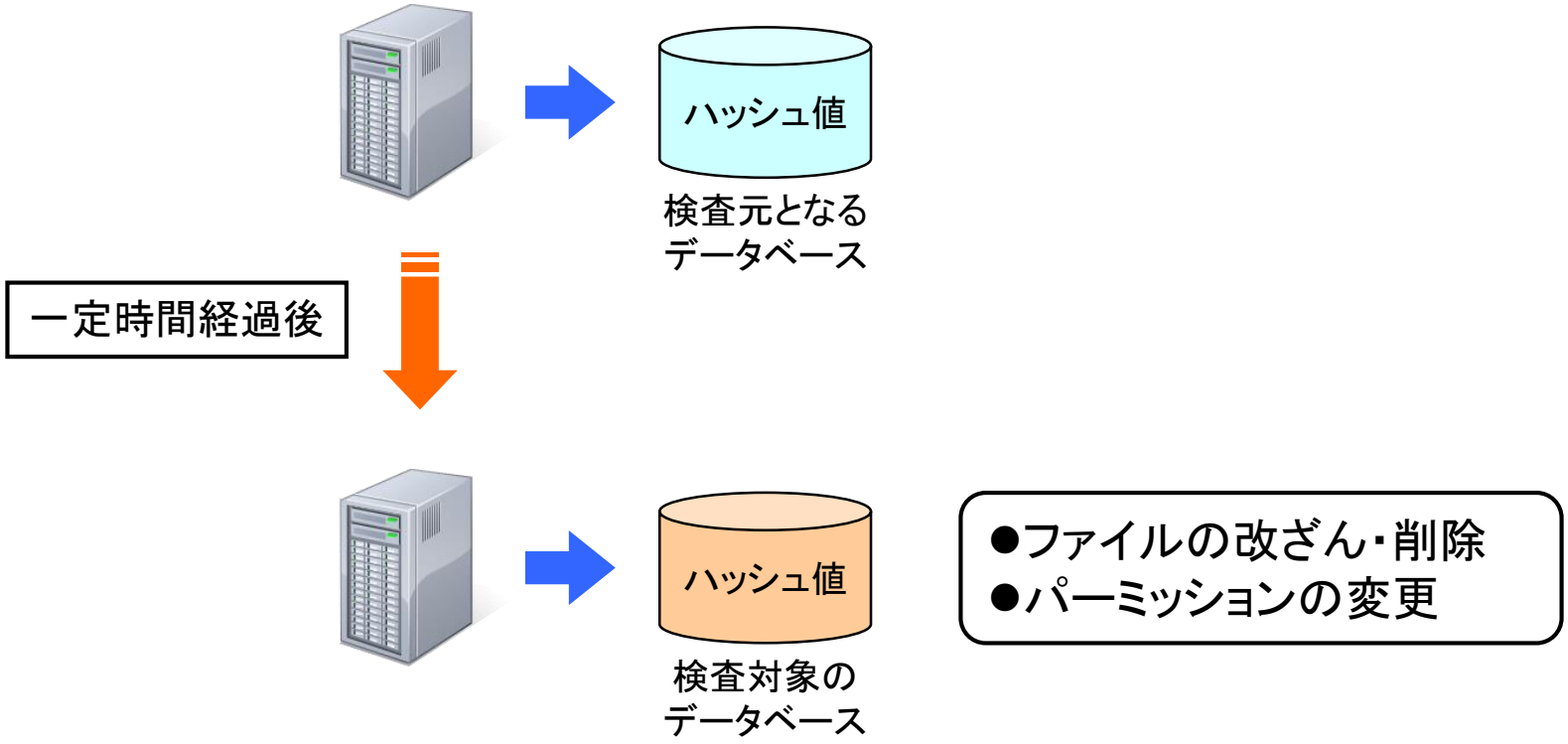
■Sourcefire　VRTルールの入手

- Subscribers（有償）
- Registerd（無償）

ホストベースIDS –Tripwire–

Tripwireとは

- ファイルやディレクトリの改ざんを検知
- 予期しないファイルの変更を管理者に通知



Tripwire関連ファイル（1）

Tripwire関連ファイル

/etc/tripwire配下にあるファイル

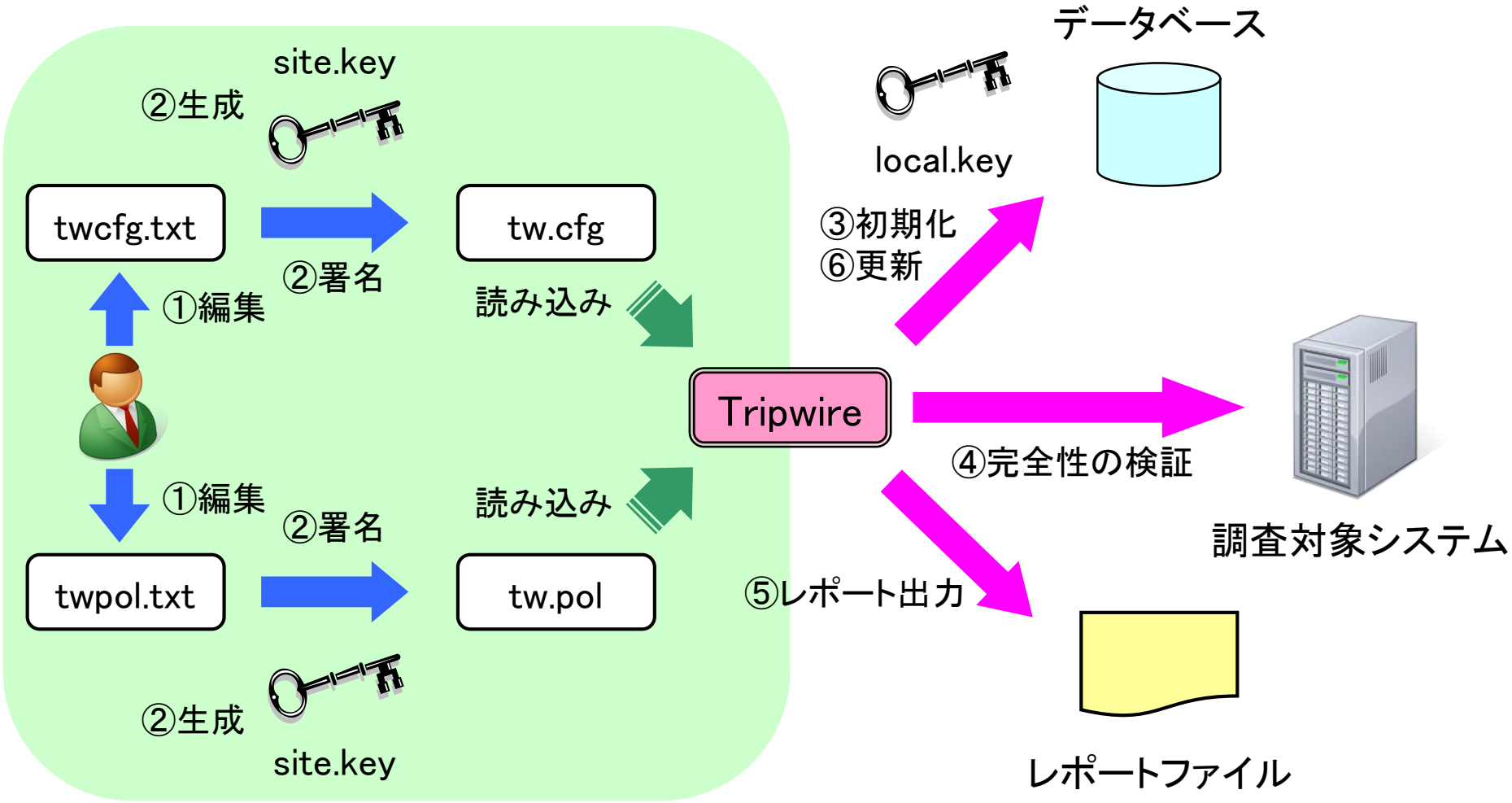
site.key	サイトキー ［設定ファイル・ポリシーファイルで使用］
$HOSTNAME-local.key	ローカルキー ［データベースで使用］
twcfg.txt	Tripwire設定ファイル
tw.cfg	署名済みの設定ファイル
twpol.txt	ポリシーファイル
tw.pol	署名済みのポリシーファイル

/var/lib/tripwire配下にあるファイル

$HOSTNAME.twd	チェック用データベース
reportディレクトリ	レポートファイル

ホストベースIDS –Tripwire–

Tripwireの仕組み



①Tripwire設定ファイル

twcfg.txtファイル

- 通常はデフォルトのまま使用して問題ない

```
ROOT                    =/usr/sbin
POLFILE                 =/etc/tripwire/tw.pol
DBFILE                  =/var/lib/tripwire/$(HOSTNAME).twd
REPORTFILE              =/var/lib/tripwire/report/$(HOSTNAME)-$(DATE).twr
SITEKEYFILE             =/etc/tripwire/site.key
LOCALKEYFILE            =/etc/tripwire/$(HOSTNAME)-local.key
EDITOR                  =/bin/vi
LATEPROMPTING           =false
LOOSEDIRECTORYCHECKING  =false
MAILNOVIOLATIONS        =true
EMAILREPORTLEVEL        =3
REPORTLEVEL             =3
MAILMETHOD              =SENDMAIL
SYSLOGREPORTING         =false
MAILPROGRAM             =/usr/sbin/sendmail -oi -t
```

Tripwireの保存場所
ポリシーファイルの指定
データベースファイルの指定
レポートファイルの指定

①Tripwireポリシーファイル

twpol.txtファイル

- 監視対象のファイル・ディレクトリ、監視方法を定義

```
@@section GLOBAL
TWROOT=/usr/sbin;
TWBIN=/usr/sbin;
TWPOL="/etc/tripwire";
TWDB="/var/lib/tripwire";
TWSKEY="/etc/tripwire";
TWLKEY="/etc/tripwire";
TWREPORT="/var/lib/tripwire/report";
HOSTNAME=station200.example.com;

@@section FS
SEC_CRIT       = $(IgnoreNone)-SHa ;
SEC_SUID       = $(IgnoreNone)-SHa ;
SEC_BIN        = $(ReadOnly) ;
SEC_CONFIG     = $(Dynamic) ;
SEC_LOG        = $(Growing) ;
SEC_INVARIANT  = +tpug ;
SIG_LOW        = 33 ;
SIG_MED        = 66 ;
SIG_HI         = 100;

...(以下略)
```

Tripwireに関するパスを変数として定義

プロパティを変数として定義

(参考)ポリシーファイルのプロパティ

a	アクセスタイムスタンプ
b	割り当てられているブロック数
c	iノードのタイムスタンプ
d	iノードが保存されているデバイスID
g	ファイルの所有グループ
i	iノード番号
l	ファイルサイズの増加
m	ファイルの更新日時
n	リンク数

p	パーミッション
r	iノードが指しているデバイスID
s	ファイルサイズ
t	ファイルの種類
u	ファイルの所有ユーザ
C	CRC-32ハッシュ値
H	Havalハッシュ値
M	MD5ハッシュ値
S	SHAハッシュ値

あらかじめ用意されている変数

ReadOnly	読み取り専用ファイル	+pinugtsdbmCM-rlacSH
Dynamic	動的に変化するファイル	+pinugtd-srlbamcCMSH
Growing	サイズが増加するファイル	+pinugtdl-srbamcCMSH
Device	デバイスファイル	+pugsdr-intlbamcCMSH
IgnoreAll	全ての属性を無視するファイル	-pinugtsdrlbamcCMSH
IgnoreNone	全ての属性を監視するファイル	+pinugtsdrbamcCMSH-l

②キーファイルの生成とファイルの署名

tripwire-setup-keyfilesスクリプト実行例

```
# tripwire-setup-keyfiles

...
Creating key files...
...
Enter the site keyfile passphrase: abcde
Verify the site keyfile passphrase: abcde
Generating key (this may take several minutes)...Key generation complete.
...
Enter the local keyfile passphrase: 123456
Enter the local keyfile passphrase: 123456
Generating key (this may take several minutes)...Key generation complete.

---------------------------------------------
Signing configuration file...
Plese enter your site passphrase: abcde
Wrote configuration file: /etc/tripwire/tw.cfg
...
Signing configuration file...
Plese enter your site passphrase: abcde
Wrote policy file: /etc/tripwire/tw.pol

...(以下略)
```

キーファイルの生成

サイトキーパスフレーズを入力

ローカルキーパスフレーズを入力

サイトキーパスフレーズを入力

設定ファイル・ポリシーファイルへの署名

③データベースの初期化

tripwire --init

```
# tripwire --init
Please enter your local passphrase: abcdefg
Parsing policy  file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
Wrote database file: /var/lib/tripwire/station200.example.com.twd
The database was successfully generated.

# ls /var/lib/tripwire
report/  station200.example.com.twd
```

サイトキーパスフレーズを入力

データベースファイル作成

④完全性の検証

tripwire --check

- 完全性の検証時に同時にレポートを生成

```
# tripwire --check
Parsing policy  file: /etc/tripwire/tw.pol
Wrote report file: /var/lib/tripwire/station200.example.com.-20120128-140000.twr

Open Source Tripwire(R) 2.4.2.2 Integrity Check Report

Report generated by:              root
Report created on:                2012年01月28日 14時00分00秒
Database last updated on:         2012年01月28日 13時30分00秒

================================================================================
Report Summary
================================================================================
Host name:                        station200.example.com
Host IP address:                  192.168.1.200
Host ID:                          None
Policy file used:                 /etc/tripwire/tw.pol
Configuration file used:          /etc/tripwire/tw.cfg
Database file used:               /var/lib/tripwire/station200.example.com.twd
Command line used:                tripwire -check

...(略)
```

⑤レポート出力

```
# twprint --print-report --twrfile station200.example.com-20120128-140000.twr
================================================================================
Rule Summary
================================================================================
--------------------------------------------------------------------------------
Section: Linux File System
--------------------------------------------------------------------------------
  Rule Name              Severity Level   Added   Removed  Modified
  ---------              --------------   -----   -------  --------
* User Directory         100              0       0        1
  (/home)
* Configuration Files    100              0       0        9
  (/etc)
Total objects scanned:  2625
Total violations found:  10
...
--------------------------------------------------------------------------------
Rule Name: User Directory
Severity Level: 100
--------------------------------------------------------------------------------
Modified:
"/home"
--------------------------------------------------------------------------------
Rule Name: Configuration Files
Severity Level: 100
--------------------------------------------------------------------------------
Modified:
"/etc"
"/etc/group"
...
"/etc/passwd"
...(以下略)
```

⑥データベースの更新

tripwire --update

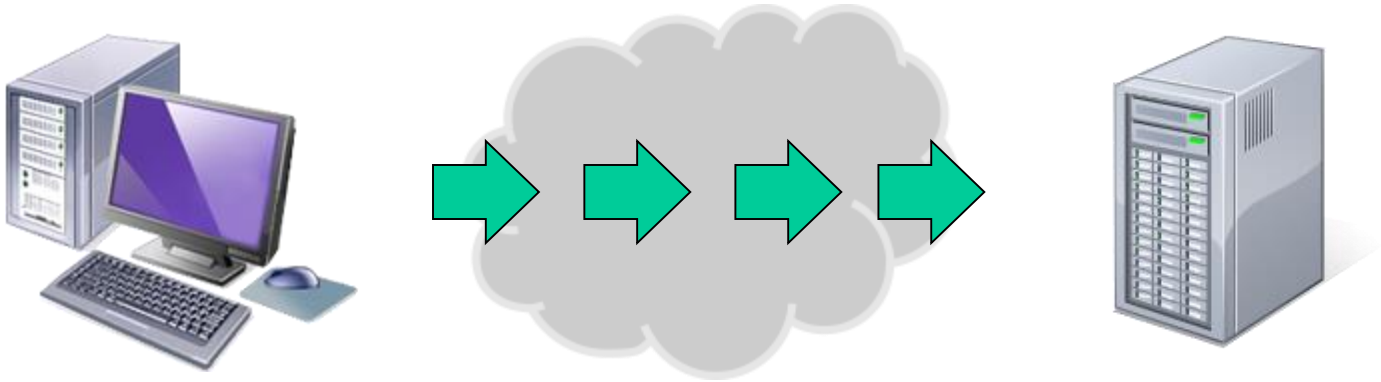
- 検査対象のデータベースを更新する

```
# cd /var/lib/tripwire/report
# tripwire --update -a --twrfile station200.example.com-20120128-133000.twr
Please enter your local passphrase: abcdefg
Wrote database file: /var/lib/tripwire/station200.example.com.twd
```

セキュリティスキャナ Nessus

Nessusの概要

- Tenable Network Security社が管理する脆弱性スキャナ
- ツールのダウンロードは無償だが、営利目的での利用は有償
- 脆弱性のチェックはプラグインを使用することにより実施
- 多種多様なプラグインが用意されている



NASL(Nessus Attack Scripting Language)

- Nessusプラグインスクリプトを記述するための言語

セキュリティスキャナ Nessus

Nessus使用上の注意点

- スキャン対象ホストやネットワークへの負荷
  - 対象ホストがダウン、ネットワークトラフィックが混雑する可能性がある
  - 管理対象ホスト以外へは使用しない

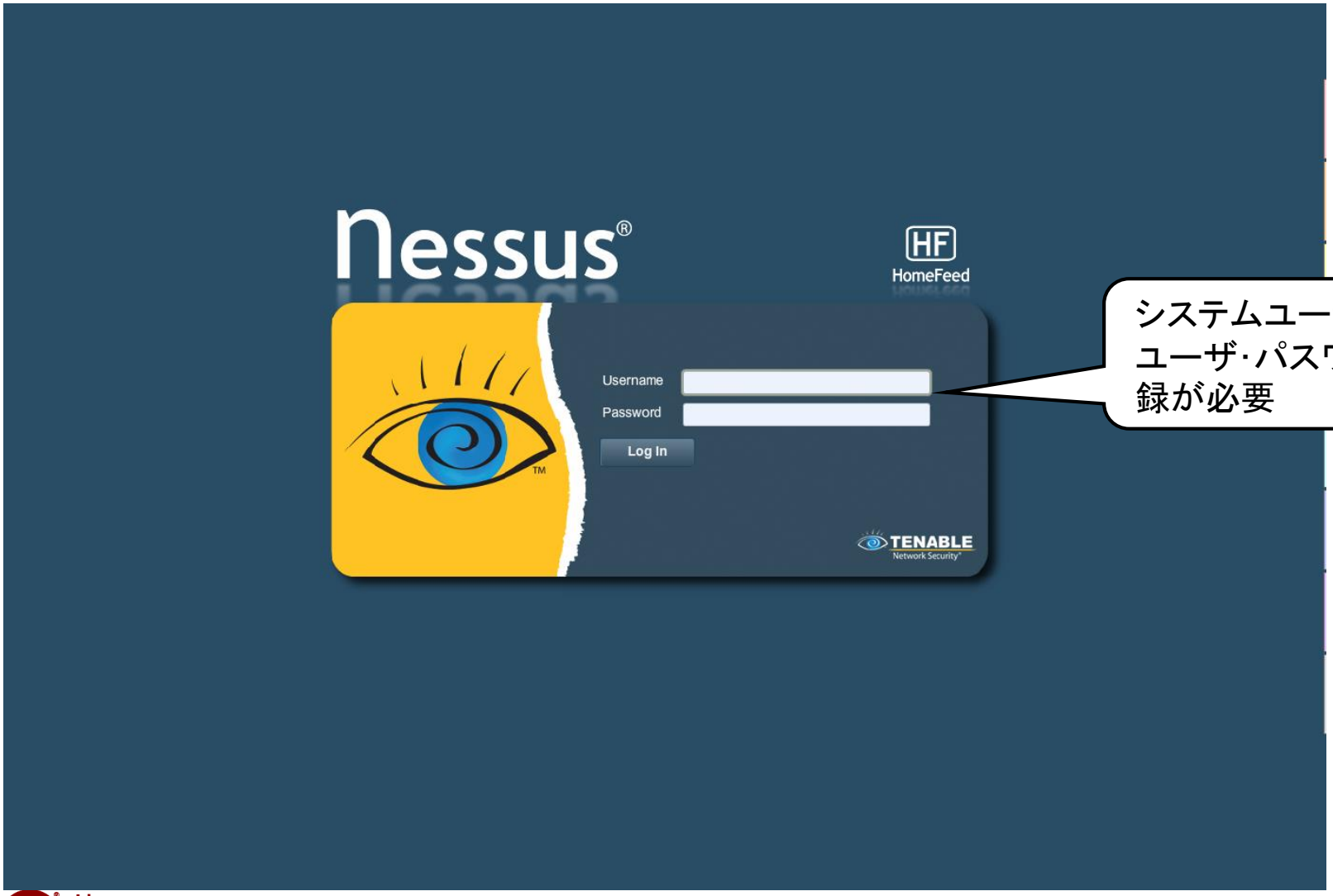
- 稼働中のファイアウォールやIDSへのスキャン
  - アラートが頻発し、攻撃とみなされることがある
  - パーソナルファイアウォールで調査がブロックされることがある

Nessusインストールに必要なもの

- Webサーバー
- Flashプラグイン

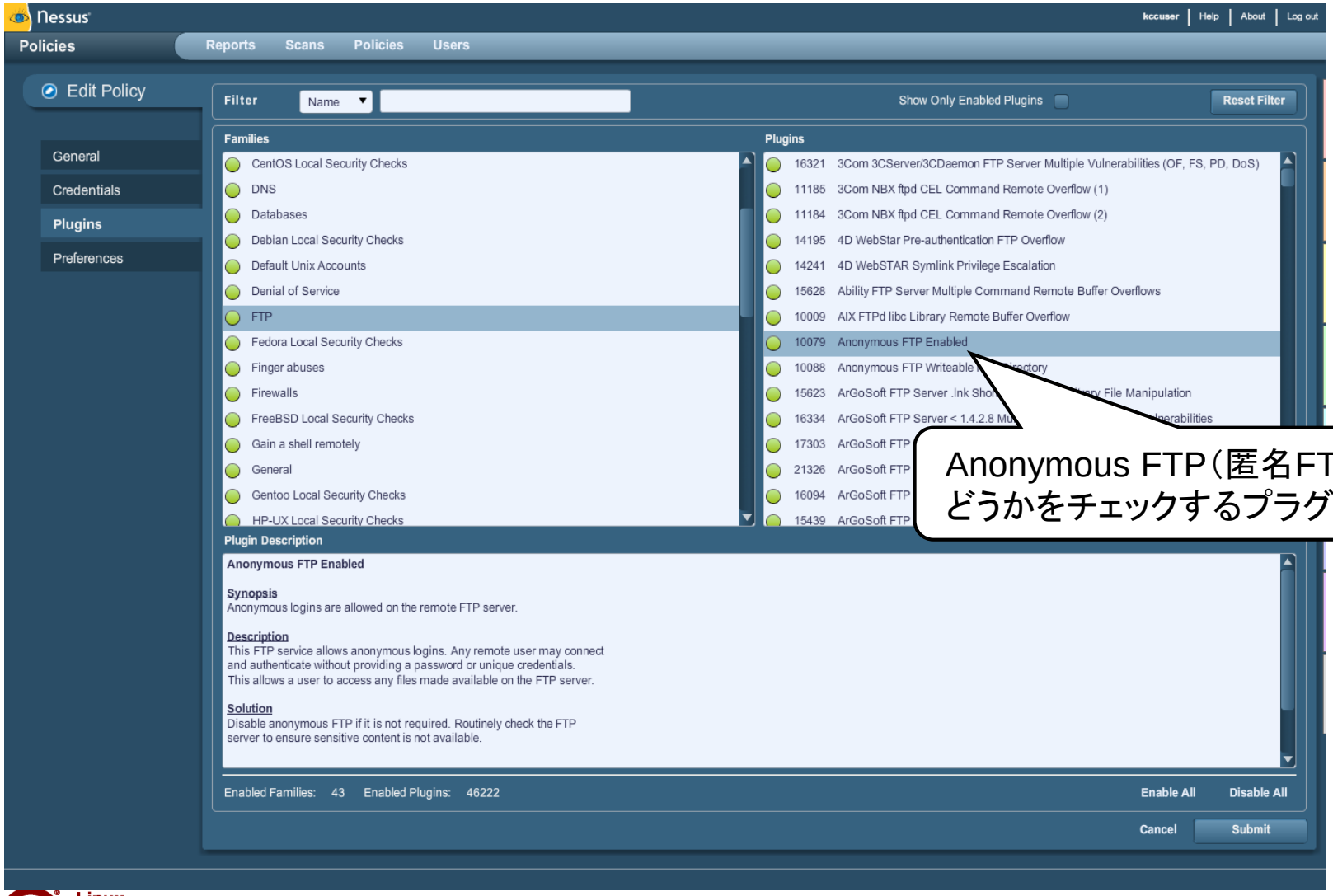
セキュリティスキャナ Nessus

■ログイン画面



セキュリティスキャナ Nessus

■プラグイン一覧



Anonymous FTP（匿名FTP）が有効かどうかをチェックするプラグイン

セキュリティスキャナ Nessus

■チェックレポート

- Synopsis: 概要
- Description: 脆弱性の説明
- Solution: 解決策
- See Also: 関連サイトURL
- Risk Factor: 危険度(High, Medium, Low)
- Plugin Output: プラグインからの出力内容
- Plugin Publication Date: プラグインの公開日
- Plugin Last Modification Date: プラグインの最終更新日

ポートスキャナ Nmap

Nmapの概要

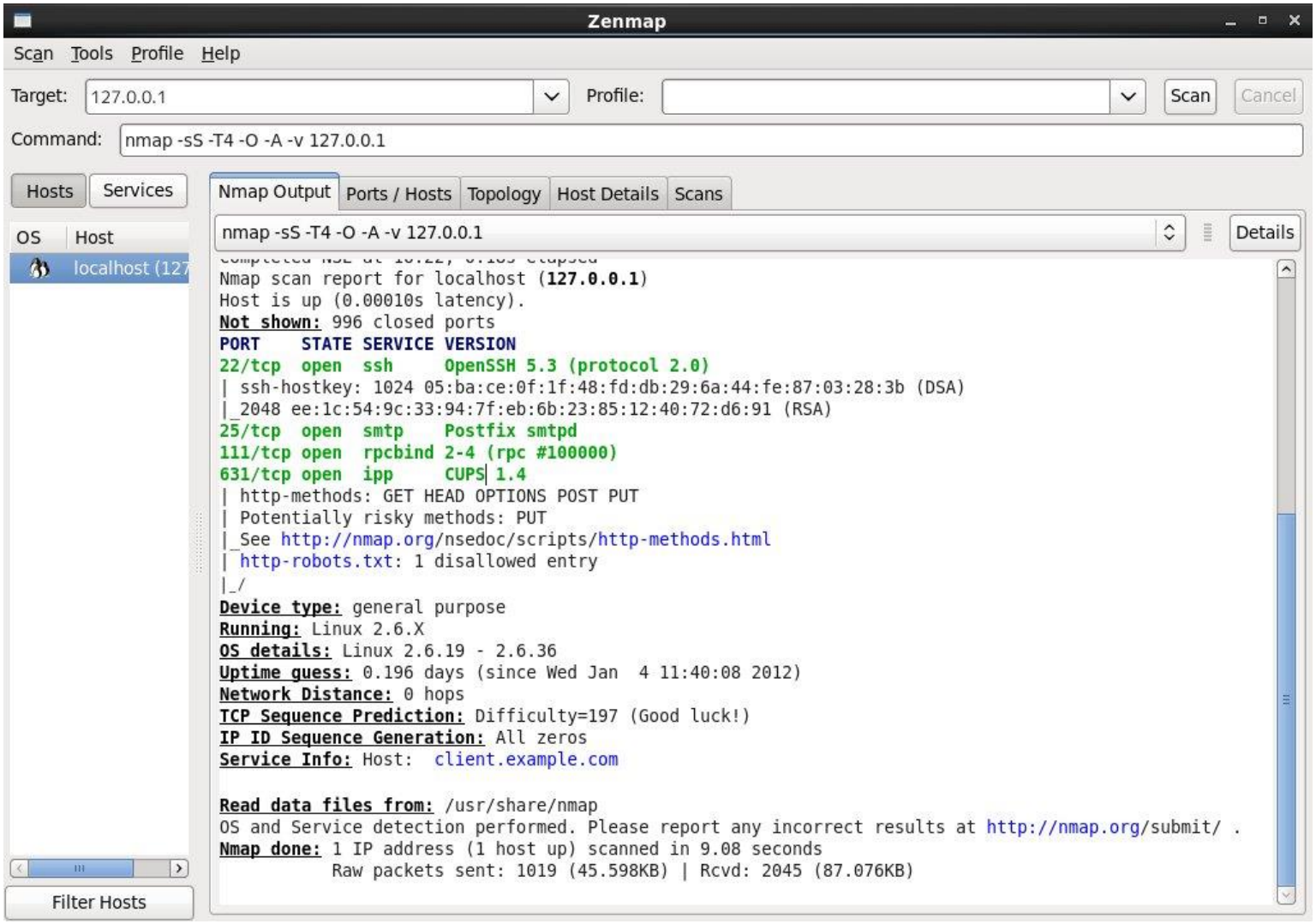
- 監視対象コンピュータでどのポートが開放されているかを調査するツール
- 開放ポートによって稼動しているサービスが特定できる

【書式】 nmap [スキャンタイプ...] [オプション] {ターゲットの指定}

-sS	TCP SYNスキャンを利用する
-sT	TCP connect()スキャンを利用する
-sF	ステルスFINスキャンを利用する
-A	OSとバージョンの検出を可能にする
-O	OS検出を実行
-T4	処理を高速に実行する
-v	冗長性レベルを上げる

ポートスキャナNmap

Zenmap GUIでの出力例



パケットキャプチャ Wireshark

Wiresharkとは

- GUIインターフェイスで動作するパケットキャプチャツール
- コンピュータが通信する際にパケットを収集し解析することができる
- CUIではtsharkコマンドを実装

Wiresharkの主な機能

- TCP/IPをはじめとする数多くのプロトコルに対応
- 特定の接続をキャプチャ/表示するためのフィルタ機能
- パケットヘッダを階層的に解析可能
- TCPストリームの組み立て可能
- 詳細な統計情報の表示、グラフ作成機能

tcpdump

概要

- コマンドラインで使用するパケットキャプチャツール

実行例

```
# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
16:29:52.332843 IP 192.168.1.100 > 192.168.1.123: ICMP echo request, id 5485, seq
1, length 64
16:29:52.359433 IP 192.168.1.123 > 192.168.1.100: ICMP echo reply, id 5485, seq 1,
length 64
16:29:53.334391 IP 192.168.1.100 > 192.168.1.123: ICMP echo request, id 5485, seq

...(中略)

4, length 64
16:29:55.337177 IP 192.168.1.123 > 192.168.1.100: ICMP echo reply, id 5485, seq 4,
length 64

8 packets captured
8 packets received by filter
0 packets dropped by kernel
```

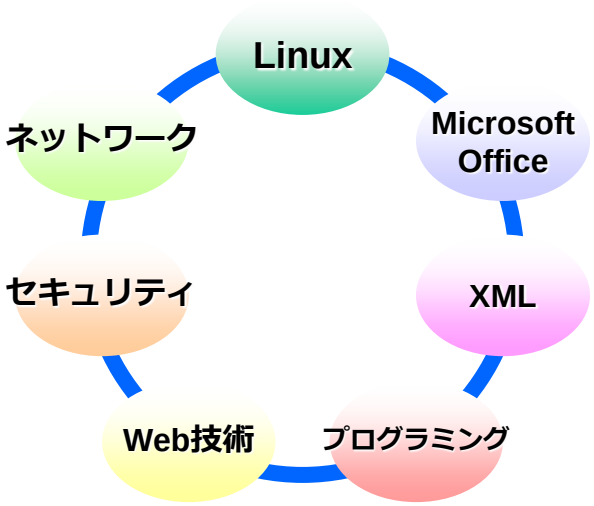
株式会社ケイ・シー・シー

■カスタマイズ研修のご案内

- LPIC試験対策研修
- Linux基礎、Linuxサーバ構築
- その他、ネットワーク・セキュリティ・XML・Web技術など
  各種IT研修をカスタマイズして提供

弊社研修サービスホームページ

http://www.kcc.co.jp/lpic/



IT技術研修

- ●Linux（基礎・システム管理・サーバ構築）
- ●ネットワーク（TCP/IP・LAN/WAN・無線技術）
- ●セキュリティ（技術解説・セキュリティマネジメント）
- ●XML（XML/DTD・XSLT・XML Schema）
- ●Web技術（HTML・CSS・JavaScript・Ajax）
- ●プログラミング（C・Java・Android・Perl・PHP・Ruby）
- ●Microsoft Office（基礎/応用・VBA）

資格試験対策

- ◆LPICレベル1～3
- ◆XMLマスター・ベーシック
- ◆CompTIA A+・Network+・Security+
- ◆Ruby技術者認定試験 Silver
- ◆情報処理技術者試験

ご清聴ありがとうございました