Monthly Research

SELinux 再入門
-基礎編-

株式会社ＦＦＲＩ
http://www.ffri.jp

Ver 2.00.01

SELinux再入門のすすめ

- 近年、仮想化やコンテナ、AndroidなどでSELinuxを使ったセキュリティ強化が進んでいる
- 一方、サーバ用途において、SELinuxを無言で無効化してきた技術者は数多い
  - Web検索のレコメンドで一目瞭然である
- 本資料は、最新のSELinux応用事例を理解するための準備資料としての活用を想定
- なお次回以降、数回に渡り最新のSELinux応用事例を調査する予定

SELinux再入門

- SELinuxの概要

- SELinuxのアクセス制御モデル
  - Type Enforcement (TE)
  - Role-based Access Control (RBAC)
  - Multi-level Security (MLS) / Multi-category Security (MCS)

- SELinuxセキュリティポリシー
  - Strict Policy (deprecated)
  - Targeted Policy
  - Minimum Policy
  - MLS/MCS Policy

SELinux再入門 –基礎編-

SELinuxの概要

SELinuxとは

- NSA (National Security Agency) が開発したLinuxカーネルのセキュリティ拡張機能
  - Linux Security Module(LSM)を使って実装されている
  - FLASKにより、多様なアクセス制御方式をサポートする

- 軍事レベルのセキュリティをLinuxで実現するために開発された
  - 例えば、SELinuxがサポートするアクセス制御方式の一つであるMulti-level securityは軍隊における機密保持のためのアクセス制御モデル

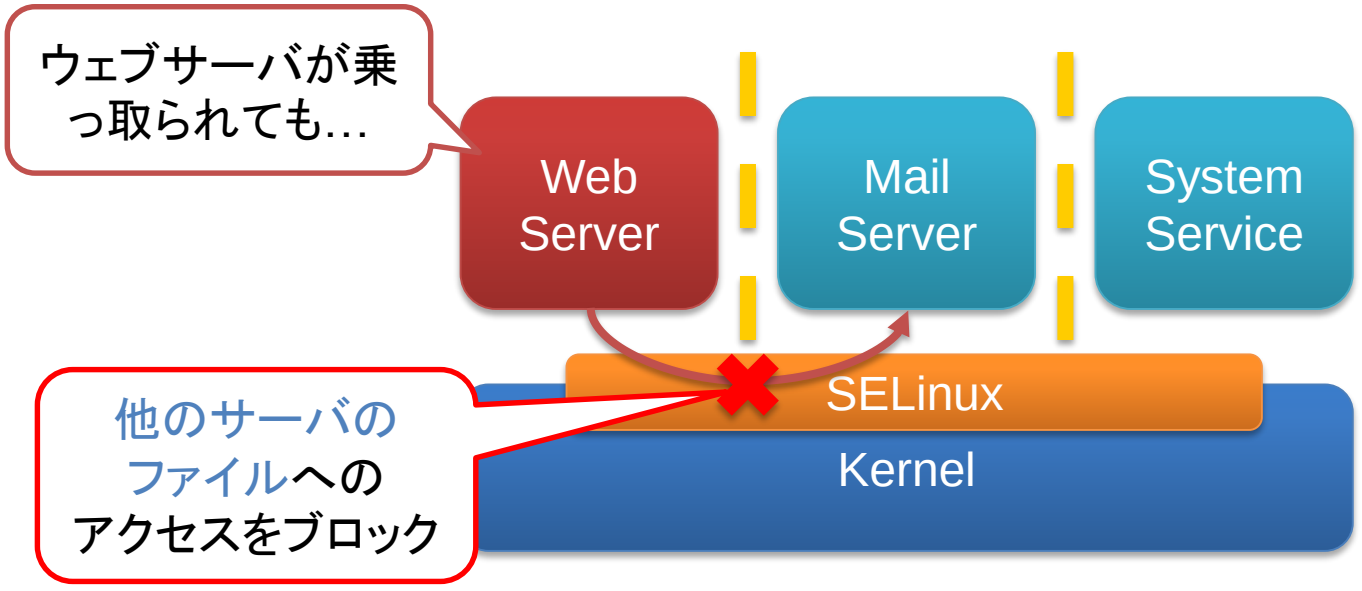
- 機密保全とシステム保護は不可分という思想
  - アプリケーションに許可する動作は最小限に
  - root権限は大きなセキュリティホールになり得る

SELinux による強制アクセス制御

- すべてのリソースにSELinuxコンテキストを付与する
- OSカーネルレベルのリファレンスモニタ（LSM）で、プロセスの挙動をすべて監視・検査する
  - 高速化のため、アクセス制御判定をユーザーランドでキャッシュするAccess Vector Cacheという仕組みが導入されている
- アプリケーションレベルのアクセス制御でもSELinuxコンテキストを使ったアクセス制御をおこなうため、様々なアプリケーションにSELinuxによるアクセス制御が組み込まれている
  - X Window System, SE-PostgreSQL, systemd, d-bus

SELinuxによるセキュリティ効果

- プロセス単位（not ユーザ）のカーネルレベルアクセス制御
- 強力なプロセス隔離
- root特権の無効化



SELinuxができないこと

- マルウェアの駆除
- 侵入検知
- メモリ保護

SELinux再入門 –基礎編-

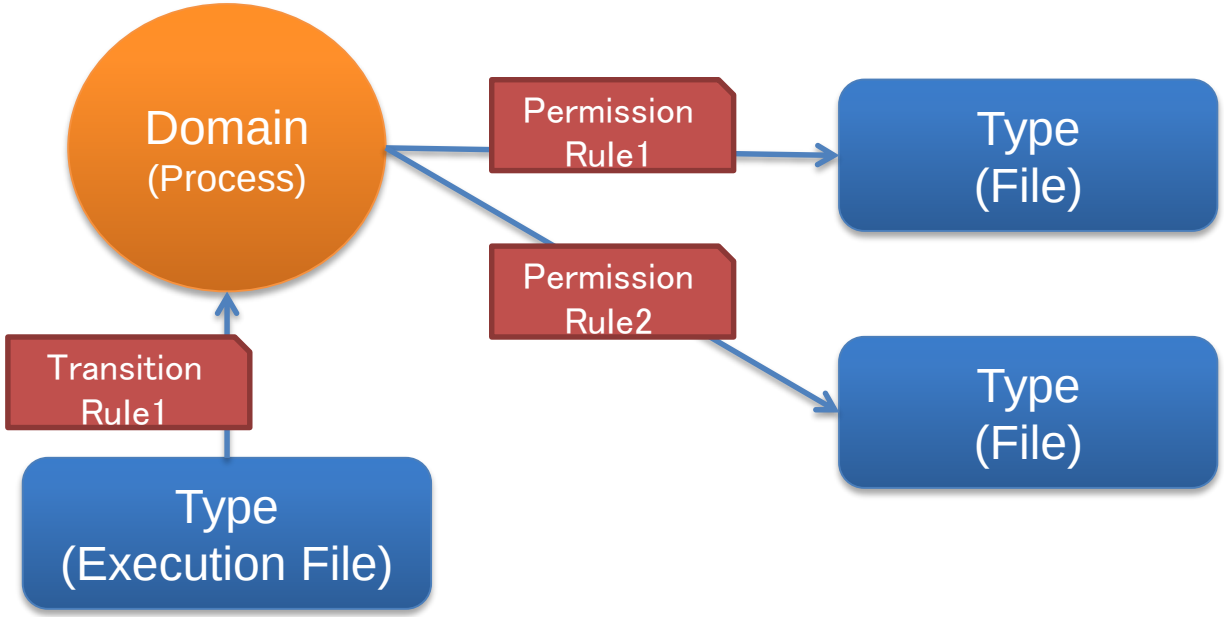
SELinuxのアクセス制御モデル

SELinuxがサポートするアクセス制御モデル

- TE: Type Enforcement
- RBAC: Role-based Access Control
- MLS/MCS: Multi-level Security/Multi-category Security

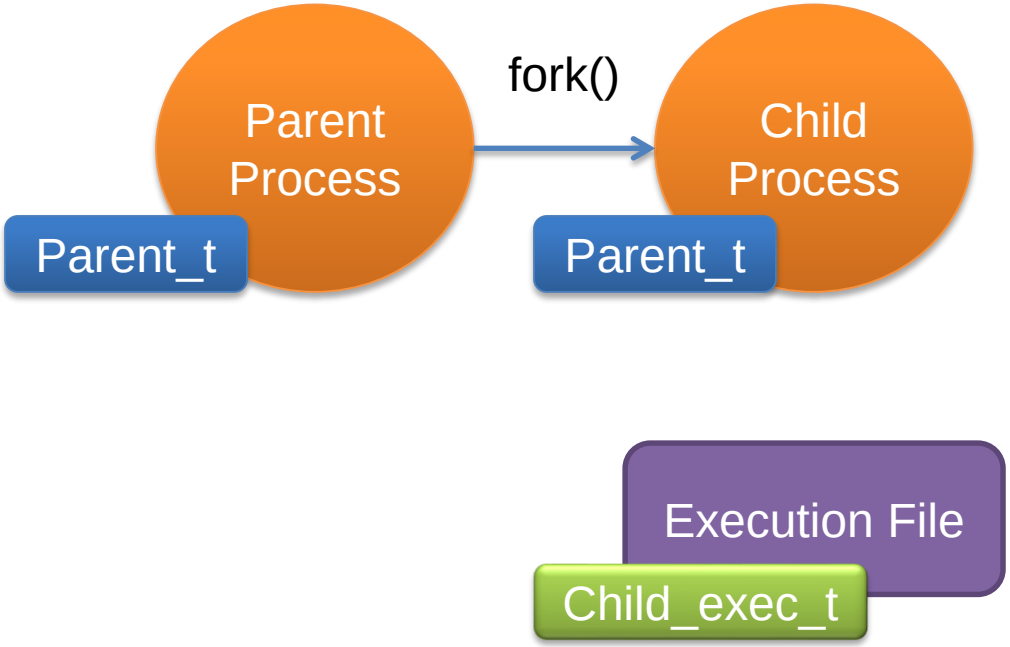
Type Enforcement

- SELinuxコンテキストとしてタイプ（Type）を定義する
- プロセスに割り当てるタイプのことをドメインとして宣言する
- タイプとドメインを使い、許可する動作をルールとして記述していく
  - パーミッションルールやドメイン遷移ルールなどがある



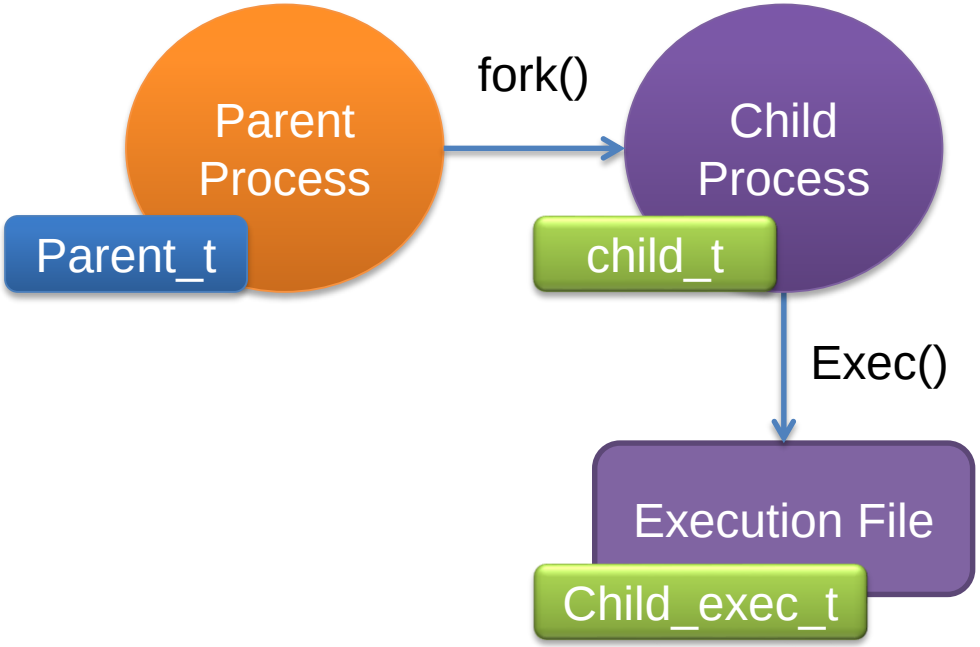
Domain transition（ドメイン遷移）

- 特に定義が無い場合、SELinuxでは親プロセスのドメインを継承して子プロセスが生成される



Domain transition（ドメイン遷移）

- 実行バイナリのタイプなどを使ってプロセス生成時にドメインを変更することをドメイン遷移という



Example1: myapp.te

```
# タイプ定義
type myapp_t;
type myapp_exec_t;

 #myapp_tはドメインである
domain_type(myapp_t)

#myapp_exec_tから起動したらそのプロセスは myapp_tドメインとなる
domain_entry_file(myapp_t, myapp_exec_t)

#ログファイルのタイプ定義
type myapp_log_t

#アプリ外からlogを読むためのマクロ（インターフェース）
logging_log_file(myapp_log_t)

#myapp_t ドメインはログに対して読み込みと追記のみが可能
 allow myapp_t myapp_log_t:file { read_file_perms append_file_perms };
```

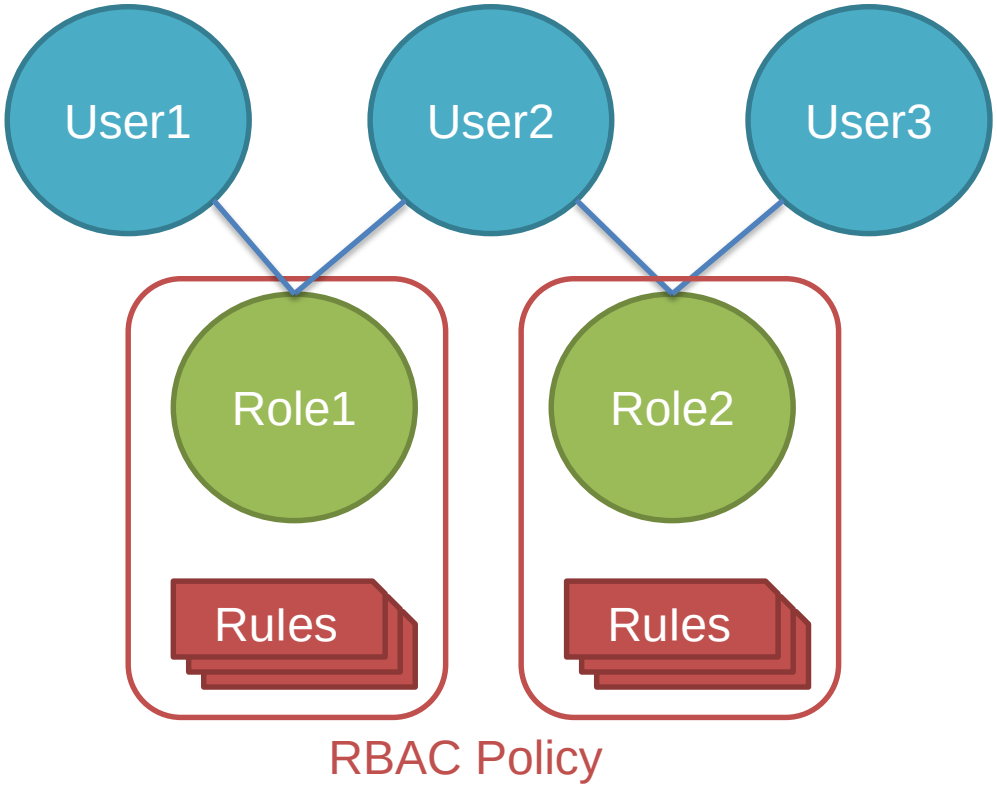
Type Enforcementのルール記述

- 非常に原始的な表現力しか持っていない
  - このため様々なマクロが用意されているが・・・

- 読解することはかなり難しく、そのポリシーがアプリケーションを過不足なく制限しているかを確かめることは困難

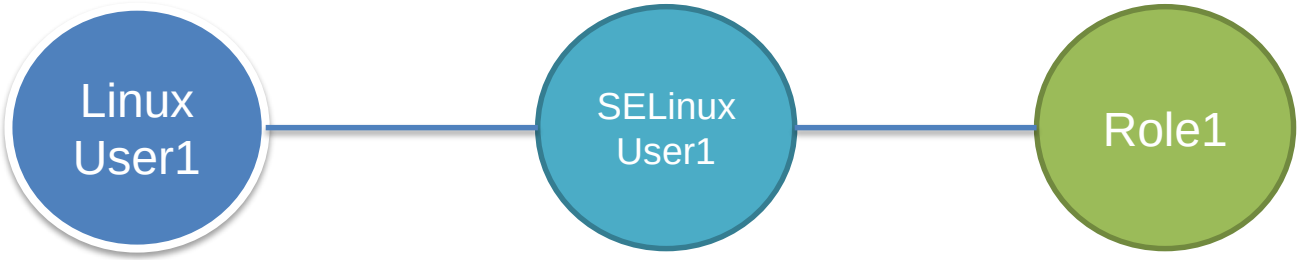
RBAC (Role-based Access Control)

- 役割 (role) とその権限を管理するアクセス制御



SELinuxユーザとロール

- SELinuxはLinuxユーザとSELinuxユーザを関連付け、更にSELinuxユーザとロールを関連づける
  - Linuxユーザは自身の権限を容易に変更できる任意アクセス制御に基づいて管理されているため



主なLinuxユーザ:
・user1
・Root

主なSELinuxユーザ:
・User_u
・root
・Staff_u
・System_u
・sysadm_u
・unconfined_u

主なロール:
・User_r
・Staff_r
・System_r
・Sysadm_r
・unconfined_r

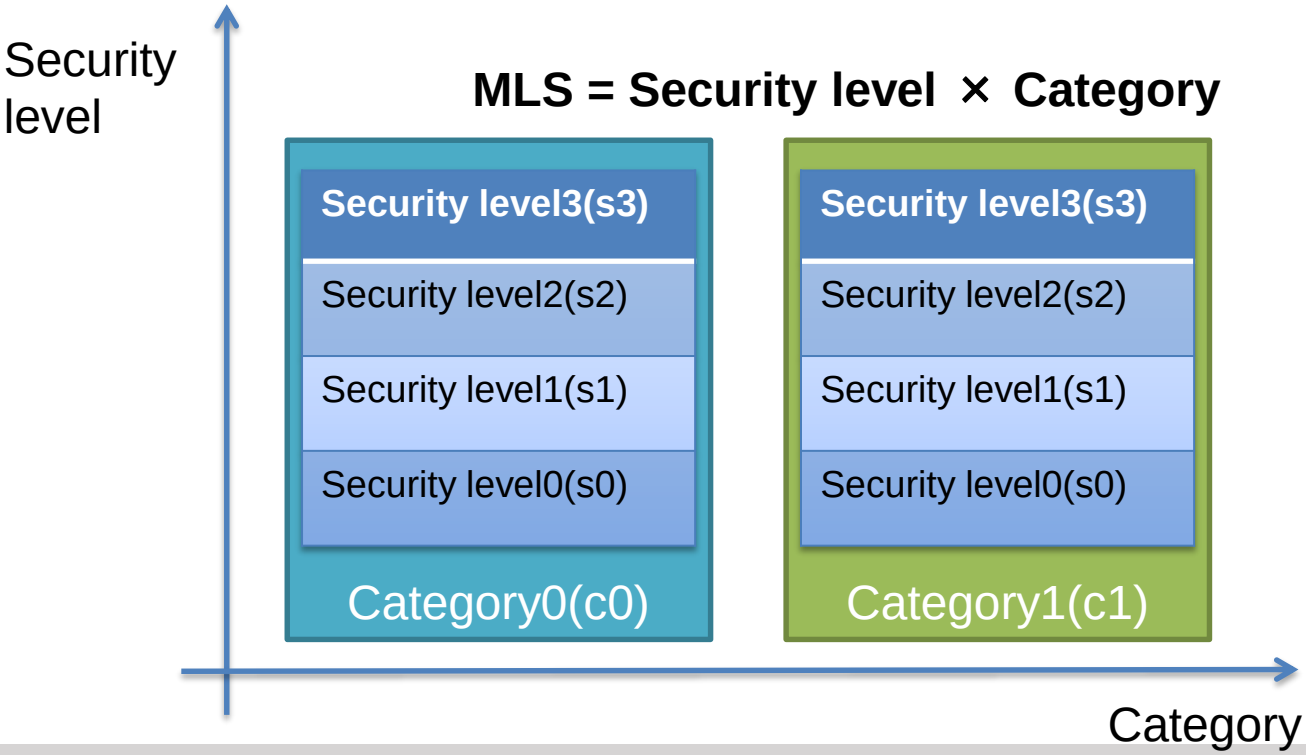
SELinux Policy Module

- 主にTEのルールをモジュール化したもの
  - モジュール単位でロードしたり、モジュールごとにPolicy Interfaceを定義でき、他のモジュールのポリシを使うことが可能
  - ポリシーの再利用性、相互利用性を高める

具体的な作り方は次を参照:
“Getting Started with Reference Policy”
http://oss.tresys.com/projects/refpolicy/wiki/GettingStarted

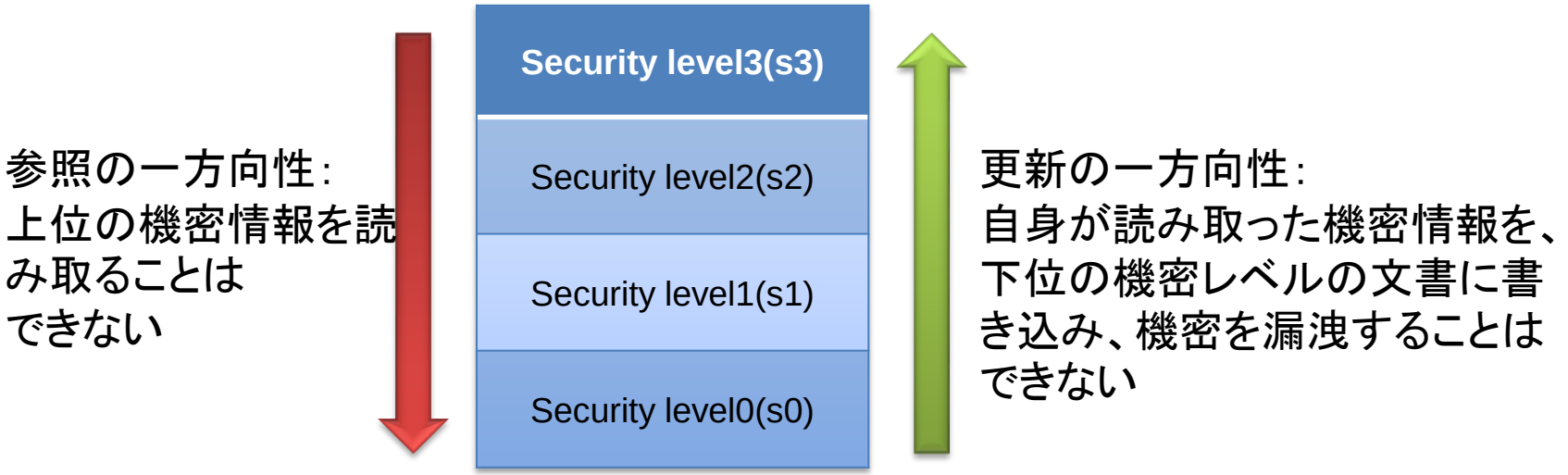
MLS/MCS (Multi Level/Category Security)

- 機密レベル(security level)とカテゴリ(category)を用いたアクセス制御
  - 機密レベルに関しては、Bell-LaPadula Modelに基づいたアクセス制御がおこなわれる



Bell-LaPadula Model

- Multi level securityの数理モデル
  - 情報の参照と更新の有無に関心を置いている
- ユーザは機密レベルが上の情報を読み取れない
  - ただし更新はできる
- ユーザーは機密レベルが同レベルもしくは下の情報を読み取れる
  - ただし更新ができない



MLS/MCSを用いたアクセス制御

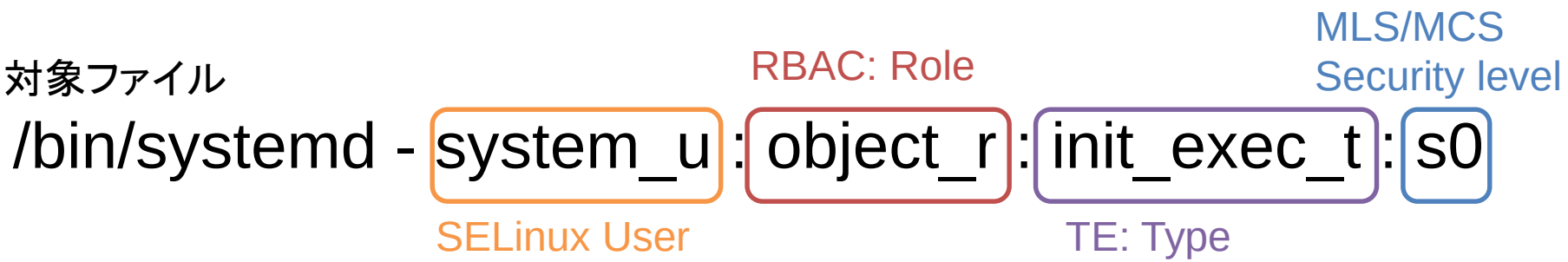
- ユーザやリソースに対してデフォルトの機密レベル・カテゴリを割り当てる
  - ユーザについては、そのユーザーのクリアランス（アクセスできる機密レベル・カテゴリの指定）も割り当てる

- 違うカテゴリのリソースにアクセスする際には、クリアランスが必要

ユーザ/ファイルに割り当てるMLSコンテキストの例

File contexts（TE, RBAC, MLS/MCS共通）

- ファイル・ソケットに割り当てるセキュリティコンテキストを列挙したもの
  - SELinuxを有効にする場合、一度すべてのファイルにセキュリティコンテキストを割り当てる必要がある



実行モード

- SELinuxは3つのモードがある
  - enforcing: SELinuxによるアクセス制御が実施される
  - permissive: ポリシーを検査するだけでアクセス制御は実施しない
  - disabled: SELinuxを無効にする

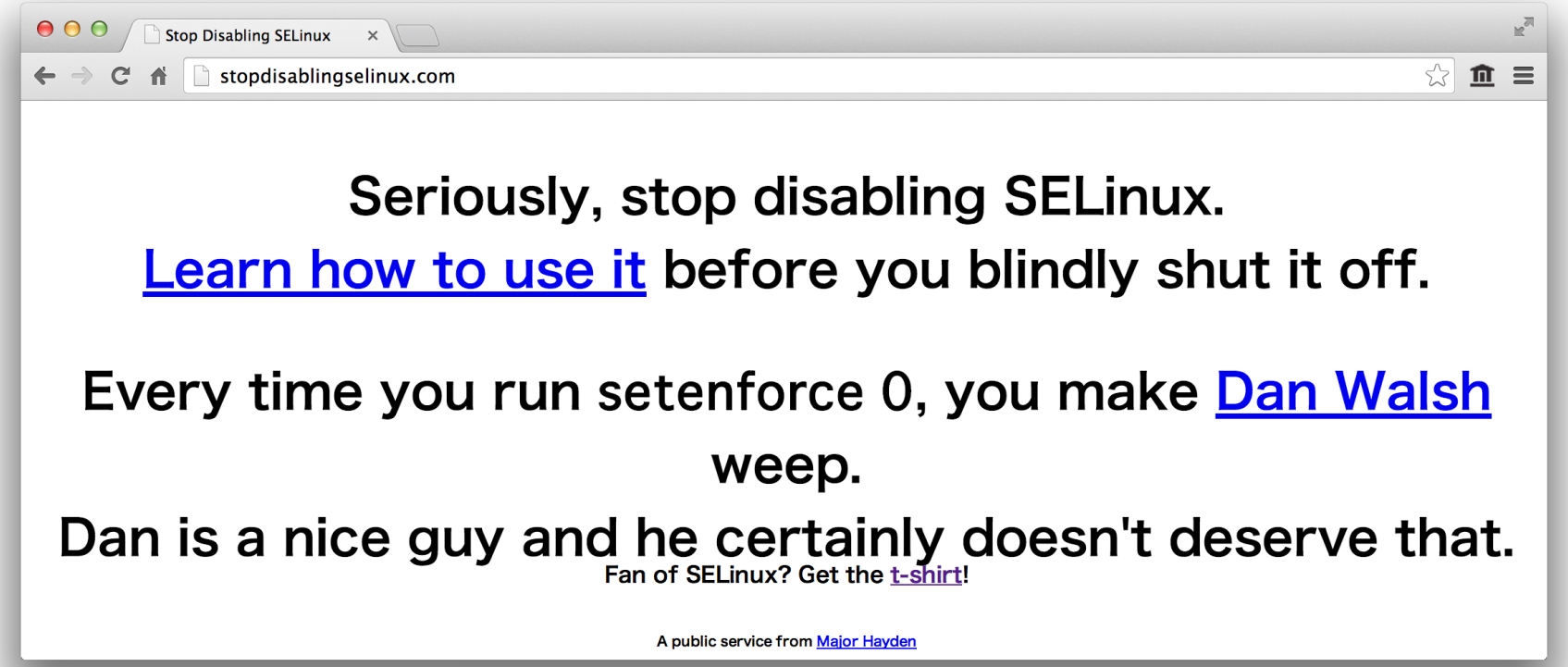
- 起動時のモードは/etc/selinux/configで変更が可能

sestatus

- 現在のSELinuxの状態や設定ファイルの場所を確認できる

```
# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      29
```

stopdisablingselinux.com



SELinux再入門 –基礎編-

SELinuxセキュリティポリシー

SELinux の代表的なセキュリティポリシー

- Type Enforcement (+RBAC)
  - Strict (deprecated)
  - Targeted
  - Minimum
- Multi Level/Category Security
  - MLS
  - MCS

※ 現行のFedora、RHEL系ではTargeted+MCSが
デフォルトのセキュリティポリシー

Strict policy (deprecated)

> *"SELinux designed to be a strict policy."* – Dan Walsh (2005)

- すべてのアプリケーションを"例外なく" 制限
  - RBACも併用し、誰が起動したかで制限が変わる
- NSAとRed Hat（開発元）の理想形
  - しかし、まともに構築・運用するにはコストがかかりすぎる
- Strictでのサーバ運用は実際的ではないということで、FedoraCore10でTargeted policyと統合された

Targeted Policy

- アプリケーションの動作をTEで制限しつつ、制限されないUnconfined_tタイプを導入したポリシー
  - Web経由の権限昇格を防ぐことに重点を置いている
  - unconfined_tが割り当てられたリソースは、実質的に動作を制限されない

- ログインシェルはunconfined_tドメインで動作
  - SELinuxによる制限を意識せずにアップデートやサーバ設定が行える

- 最新のFedora20では相当数のポリシーモジュールが有効
  - 管理者がポリシーを直接記述したりする必要はほとんどない
  - unconfined_t, unconfined_rを取り除ければstrict相当になるが・・・
    参考：http://danwalsh.livejournal.com/27885.html

Fedora20のTargeted Policyで用意されているポリシー一覧

abrt.pp, accountsd.pp, acct.pp, afs.pp, aiccu.pp, aide.pp, ajaxterm.pp, alsa.pp, amanda.pp, amtu.pp, anaconda.pp, antivirus.pp, apache.pp, apcupsd.pp, apm.pp, application.pp, arpwatch.pp, asterisk.pp, auditadm.pp, authconfig.pp, authlogin.pp, automount.pp, avahi.pp, awstats.pp, bacula.pp, bcfg2.pp, bind.pp, bitlbee.pp, blueman.pp, bluetooth.pp, boinc.pp, bootloader.pp, brctl.pp, bugzilla.pp, bumblebee.pp, cachefilesd.pp, calamaris.pp, callweaver.pp, canna.pp, ccs.pp, cdrecord.pp, certmaster.pp, certmonger.pp, certwatch.pp, cfengine.pp, cgroup.pp, chrome.pp, chronyd.pp, cipe.pp, clock.pp, clogd.pp, cloudform.pp, cmirrord.pp, cobbler.pp, collectd.pp, colord.pp, comsat.pp, condor.pp, conman.pp, consolekit.pp, couchdb.pp, courier.pp, cpucontrol.pp, cpufreqselector.pp, cron.pp, ctdb.pp, cups.pp, cvs.pp, cyphesis.pp, cyrus.pp, daemontools.pp, dbadm.pp, dbskk.pp, dbus.pp, dcc.pp, ddclient.pp, denyhosts.pp, devicekit.pp, dhcp.pp, dictd.pp, dirsrv-admin.pp, dirsrv.pp, dmesg.pp, dmidecode.pp, dnsmasq.pp, dnssec.pp, docker.pp, dovecot.pp, drbd.pp, dspam.pp, entropyd.pp, exim.pp, fail2ban.pp, fcoe.pp, fetchmail.pp, finger.pp, firewalld.pp, firewallgui.pp, firstboot.pp, fprintd.pp, freeipmi.pp, freqset.pp, fstools.pp, ftp.pp, games.pp, gear.pp, getty.pp, git.pp, gitosis.pp, glance.pp, glusterd.pp, gnome.pp, gpg.pp, gpm.pp, gpsd.pp, gssproxy.pp, guest.pp, hddtemp.pp, hostname.pp, hypervkvp.pp, icecast.pp, inetd.pp, init.pp, inn.pp, iodine.pp, ipa.pp, ipsec.pp, iptables.pp, irc.pp, irqbalance.pp, iscsi.pp, isns.pp, jabber.pp, jetty.pp, jockey.pp, kdump.pp, kdumpgui.pp, keepalived.pp, kerberos.pp, keyboardd.pp, keystone.pp, kismet.pp, ksmtuned.pp, ktalk.pp, l2tp.pp, ldap.pp, libraries.pp, likewise.pp, lircd.pp, livecd.pp, lldpad.pp, loadkeys.pp, locallogin.pp, lockdev.pp, logadm.pp, logging.pp, logrotate.pp, logwatch.pp, lpd.pp, lsm.pp, lvm.pp, mailman.pp, mailscanner.pp, man2html.pp, mandb.pp, mcelog.pp, mediawiki.pp, memcached.pp, milter.pp, mip6d.pp, miscfiles.pp, mock.pp, modemmanager.pp, modutils.pp, mojomojo.pp, motion.pp, mount.pp, mozilla.pp, mpd.pp, mplayer.pp, mrtg.pp, mta.pp, munin.pp, mysql.pp, mythtv.pp, nagios.pp, namespace.pp, ncftool.pp, netlabel.pp, netutils.pp, networkmanager.pp, ninfod.pp, nis.pp, nova.pp, nscd.pp, nsd.pp, nslcd.pp, ntop.pp, ntp.pp, numad.pp, nut.pp, nx.pp, obex.pp, oddjob.pp, openct.pp, openhpid.pp, openshift-origin.pp, openshift.pp, opensm.pp, openvpn.pp, openvswitch.pp, openwsman.pp, oracleasm.pp, osad.pp, pads.pp, passenger.pp, pcmcia.pp, pcp.pp, pcscd.pp, pegasus.pp, permissivedomains.pp, pesign.pp, pingd.pp, piranha.pp, pkcsslotd.pp, pki.pp, plymouthd.pp, podsleuth.pp, policykit.pp, polipo.pp, portmap.pp, portreserve.pp, postfix.pp, postgresql.pp, postgrey.pp, ppp.pp, prelink.pp, prelude.pp, privoxy.pp, procmail.pp, prosody.pp, psad.pp, ptchown.pp, publicfile.pp, pulseaudio.pp, puppet.pp, pwauth.pp, qmail.pp, qpid.pp, quantum.pp, quota.pp, rabbitmq.pp, radius.pp, radvd.pp, raid.pp, rasdaemon.pp, rdisc.pp, readahead.pp, realmd.pp, redis.pp, remotelogin.pp, rhcs.pp, rhev.pp, rhgb.pp, rhnsd.pp, rhsmcertd.pp, ricci.pp, rkhunter.pp, rlogin.pp, rngd.pp, roundup.pp, rpc.pp, rpcbind.pp, rpm.pp, rshd.pp, rssh.pp, rsync.pp, rtas.pp, rtkit.pp, rwho.pp, samba.pp, sambagui.pp, sandbox.pp, sandboxX.pp, sanlock.pp, sasl.pp, sblim.pp, screen.pp, secadm.pp, sectoolm.pp, selinuxutil.pp, sendmail.pp, sensord.pp, setrans.pp, setroubleshoot.pp, seunshare.pp, sge.pp, shorewall.pp, slocate.pp, slpd.pp, smartmon.pp, smokeping.pp, smoltclient.pp, smsd.pp, snapper.pp, snmp.pp, snort.pp, sosreport.pp, soundserver.pp, spamassassin.pp, speech-dispatcher.pp, squid.pp, ssh.pp, sssd.pp, staff.pp, stapserver.pp, stunnel.pp, su.pp, sudo.pp, svnse
sysstat.pp, systemd.pp, tcpd.pp, tcsd.pp, telepathy.pp, telnet.pp, tftp.pp, tgtd.pp, thi
tvtime.pp, udev.pp, ulogd.pp, uml.pp, unconfined.pp, unconfineduser.pp, unlabeledr
userdomain.pp, userhelper.pp, usermanage.pp, usernetctl.pp,uucp.pp, uuidd.pp, va
vmtools.pp, vmware.pp, vnstatd.pp, vpn.pp, w3c.pp, watchdog.pp, wdmd.pp, webac
xguest.pp,xserver.pp, zabbix.pp, zarafa.pp, zebra.pp, zoneminder.pp, zosremote.pp

自分でーからポリシーを書かなければならないケースは多くない

Minimum Policy

- Fedora10から追加されたポリシータイプ
  - Targeted policyからポリシーモジュールの数を絞り、メモリ使用量を圧縮

- 組み込みやコンテナ、クラウド向けOSなど、メモリ消費量を抑えたいときに使う
  - Targeted Policyが有効なLinuxカーネルのメモリ使用量は、SELinuxが無効であるカーネルより30MB程度、固定で増加している
  - ポリシーのバイナリサイズはTargetedの3.5Mと比べて2.0M程度と半分程度

Multi Level Security Policy (MLS)

- MLSポリシーはMLSアクセス制御モデルを強制する
- 完全に軍事組織向け
- 今のところ、MLSでX window環境は利用不可（ポリシー整備が追いついていない）

```
# ls –laZ
--snip--
lrwxrwxrwx. root root system_u:object_r:bin_t:s0       sbin -> usr/sbin
drwxr-xr-x. root root system_u:object_r:var_t:s0       srv
dr-xr-xr-x. root root system_u:object_r:sysfs_t:s0     sys
drwxrwxrwt. root root system_u:object_r:tmp_t:s0       tmp
drwxr-xr-x. root root system_u:object_r:usr_t:s0       usr
drwxr-xr-x. root root system_u:object_r:var_t:s0       var
```

Multi Category Security Policy (MCS)

- TEでカバーしにくいドキュメントファイルなどのアクセス制御を使うことを目的としている
- SELinux-sandboxやSELinuxコンテナ、仮想化利用は、MCSを応用することで実現している
- FedoraCore6からデフォルトで有効
  - ただしあらゆるユーザーにカテゴリ0から1023までのすべてのカテゴリに属するリソースにアクセス可能なクリアランスが設定されている
  - リソースに対してはデフォルトでカテゴリなし

```
# id –Z
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

SELinuxセキュリティポリシーのまとめ

- 最新のFedoraやRHELでは、Targetedポリシーがデフォルト
  - Web経由の権限昇格を防ぐことに重点を置いている
- MCSポリシーは手頃なリソース隔離を実現する
  - sVirt, SELinux-sandboxなどに応用されている

SELinux を無効にしない運用 (Targeted+MCS)

- まず、ファイルのSELinuxコンテキスト変更で対応することを考える
  - Semanage fcontext –lで既存のSELinuxコンテキストを確認
  - もし実行時にアクセス違反が出た場合、Sealertコマンドを使ってその違反ログから適切なタイプを推薦させる

- SELinuxコンテキストが変更できない場合、ポリシーモジュールを作成する
  - Audit2allowコマンドを使う
    - ただし生成したポリシーは必ず確認する
    - 意図しない許可を与えてしまうこともありうる
  - ポリシーを直接記述
    - 難しい

Quick Reference

- Sestatus
  - SELinuxの状態を確認する
- Semanage module/user/login/fcontext –l
  - システムで有効になっている、各種SELinux設定を確認できる
- sealert -a /var/log/audit/audit.log
  - アクセス違反ログから解決方法を推薦させる

Quick Reference(2)

- Chcon –t hogehoge_t /var/www/hogehoge/index.html
  - 一時的にSELinuxコンテキストを変更する
- Semanage fcontext –a –t hogehoge_t “/var/www/hogehoge(/.*?)”
  - 永続的なSELinuxコンテキスト設定ルールの変更
- Restorecon –rv /var/www
  - SELinuxコンテキスト設定ルールの適用

基礎編まとめ

- SELinuxで利用可能なアクセス制御モデルは、一つ一つはシンプル
- 実際のセキュリティポリシーは難解極まる
- 主にSELinuxを推進しているRed Hat社も、使いにくさを改善する方向に投資を続けている
  - デフォルトポリシーの充実
  - ポリシーのモジュール化
  - 違反ログからポリシーモジュールを自動生成

今後の予定

- アプリケーション応用編（予定）
  - X window system, systemd, d-bus
  - SELinux-sandbox
  - Xguest
- コンテナ・仮想化応用編（予定）
  - sVirt, securecontainer
- Android編（予定）
  - SE for Android

Appendix: SELinux history

- 2003
  - Merged Linux kernel 2.6
- 2004
  - Enabling SELinux default on FC2(targeted)
- 2006
  - Policy reconstruction with reference policy (semanage, policy module)
  - Full labeled networking support
  - setroubleshot developed by Red Hat
  - MCS debut on FC5
- 2007
  - Xguest developed by Dan Walsh
  - SE-PostgreSQL developed by Kohei Kaigai
  - Strict policy sunked on Fedora 8 (merged targeted policy)
- 2009
  - sVirt presented by Red Hat
  - SELinux Sandbox developed by Dan Walsh
- 2012
  - SE-Android developed by NSA
- 2014
  - Enforcing SELinux on Android 4.4

参考文献

- “The Flask Security Architecture: System Support for Diverse Security Policies” http://www.nsa.gov/research/_files/publications/flask.pdf
- “SELinux Targeted vs Strict policy History and Strategy” http://selinuxsymposium.org/2005/presentations/session4/4-1-walsh.pdf
- SELinux/Tutorials/How is the policy provided and loaded http://wiki.gentoo.org/wiki/SELinux/Tutorials/How_is_the_policy_provided_and_loaded
- NB PolicyType http://selinuxproject.org/page/NB_PolicyType#Policy_Versions_Monolithic
- Getting Started with Reference Policy http://oss.tresys.com/projects/refpolicy/wiki/GettingStarted
- Using SELinux on RHEL 6 http://www.redhat.com/summit/2012/pdf/2012-DevDay-Lab-SELinux-Hacker.pdf
- Introducing the SELinux Sandbox http://danwalsh.livejournal.com/28545.html
- Fedora19 Security Guide - Fedora Documentation http://docs.fedoraproject.org/en-US/Fedora/19/html/Security_Guide/ch09.html
- Fedora Core 5の新機能 MCS (Multi Category Security) http://www.secureos.jp/index.php?plugin=attach&refer=events&openfile=20060531_lwe2006_KaiGai.pdf

Contact Information

E-Mail ：research—feedback@ffri.jp
Twitter：@FFRI_Research