Open Platform Trust Services (OpenPTS)
User's Guide

Version 0.2.4

Seiji Munetoh

May 6, 2011

Mailing list for comments: openpts-users@lists.sourceforge.jp
Web access (preferred): http://sourceforge.jp/projects/openpts

Contents

1 Introduction

1.1 Purpose

The purpose of this User's Guide is to provide a description of the usage of Open Platform Trust Services (OpenPTS).

1.2 Scope

System administrator and developer of Trusted Platform.

1.3 Architecture

Figure 1 shows brief overview of OpenPTS architecture. OpenPTS is used by both collector (target platform) and verifier sides. Collector side, 'ptsc' command manages the integrity of target platform. Verifier side, 'openpts' command is used to validate the target platform by remote attestation. The protocol between ptsc and openpts is based on TCG IF-M protocol. OpenPTS uses SSH between collector and verifier to secure the remote attestation. This figure shows stand-alone operation mode. OpenPTS supports IMC and IMV interfaces for TNC (Trusted Network Connect).

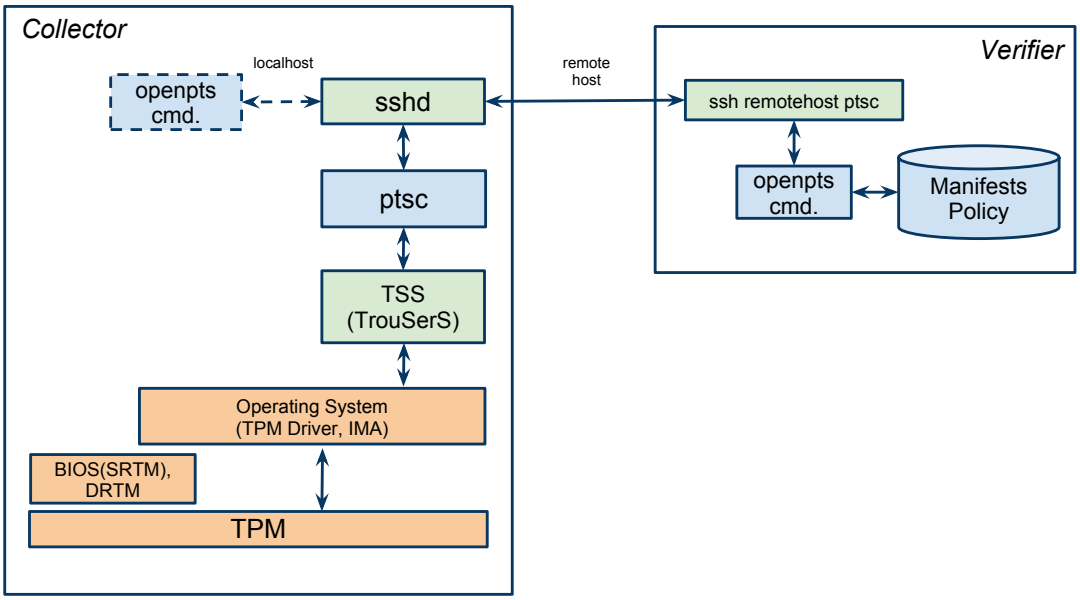

Figure 1: OpenPTS - Architecture (Standalone Mode)

1.4 Operations

Figure 2 shows how OpenPTS manage the integrity. OpenPTS uses a model which describe the behavior of transitive trust chain of target platform. The model is Finite State Machine (FSM) written by UML state diagram. OpenPTS uses this model to parse the integrity measurement log (IML) and generate the reference manifest (RM).

The behavior model just describe the general behavior of transitive trust chain and is used to generate RM and integrity report (IR). OpenPTS supports generic model of x86(PC) platform. The binary model contains actual digest value of target and used to validate the IML.

By using the model, we can translate the binary measurement (hash value) into security properties. Therefore we can use a policy to validate the property. This provides a flexible management of target platform. Finally, we get the validation result, VALID/INVALID/UNKNOWN.

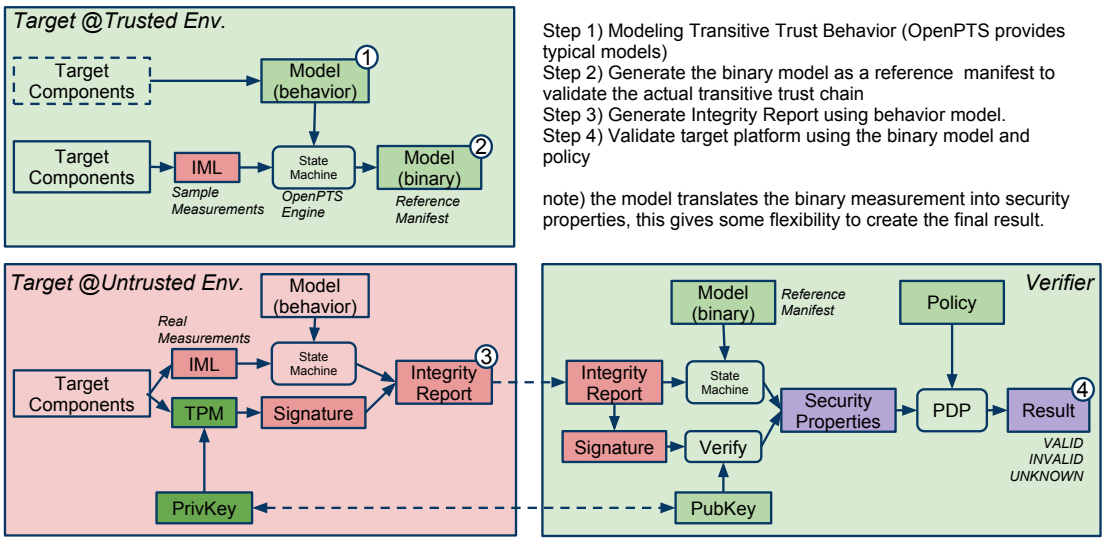

Figure 2: OpenPTS - Integrity Management Flow

1.5 Limitation

- AIDE and TNC integration is still under development.
- Need to apply the patch to TrouSerS (TSS) to handle eventlog properly.
- There is no interoperability of standalone IF-M mode between version 0.2.3 and 0.2.4 since we had changed this operation from version 0.2.4. Version 0.2.3 used ptscd daemon and SSH tunnel. This was deprecated since the management of SSH tunnel did not scale. Now we use simple SSH remote command execution. The IF-M go through the pipe between collector(ptsc command) and verifier(openpts command) protected by SSH.

2 Use case 1. Standalone Remote Attestation

In this use case, We use individual reference manifest and integrity database for each target platform. Thus, the reference manifest and integrity database are created by collector running at the target platform. Fig 3 shows the operation flow of OpenPTS.

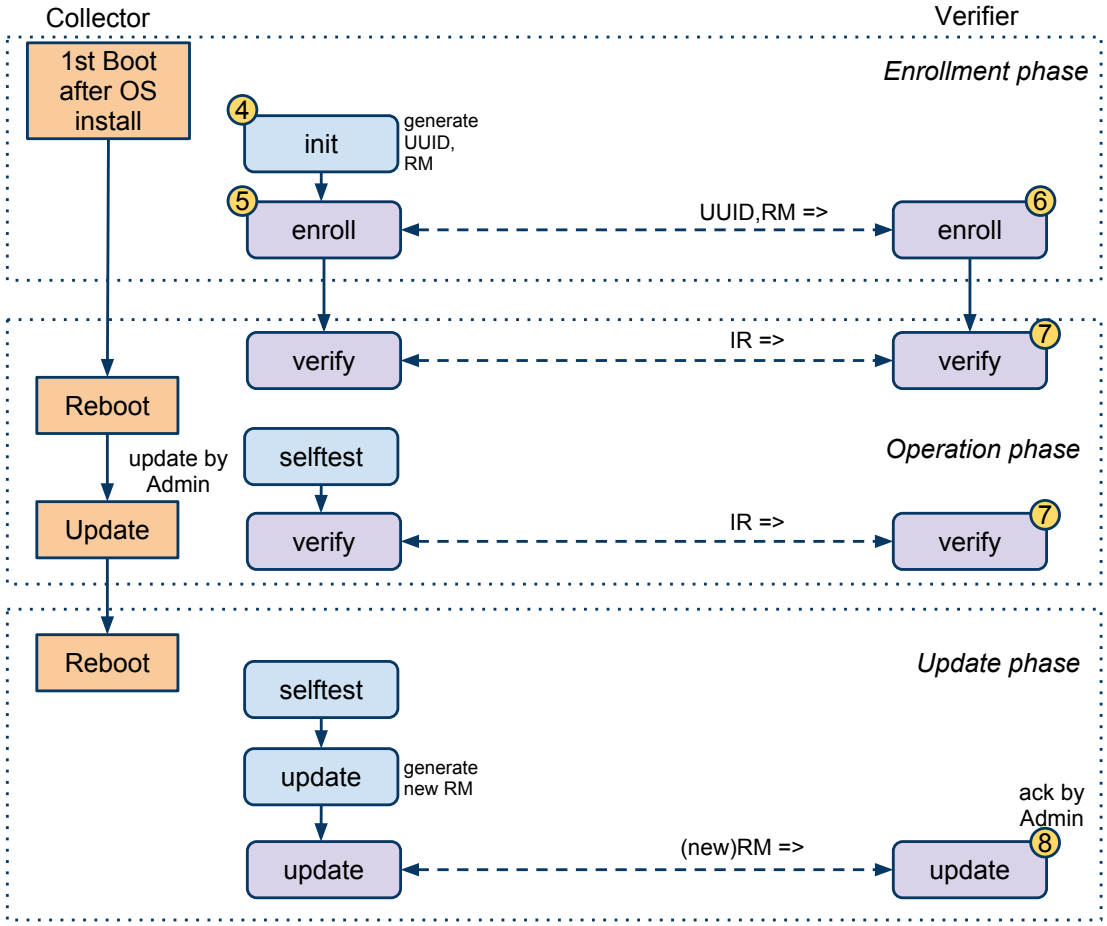

Figure 3: OpenPTS - Operation Flow

This use case have three operation phases as follows.

Enrollment phase We trust an installation process[1]. The collector generate the new UUID to identify the target and reference manifest based on the measurement of initial boot. Thus, the reference manifests are based on actual BIOS [2] and Operating System measurement at this phase. Verifier get the UUID and manifests from the Collector and securely stored them.

Operation phase Verifier validate the target (remote attestation).

Update phase After the BIOS or OS update, manifest must be updated. The OpenPTS collector do selftest at the startup run (ptsc -s). If validation was failed due to the change, it generates the new manifest.

If the update was expected, Verifier update the manifest too.

[1] If we have the EK credential of TPM, we can trust the remote platform

[2] OpenPTS generate manifest of actual measurement since there are no PC and BIOS vendors which disclose integrity information.

Enrollment Initial setup (Trusted environment)

Operation Status check by Remote Attestation (Untrusted environment)

Update Update the SW status (Trusted environment)

2.1 Setup the Collector (target platform)

2.1.1 Take the TPM ownerdhip

Take ownership of your TPM with well known secret.

```
# tpm_takeownership -y -z
```

2.1.2 Install openpts

(see the section 7, how to build)

```
# rpm -ivh openpts-0.2.4-1.x86_64.rpm
```

2.1.3 Configure ptsc

After the installation, adjust the configuration file '/etc/ptsc.conf' If you are using GRUB-IMA, assign the validation models to PCR[4,5,8].

```
rm.num=2
rm.model.1.pcr.4=grub_pcr4hdd.uml
rm.model.1.pcr.5=grub_pcr5.uml
rm.model.1.pcr.8=grub_pcr8.uml
```

If you enabled Linux-IMA, assign the validation model of IMA to PCR[10].

```
rm.model.1.pcr.10=f12_ima_pcr10.uml
```

Set the platform infomation. e.g.

```
platform.system.manufacturer=LENOVO
platform.system.productname=745749J
platform.system.version=ThinkPad X200
platform.bios.version=6DET58WW
```

The platform infomation stored in SMBIOS can be checked by 'dmidecode' command.

2.1.4 Setup the AIDE database (OPTIONAL)

Create the sample AIDE DB from current IML. (It takes long time).

```
# iml2aide -c /etc/ptsc.conf -o /var/lib/aide/aide.db.gz
```

Or create the AIDE DB. (It takes long time too).

```
# cp /usr/share/openpts/aide.conf  /etc/aide.conf
# aide -i
```

2.1.5 Initialize Collector ptsc

e.g.

```
# /usr/sbin/ptsc -i
Generate uuid                : 186bebba-2781-11e0-bcdb-001f160c9c28
Sign key  location           : SYSTEM
Generate UUID (for RM)       : 19566e16-2780-11e0-bf2e-001f160c9c28
level 0 Reference Manifest   : /var/lib/openpts//19566e16-...9c28/rm0.xml
level 1 Reference Manifest   : /var/lib/openpts//19566e16-...9c28/rm1.xml
```

Selftest the target platform.

```
# /usr/sbin/ptsc -t
selftest - OK
```

2.1.6 Startup ptsc

```
# service ptsc start
Starting ptsc:                                             [  OK  ]
```

Also, set whether ptsc should run on startup.

```
# chkconfig --add ptsc
```

Setup of the PTS collector is done.

2.2 Setup the Verifier

Install openpts to the localhost (or any remote verifier box).

2.2.1 Install openpts

Use the same package, it contains both collector and verifier.

```
# rpm -ivh openpts-0.2.4-1.x86_64.rpm
```

2.2.2 Setup SSH pub-key authentication

You have to setup SSH public key authentication between collector and verifier. The following example uses foo@localhost as the target (collector). e.g.

```
$ ssh-keygen -t rsa
$ ssh-copy-id foo@localhost
```

2.2.3 Enrollment with Collector

First, you enroll the collector. e.g.

```
$ openpts -i localhost
/usr/bin/openpts -i -l foo localhost
/home/foo/.openpts is missing. create [Y/n]:Y
Target             : localhost
Collector UUID     : 186bebba-2781-11e0-bcdb-001f160c9c28
Manifest UUID      : 19566e16-2780-11e0-bf2e-001f160c9c28
manifest[0]        : /home/foo/.openpts/186bebba-...9c28//19566e16-...9c28/rm0.xml
manifest[1]        : /home/foo/.openpts/186bebba-...c9c28//19566e16-...c9c28/rm1.xml
configuration      : /home/foo/.openpts/186bebba-...9c28/target.conf
validation policy  : /home/foo/.openpts/186bebba-...9c28/policy.conf
```

target.conf, policy.conf (and aide.ignore) are automaticaly generated. rm0.xml, rm1.xml and aide.db.gz are received from collector. To override existing setting, use ”-f” option.

```
$ openpts -i -f localhost
```

See the Table 2 about the file used by openpts command.

2.2.4 Remote Attestation

```
$ openpts -v localhost
Target          : localhost
Collector UUID : 186bebba-2781-11e0-bcdb-001f160c9c28
Manifest UUID  : 19566e16-2780-11e0-bf2e-001f160c9c28
username(ssh)   : default
port(ssh)       : default
policy file     : /home/foo/.openpts/186bebba-...9c28/policy.conf
property file   : /home/foo/.openpts/186bebba-...9c28/vr.properties
integrity       : valid
```

2.3 Collector update the manifests

At the startup, the collector selftest the platform. If the seftest was faild, the collector generate the new manifest against current measurements. This happen if you update any relevent components, such as the BIOS or OS image.

2.4 Verifier update the manifests

2.4.1 Accept the change happen on the collector

When the collector update the manifest, verification of the target was fail since there are mismatch between the collector and the verifier.

```
$ openpts -v localhost
Target          : localhost
Collector UUID : 1dbac28e-2787-11e0-b84a-001f160c9c28
Manifest UUID  : 1df210fe-2787-11e0-b84a-001f160c9c28
port            : 6678 (localhost)
policy file     : /home/foo/.openpts/1dbac28e-2787-11e0-b84a-001f160c9c28/policy.conf
property file   : /home/foo/.openpts/1dbac28e-2787-11e0-b84a-001f160c9c28/vr.properties
integrity       : unknown (INTERNAL ERROR) rc=35
Reasons
    0 Missing Reference Manifest(RM)
    1 Collector hostname = localhost
    2 Collector UUID = 1dbac28e-2787-11e0-b84a-001f160c9c28
    3 Collector RM UUID = 33b88c38-2787-11e0-adc0-001f160c9c28
New reference manifest exist. if this is expected change, update the manifest by openpts -i -f
```

If this is predicted or legitimate change. Update the target information.

```
$ openpts -i -f  localhost
Target             : localhost
Collector UUID     : 1dbac28e-2787-11e0-b84a-001f160c9c28
Manifest UUID      : 33b88c38-2787-11e0-adc0-001f160c9c28
manifest[0]        : /home/foo/.openpts/1dbac28e-...c9c28//33b88c38-...9c28/rm0.xml
manifest[1]        : /home/foo/.openpts/1dbac28e-...9c28//33b88c38-...9c28/rm1.xml
configuration      : /home/foo/.openpts/1dbac28e-...9c28/target.conf
validation policy  : /home/foo/.openpts/1dbac28e-...9c28/policy.conf
```

Then verify again.

```
$ openpts -v localhost
Target          : localhost
Collector UUID : 1dbac28e-2787-11e0-b84a-001f160c9c28
Manifest UUID  : 33b88c38-2787-11e0-adc0-001f160c9c28
```

```
username(ssh)   : default
port(ssh)       : default
policy file     : /home/foo/.openpts/1dbac28e-...9c28/policy.conf
property file   : /home/foo/.openpts/1dbac28e-...9c28/vr.properties
integrity       : valid
```

2.5 Check the status

2.5.1 Status of the collector

```
# /usr/sbin/ptsc -D
openpts version 0.2.2.svn

config file                 : /etc/ptsc.conf
UUID                        : 186bebba-...c9c28 (/var/lib/openpts/uuid)
IML access mode             : TSS
  Runtime IML type          : IMA (kernel 2.6.32)
RM UUID (current)           : 19566e16-2780-11e0-bf2e-001f160c9c28
RM UUID (for next boot)     : (null)
List of RM set              : 1 RM set in config dir
                               ID  UUID                  date(UTC)          status
                              ------------------------------------------------------------
                                0 d5086d88-...c9c28 2011-01-24-05:50:21 state=UNKNOWN
                              ------------------------------------------------------------
Integrity Report            : /var/lib/openpts/ir.xml
Model dir                   : /usr/share/openpts/models
                              Behavior Models
                              PCR lv  FSM files
                              ------------------------------------------------------------
                               0  0  /usr/share/openpts/models/bios_pcr0.uml
                               1  0  /usr/share/openpts/models/bios_pcr1.uml
                               2  0  /usr/share/openpts/models/bios_pcr2.uml
                               3  0  /usr/share/openpts/models/bios_pcr3.uml
                               4  0  /usr/share/openpts/models/bios_pcr4.uml
                               4  1  /usr/share/openpts/models/grub_pcr4hdd.uml
                               5  0  /usr/share/openpts/models/bios_pcr5.uml
                               5  1  /usr/share/openpts/models/grub_pcr5.uml
                               6  0  /usr/share/openpts/models/bios_pcr6.uml
                               7  0  /usr/share/openpts/models/bios_pcr7.uml
                               8  1  /usr/share/openpts/models/grub_pcr8.uml
                              10  1  /usr/share/openpts/models/f12_ima_pcr10.uml
                              ------------------------------------------------------------
```

2.5.2 Status of the verifier

```
$ openpts -D
Show openpts config
------------------------------------------------------------------------
config file                 : /home/foo/.openpts/openpts.conf
uuid                        : 240fbb66-6fdb-11e0-a735-001f160c9c28
------------------------------------------------------------------------
target[0] uuid              : cf9cfa32-6fba-11e0-84de-001f160c9c28
target[0] config            : /home/foo/.openpts/cf9cfa32-6fba-11e0-84de-001f160c9c28/target.conf
target[0] hostname          : localhost
------------------------------------------------------------------------
target[1] uuid              : 0445a354-6fdb-11e0-aa91-001f160c9c28
target[1] config            : /home/foo/.openpts/0445a354-6fdb-11e0-aa91-001f160c9c28/target.conf
target[1] hostname          : host023
------------------------------------------------------------------------
```

3 OpenPTS Commands Usage

3.1 ptsc

PTS collector.

```
Usage: ptsc [options] [command]

Commands: (forgrand)
  -i                    Initialize PTS collector
  -t                    Self test (attestation)
  -s                    Startup (selftest + timestamp)
  -u                    Update the RM
  -U                    Update the RM (auto)
  -D                    Display the configuration
  -m                    IF-M mode

Miscellaneous:
  -h                    Show this help message
  -v                    Verbose mode. Multiple -v options increase the verbosity.

Options:
  -c configfile         Set configuration file. defalt is /etc/ptsc.conf
  -P name=value         Set properties.
  -R                    Remove RMs
  -z                    Use the SRK secret to all zeros (20 bytes of zeros)
```

3.2 openpts

PTS verifier.

```
Usage: openpts [options] {-i [-f]|[-v]|-D} <target>
       openpts -D

Commands:
  -i [-f]               Initialize [forcibly] the PTS verifier with the target(collector).
  [-v]                  Verify target(collector) integrity against know measure.
  -D                    Display the configuration (target/ALL)

Miscellaneous:
  -h                    Show this help message
  -V                    Verbose mode. Multiple -V options increase the verbosity.

Options:
  -u                    Selects 'yes' as the the default answer when an update is available [no]
  -l username           ssh username [ssh default]
  -p port               ssh port number [ssh default]
  -c configfile         Set configuration file [~/.openpts/openpts.conf]
```

3.3 uml2dot

Generate dot file from the UML State Diagram model.

```
Usage: uml2dot [options] umlfile

Options
   -o output            Set output file (default is stdout)

Example:
$ uml2dot -o pcr0.dot pcr0.uml
$ dot -Tpng pcr0.dot -o pcr0.png
$ eog pcr0.png
```

3.4 rm2dot

Generate dot file from Reference Manifest (RM). Select pcr index since the RM may contains multiple FSMs for each PCRs.

```
Usage: rm2dot [options] rmfile

Options
        -o output       set output file (default is stdout)
        -p pcrindex     set PCR index
        -l level        set snapshot level (0 or 1)

Example:
$ rm2dot -p 0 -o pcr0.dot rm.uml
$ dot -Tpng pcr0.dot -o pcr0.png
$ eog pcr0.png
```

3.5 iml2text

Dump the eventlog in text. It take out the eventlog from TSS or securityfs file directly.

```
Usage: iml2text [options]

Options:
  -i filename           Set binary eventlog file (at securityfs)
  -p pcr_index          Select pcr (TSS)
  -I mode               Select IMA's log format (Kernel 2.6.32:32)
  -V                    Verify
  -D                    DRTM
  -E                    Enable endian conversion (BE->LE or LE->BE)
  -h                    Show this help message


Example:
$ iml2text
 Idx PCR        Type     Digest                                     EventData
-------------------------------------------------------------------------------
   0   0 0x00000008 1dfce7dde0cf13cfff102b1eb01875f752d5090c [BIOS:EV...
   1   0 0x00000001 1c41801dd329198e50a3d98040230095693e49b3 [BIOS:EV...
   2   0 0x00000001 16fb111792cb98a3de12f3abd0406fc04c7e5fca [BIOS:EV...
   3   0 0x00000001 dd261ca7511a7daf9e16cb572318e8e5fbd22963 [BIOS:EV...
<snip>
```

3.6 iml2aide

Convert IML to AIDE database.

```
Usage: iml2aide [option]

Options:
  -c filename           Set config file
  -i filename           Set IMA IML file. default, get IML via TSS
  -r filename           Set AIDE DB file as reference of fullpathname
  -o filename           Set output file (AIDE DB format, gziped)
  -w filename           Set output file (Ignore name list, plain text format)
  -h                    Show this help message

Example:
$ src/iml2aide -c /etc/ptsc.conf -r /var/lib/aide/aide.db.new.gz \
  -o /tmp/aide.db.gz
  AIDE DB(ref) : 241826 entries (/var/lib/aide/aide.db.new.gz)
  IML          : 5681 events (TSS)
  AIDE DB      : 3986 entries (tests/data/Fedora12/aide.db.gz)
```

3.7 ir2text

Convert Integrity Report (IR) to text format or binary format (=IML).

```
Usage: ir2text [options]

Options:
  -i filename           Set IR file
  -o filename           Set output file, else stdout
  -P filename           Set PCR output file (option)
  -b                    Binary, (Convert IR to IML)
  -E                    Enable endian conversion (BE->LE or LE->BE)
  -h                    Show this help message
```

3.8 tboot2iml

Convert the tboot messages to IML.

```
Usage: tboot2iml [options]

Options:
  -i filename           txt-stat file to read (default is STDIN)
  -g filename           grub.conf file to read (OPTION)
  -p path               grub path (OPTION)
  -o filename           Output to file (default is STDOUT)
  -v                    Verbose message
  -h                    Help

Example:
  # txt-stat >  txt-stat
  # tboot2iml -i txt-stat >> binary_rtm_measurments
```

4 OpenPTS Configuration Files

4.1 Files

OpenPTS generates and uses many files as described below. Table 6 lists the files used by collector (ptsc command). Table 2 lists the files used by verifier (openpts command). The verifier store the target infomation at the user's home directory.

Table 1: Files - collector side, (ptsc command)

File	Description
/etc/ptsc.conf	configuration file of collector
/var/lib/openpts/uuid	uuid of this platform
/var/lib/openpts/rm_uuid	uuid of current manifest (=RM_UUID)
/var/lib/openpts/newrm_uuid	uuid of next boot-cycle manifest (=NEWRM_UUID) TBD
/var/lib/openpts/{RM_UUID}/rm0.xml	Reference Manifest (BIOS)
/var/lib/openpts/{RM_UUID}/rm1.xml	Reference Manifest (IPL and OS)
/var/lib/aide/aide.db.gz	AIDE database file
/tmp/.ptsc/openpts/{VERIFIER_UUID}_{IR_UUID}.xml	Integrity Reports of each attestation

Table 2: Files - verifier side (openpts command)

File	Description
HOME/.openpts/openpts.conf	configuration file of verifier
HOME/.openpts/{COLLECTOR_UUID}/target.conf	configuration file of each target
HOME/.openpts/{COLLECTOR_UUID}/policy.conf	validation policy
HOME/.openpts/{COLLECTOR_UUID}/ir.xml	Integrity Report (XML)
HOME/.openpts/{COLLECTOR_UUID}/vr.properties	target properties
HOME/.openpts/{COLLECTOR_UUID}/{RM_UUID}/rm0.xml	Reference Manifest (BIOS) (XML)
HOME/.openpts/{COLLECTOR_UUID}/{RM_UUID}/rm1.xml	Reference Manifest (IPL and OS) (XML)
HOME/.openpts/{COLLECTOR_UUID}/aide.db.gz	AIDE database as Integrity Database
HOME/.openpts/{COLLECTOR_UUID}/aide.ignore	list of valid components not listed on AIDE database

4.2 /etc/ptsc.conf

Table 3: /etc/ptsc.conf

Name	Value	Description
config.dir	/var/lib/openpts	Set location of ptsc data
srk.password.mode	known	SRK password is well known secret (20 bytes of zeros)
	null	SRK password is null, SHA1("")
iml.mode	tss	Get IML via TSS
	securityfs	Get IML from securityfs filesystem
runtime.iml.type	IMA32	kernel 2.6.32
rm.num	1	Number of manifest. 1: Platform only 2: Platform and Runtime
rm.basedir	/var/lib/openpts/	Dir for Manifests
ir.dir	/tmp/.ptsc	Dir for Integrity Reports
uuid.file	/var/lib/openpts/uuid	UUID of Collector
rm.uuid.file	/var/lib/openpts/rm_uuid	UUID of Manifest
newrm.uuid.file	/var/lib/openpts/newrm_uuid	UUID of new Manifest
model.dir	/usr/share/openpts/models	Dir of validation models
rm.model.0.pcr.0	bios_pcr0.uml	validation model of BIOS PCR[0], CRTM
rm.model.0.pcr.1	bios_pcr1.uml	validation model of BIOS PCR[1]
rm.model.0.pcr.2	bios_pcr2.uml	validation model of BIOS PCR[2], OPTION ROM
rm.model.0.pcr.3	bios_pcr3.uml	validation model of BIOS PCR[3]
rm.model.0.pcr.4	bios_pcr4.uml	validation model of BIOS PCR[4], IPL
rm.model.0.pcr.5	bios_pcr5.uml	validation model of BIOS PCR[5]
rm.model.0.pcr.6	bios_pcr6.uml	validation model of BIOS PCR[6]
rm.model.0.pcr.7	bios_pcr7.uml	validation model of BIOS PCR[7]
rm.model.1.pcr.4	grub_pcr4hdd.uml	validation model of GRUB PCR[4], IPL
rm.model.1.pcr.5	grub_pcr5.uml	validation model of GRUB PCR[5], IPL data
rm.model.1.pcr.8	grub_pcr8.uml	validation model of GRUB PCR[8], OS images
rm.model.1.pcr.10	ima_pcr10.uml	validation model of Linux-IMA
rm.model.1.pcr.11	openpts.uml	validation model of OpenPTS
platform.system.manufacturer		
platform.system.productname		
platform.system.version		
platform.bios.version		
runtime.vendor.name	redhat	
runtime.distro.name	rhel	
runtime.distro.version	6	

4.3 /.openpts/openpts.conf

Table 4: /.openpts/openpts.conf

Name	Value	Description
uuid.file	./uuid	
verifier.logging.dir	./	

4.4 /.openpts/UUID/target.conf

Table 5: /.openpts/UUID/target.conf

Name	Value	Description
hostname	(hostname)	Target hostname
port	6678	Target port
ssh.mode	on	Use SSH tunnel
	off	direct access (localhost)
ssh.username	(foo)	SSH account name
ssh.port	(6680)	SSH tunneling port
target.uuid		UUID string
target.pubkey	(base64)	Publik Key
ima.validation.mode	none	
rm.num	1 or 2	Number of Manifest
rm.basedir	./	
rm.uuid.file	./rm_uuid	
newrm.uuid.file	./newrm_uuid	
oldrm.uuid.file	./oldrm_uuid	
ir.file	./ir.xml	
prop.file	./vr.properties	
policy.file	./policy.conf	
verifier.logging.dir	./	

5 Configulation of Trusted Platform

Unfortunately, There is no Linux distribution which configure the Trusted Platform well.

Table 6: Linux distribution and TC support

OS	Kernel	CONFIG_IMA	IPL	SRTM	DRTM
Fedora 12	2.6.32	Yes	Grub-0.97	patch	NA
Fedora 13	2.6.34	Yes	Grub-0.97	patch	NA
Fedora 14	2.6.35	Yes			tboot
Fedora 15	2.6.38	Yes			tboot
RHEL 6.0	2.6.32	Yes	Grub-0.97		NA
Ubuntu 10.04 LTS	2.6.32	No	Grub2	NA	NA
Ubuntu 10.10	2.6.35	No	Grub2	NA	OK

5.1 RHEL 6.0 - SRTM

SRTM based Trusted Boot (BIOS, no UEFI) and IMA could be enabled.

5.1.1 GRUB-IMA

Download source code "grub-0.97-68.el6.src.rpm" and patch.

```
$ su -c 'yum install ncurses-devel ncurses-static gnu-efi glibc-static \
  glibc-devel-2.12-1.7.el6_0.3.i686 glibc-static-2.12-1.7.el6_0.3.i686'
$ rpm -Uvh grub-0.97-68.el6.src.rpm
$ cd ~/rpmbuild/SOURCES
$ wget http://osdn.dl.sourceforge.jp/openpts/40294/grub-0.97-68.el6.ima-1.1.0.0.patch
$ cd ~/rpmbuild/SPECS
```

Modify grub.spec file as follows.

```
Release: 68%{?dist}.ima
<snip>
Patch32: grub-0.97-68.el6.ima-1.1.0.0.patch
<snip>
%patch32 -p1
<snip>
%configure --sbindir=/sbin --disable-auto-linux-mem-opt \
--enable-ima --datarootdir=%{_datadir}
```

Build the RPM and intall.

```
$ rpmbuild -ba grub.spec
$ su -c 'rpm -ivh ../RPMS/x86_64/grub-0.97-68.el6.ima.x86_64.rpm'
$ su -c 'grub-install /dev/sda'
$ grep TCG /boot/grub/*
Binary file /boot/grub/stage1 matches
Binary file /boot/grub/stage2 matches
```

5.1.2 Linux IMA

Add option "ima=on" at the kernel line in /boot/grub/grub.conf file. If you have Intel TPM (Thinkpad X200, T400 etc), you also need additional options. Add tpm_tis.itpm=1 tpm_tis.force=1 tpm_tis.interrupts=0 ima=on at the kernel line Set SELinux to permissive mode. System-¿Admin-¿SELinux management if you don't have /sys/kernel/security/ direcroty, please add following line to /etc/fstab

```
    * securityfs /sys/kernel/security securityfs rw 0 0
```

5.1.3 TrouSetS(TSS)

TrouSerS is provided by RedHat. However, you have to use the latest TrouSerS since trousers-0.3.4-4.el6.x86_64 do not support the eventlog created by Linux-IMA.

```
$ git clone git://trousers.git.sourceforge.net/gitroot/trousers/trousers trousers-git
$ cd trousers-git
$ sh bootstrap.sh
$ ./configure
$ cd ..
$ ln -s trousers-git trousers-0.3.6git
$ tar zcvf ~/rpmbuild/SOURCES/trousers-0.3.6git.tar.gz ./trousers-0.3.6git/*
$ rpmbuild -bb trousers-0.3.6git/dist/trousers.spec

# rpm -ivh --force trousers-0.3.6git-1.x86_64.rpm
```

notes) You may need to fix the package dependancies in trousers.spec
Modify /etc/tcsd.conf file as follows

```
firmware_log_file = /sys/kernel/security/tpm0/binary_bios_measurements
kernel_log_file = /sys/kernel/security/ima/binary_runtime_measurements
firmware_pcrs = 0,1,2,3,4,5,6,7,8
kernel_pcrs = 10
```

notes) if you already taken the ownership and the system.data is missing. please copy the dummy system.data.

```
cp ./dist/dummy\_tss\_system.data /var/lib/tpm/system.data
```

Ok, enable tcsd daemon.

```
chkconfig tcsd on
service tcsd start
```

5.2 Fedora 12 - SRTM

SRTM based Trusted Boot and IMA could be enabled.

5.2.1 GRUB-IMA

Download source code and patch.

```
$ su -c 'yumdownloader --source grub'
$ su -c 'yum-builddep grub-0.97-62.fc12.src.rpm'
$ rpm -Uvh grub-0.97-62.fc12.src.rpm
$ cd ~/rpmbuild/SOURCES
$ wget http://osdn.dl.sourceforge.jp/openpts/40294/grub-0.97-62.fc12.ima-1.1.0.0.patch
$ cd ~/rpmbuild/SPECS
```

Modify grub.spec file as follows.

```
+Release: 62%{?dist}.ima
+Patch2: grub-0.97-62.fc12.ima-1.1.0.0.patch
+%patch2 -p1
+%configure --sbindir=/sbin --disable-auto-linux-mem-opt \
--enable-ima --datarootdir=%{_datadir}
```

Build the RPM and intall.

```
$ rpmbuild -ba grub.spec
$ su -c 'rpm -ivh ../RPMS/x86_64/grub-0.97-62.fc12.ima.x86_64.rpm'
$ su -c 'grub-install /dev/sda'
```

5.2.2 Linux IMA

Add option "ima_tcb=1" at the kernel line in /boot/grub/grub.conf file.

If you have Intel TPM (Thinkpad X200, T400 etc), you also need additional options.

Add tpm_tis.itpm=1 tpm_tis.force=1 tpm_tis.interrupts=0 ima_tcb=1 at the kernel line

Set SELinux to permissive mode. System-¿Admin-¿SELinux management

if you don't have /sys/kernel/security/ direcroty, please add following line to /etc/fstab

```
    * securityfs /sys/kernel/security securityfs rw 0 0
```

5.2.3 TrouSetS(TSS)

Modify /etc/tcsd.conf file as follows

```
firmware_log_file = /sys/kernel/security/tpm0/binary_bios_measurements
kernel_log_file = /sys/kernel/security/ima/binary_runtime_measurements
firmware_pcrs = 0,1,2,3,4,5,6,7,8
kernel_pcrs = 10
```

5.3 Fedora 15 - SRTM and DRTM

The target platform must support Intel TXT. This example uses Lenovo Thinkpad X200. Note that this is experimental support and configulation of the tboot may differ according to the hardware.

5.3.1 Configure tboot

tboot did not support well known secret against the ownership.

```
tpm_takeownership -z
```

```
# yum install tboot
```

Change /boot/grub/grub.conf

```
title Fedora (2.6.38.1-6.fc15.x86_64) tboot
        root (hd0,0)
        kernel /tboot.gz logging=serial,vga,memory vga_delay=5
        module /vmlinuz-2.6.38.1-6.fc15.x86_64 ro root=/dev/mapper/vg_munetohx200f15-lv_root \
        rd_LVM_LV=vg_munetohx200f15/lv_root rd_LVM_LV=vg_munetohx200f15/lv_swap rd_NO_LUKS \
        rd_NO_MD rd_NO_DM LANG=en_US.UTF-8 SYSFONT=latarcyrheb-sun16 KEYTABLE=us selinux=0 \
        rhgb quiet xdriver=vesa nomodeset 1
        module /initramfs-2.6.38.1-6.fc15.x86_64.img
        module /GM45_GS45_PM45_SINIT_21.BIN
```

Create LCP policy

```
# cd /root
# lcp_mlehash -c "logging=serial,vga,memory vga_delay=5" /boot/tboot.gz > mle_hash
# lcp_crtpol -t hashonly -m mle_hash -o lcp.pol
```

```
$ tb_polgen --create --type nonfatal vl.pol
```

```
$ tb_polgen --add --num 0 --pcr none --hash image --cmdline "ro \
  root=/dev/mapper/vg_munetohx200f15-lv_root rd_LVM_LV=vg_munetohx200f15/lv_root \
  rd_LVM_LV=vg_munetohx200f15/lv_swap rd_NO_LUKS rd_NO_MD rd_NO_DM \
  LANG=en_US.UTF-8 SYSFONT=latarcyrheb-sun16 KEYTABLE=us selinux=0 rhgb quiet \
  xdriver=vesa nomodeset 1" --image /boot/vmlinuz-2.6.38.1-6.fc15.x86_64 vl.pol
```

```
$ tb_polgen --add --num 1 --pcr 19 --hash image --cmdline "" --image \
  /boot/initramfs-2.6.38.1-6.fc15.x86_64.img vl.pol
```

```
$ tpmnv_defindex -i 0x20000002 -s 8 -pv 0 -rl 0x07 -wl 0x07 -p TPM-password
$ tpmnv_defindex -i owner -p TPM-password
$ tpmnv_defindex -i owner -s 34 -pv 0x02 -p TPM-password
$ tpmnv_defindex -i 0x20000001 -s 256 -pv 0x02 -p TPM-password
```

```
$ tpmnv_getcap
  The response data is:
  20 00 00 02 20 00 00 01 40 00 00 01 50 00 00 01
  50 00 00 02 10 00 00 01

  6 indices have been defined
  list of indices for defined NV storage areas:
  0x20000002 0x20000001 0x40000001 0x50000001 0x50000002 0x10000001
```

```
$ lcp_writepol -i owner -f lcp.pol -p TPM-password
  Successfully write policy into index 0x40000001
```

```
$ lcp_writepol -i 0x20000001 -f vl.pol -p TPM-password
  Successfully write policy into index 0x20000001
```

Reboot the system.

5.3.2 Configure openpts

Modify '/etc/tcsd.conf' to read this SRTM+DRTM IML

```
firmware_log_file = /var/lib/openpts/binary_rtm_measurements
firmware_pcrs = 0,1,2,3,4,5,6,7,8,17,18,19
```

Replace tcsd init script.

```
# cp -b /usr/share/openpts/tboot/tcsd /etc/init.d/tcsd
```

Start TSS daemon

```
service tcsd start
```

Confirm the eventlog that contains events at PCR[17] to PCR[18]

```
$ iml2text
```

Modify ptsc.conf

```
service tcsd start
```

Initialize ptsc

```
# ptsc -i
# ptsc -t
```

5.4 Ubuntu 10.04

SRTM based Trusted Boot covers BIOS only. You need to recompile the kernel to use the IMA.

6 Build OpenPTS

6.1 Linux RPM package

Install required packages to build.

```
# yum install libtool trousers-devel openssl-devel libxml2-devel libuuid-devel sqlite-devel
```

Build RPM pcakage od OpenPTS.

```
$ sh bootstrap.sh
$ ./configure
$ make rpmbuild-ba
$ rpm -qlp ~/rpmbuild/RPMS/x86\_64/openpts-0.2.4-1.x86\_64.rpm
/etc/ptsc.conf
/etc/rc.d/init.d/ptsc
/usr/bin/iml2aide
/usr/bin/iml2text
/usr/bin/openpts
/usr/bin/rm2dot
/usr/bin/tpm_createkey
/usr/bin/uml2dot
<snip>
```

6.2 Linux DEB package

Ubuntu does not support IMA.

```
$ sh bootstrap.sh
$ ./configure
$ make dpkg-buildpackage
$ dpkg-deb --contents ../openpts_0.2.4_i386.deb
<snip>
```

6.3 User's Guide

User's guide is written in Latex. Install the latex environments before generate the document.

```
# yum install texlive texlive-latex dvipdfmx
```

Generate PDF.

```
$ cd doc
$ make ug
$ evince userguide.pdf
```

6.4 Design document

```
$ cd models
$ make png
$ cd ..
$ cd doc
$ make hldd
$ evince desgin.pdf
```

6.5 API document

```
$ cd doc
$ make lldd
$ firefox apidoc_html/index.html
```

7 Common errors and problems

7.1 tpm_takeownership failed (0x0008)

```
Tspi_TPM_TakeOwnership failed: 0x00000008 - layer=tpm, code=0008 (8),
The TPM target command has been disabled
```

Your TPM already taken the ownership. If you don't know the owner password, you have to clear the TPM. To clear the TPM, Your PC needs cold boot, then enter the BIOS menu and clear the TPM.

7.2 Key generation failed

```
ERROR: Tspi_Context_LoadKeyByUUID (SRK) failed rc=0x2020
Your key storage of tcsd is damaged or missing.
```

Check the key storage file "/var/lib/tpm/system.data" If the size is zero, your install TSS after someone take the ownership. If you know the owner password. you can recover the storage file.

```
# cp /XXX/dummy_tss_system.data /var/lib/tpm/system.data
# service tcsd restart
```

7.3 validation failed - POLICY-L010

```
Reasons
    0 [POLICY-L010] tpm.quote.pcr.10 is Zjw44Y9jXCBbf8cRurxgzOwspQo=, not K2ruQ8H5ieZWI57wrUguMe6erPo=
```

PCR10 is changeed by IMA, comment out the policy file, ' /.openpts/{UUID}/policy.conf'

```
tpm.quote.pcr.8=rKefmpUQOPKHX6zoIQn+5vnpr0E=
# tpm.quote.pcr.10=K2ruQ8H5ieZWI57wrUguMe6erPo=
bios.pcr0.integrity=valid
```

7.4 TPM reports 0x803 Error

This is TPM_DEFEND_LOCK_RUNNING error. Your TPM is defending against dictionary attacks. And can be cleared by 'tpm_resetdalock' command with owner secret. However some TPM assert this flag without attack. the workaround is,

- take TPM owenweship with -y (known-secret) option.
- add 'tpm.resetdalock=on' in /etc/ptsc.conf