# dm-ima

For a given system, various external services/infrastructure tools (including the attestation service) interact with it - both during the setup and during rest of the system run-time. They share sensitive data and/or execute critical workload on that system. The external services may want to verify the current run-time state of the relevant kernel subsystems before fully trusting the system with business-critical data/workload.

Device mapper plays a critical role on a given system by providing various important functionalities to the block devices using various target types like crypt, verity, integrity etc. Each of these target typesâ€™ functionalities can be configured with various attributes. The attributes chosen to configure these target types can significantly impact the security profile of the block device, and in-turn, of the system itself. For instance, the type of encryption algorithm and the key size determines the strength of encryption for a given block device.

Therefore, verifying the current state of various block devices as well as their various target attributes is crucial for external services before fully trusting the system with business-critical data/workload.

IMA kernel subsystem provides the necessary functionality for device mapper to measure the state and configuration of various block devices -

- by device mapper itself, from within the kernel,
- in a tamper resistant way,
- and re-measured - triggered on state/configuration change.

## Setting the IMA Policy:

For IMA to measure the data on a given system, the IMA policy on the system needs to be updated to have following line, and the system needs to be restarted for the measurements to take effect.

```
/etc/ima/ima-policy
   measure func=CRITICAL_DATA label=device-mapper template=ima-buf
```

The measurements will be reflected in the IMA logs, which are located at:

```
/sys/kernel/security/integrity/ima/ascii_runtime_measurements
/sys/kernel/security/integrity/ima/binary_runtime_measurements
```

Then IMA ASCII measurement log has the following format:

```
<PCR> <TEMPLATE_DATA_DIGEST> <TEMPLATE_NAME> <TEMPLATE_DATA>

PCR := Platform Configuration Register, in which the values are registered.
      This is applicable if TPM chip is in use.

TEMPLATE_DATA_DIGEST := Template data digest of the IMA record.
TEMPLATE_NAME := Template name that registered the integrity value (e.g. ima-buf).

TEMPLATE_DATA := <ALG> ":" <EVENT_DIGEST> <EVENT_NAME> <EVENT_DATA>
                 It contains data for the specific event to be measured,
                 in a given template data format.

ALG := Algorithm to compute event digest
EVENT_DIGEST := Digest of the event data
EVENT_NAME := Description of the event (e.g. 'dm_table_load').
EVENT_DATA := The event data to be measured.
```

*NOTE #1:*
The DM target data measured by IMA subsystem can alternatively be queried from userspace by setting DM_IMA_MEASUREMENT_FLAG with DM_TABLE_STATUS_CMD.

*NOTE #2:*
The Kernel configuration CONFIG_IMA_DISABLE_HTABLE allows measurement of duplicate records.
To support recording duplicate IMA events in the IMA log, the Kernel needs to be configured with CONFIG_IMA_DISABLE_HTABLE=y.

## Supported Device States:

Following device state changes will trigger IMA measurements:

1. Table load
2. Device resume
3. Device remove
4. Table clear
5. Device rename

### 1. Table load:

When a new table is loaded in a device's inactive table slot, the device information and target specific details from the targets in the table are measured.

The IMA measurement log has the following format for 'dm_table_load':

```
EVENT_NAME := "dm_table_load"
EVENT_DATA := <dm_version_str> ";" <device_metadata> ";" <table_load_data>

dm_version_str := "dm_version=" <N> "." <N> "." <N>
                  Same as Device Mapper driver version.
device_metadata := <device_name> "," <device_uuid> "," <device_major> "," <device_minor> ","
                   <minor_count> "," <num_device_targets> ";"

device_name := "name=" <dm-device-name>
device_uuid := "uuid=" <dm-device-uuid>
device_major := "major=" <N>
device_minor := "minor=" <N>
```

```
minor_count := "minor_count=" <N>
num_device_targets := "num_targets=" <N>
dm-device-name := Name of the device. If it contains special characters like '\', ',', ';',
                  they are prefixed with '\'.
dm-device-uuid := UUID of the device. If it contains special characters like '\', ',', ';',
                  they are prefixed with '\'.

table_load_data := <target_data>
                   Represents the data (as name=value pairs) from various targets in the table,
                   which is being loaded into the DM device's inactive table slot.
target_data := <target_data_row> | <target_data><target_data_row>

target_data_row := <target_index> "," <target_begin> "," <target_len> "," <target_name> ","
                   <target_version> "," <target_attributes> ";"
target_index := "target_index=" <N>
                Represents nth target in the table (from 0 to N-1 targets specified in <num_device_targets>)
                If all the data for N targets doesn't fit in the given buffer - then the data that fits
                in the buffer (say from target 0 to x) is measured in a given IMA event.
                The remaining data from targets x+1 to N-1 is measured in the subsequent IMA events,
                with the same format as that of 'dm_table_load'
                i.e. <dm_version_str> ";" <device_metadata> ";" <table_load_data>.

target_begin := "target_begin=" <N>
target_len := "target_len=" <N>
target_name := Name of the target. 'linear', 'crypt', 'integrity' etc.
               The targets that are supported for IMA measurements are documented below in the
               'Supported targets' section.
target_version := "target_version=" <N> "." <N> "." <N>
target_attributes := Data containing comma separated list of name=value pairs of target specific attributes.

For instance, if a linear device is created with the following table entries,
 # dmsetup create linear1
 0 2 linear /dev/loop0 512
 2 2 linear /dev/loop0 512
 4 2 linear /dev/loop0 512
 6 2 linear /dev/loop0 512

Then IMA ASCII measurement log will have the following entry:
(converted from ASCII to text for readability)

10 a8c5ff755561c7a28146389d1514c318592af49a ima-buf sha256:4d73481ecce5eadba8ab084640d85bb9ca899af4d0a122989252a76efadc5
dm_table_load
dm_version=4.45.0;
name=linear1,uuid=,major=253,minor=0,minor_count=1,num_targets=4;
target_index=0,target_begin=0,target_len=2,target_name=linear,target_version=1.4.0,device_name=7:0,start=512;
target_index=1,target_begin=2,target_len=2,target_name=linear,target_version=1.4.0,device_name=7:0,start=512;
target_index=2,target_begin=4,target_len=2,target_name=linear,target_version=1.4.0,device_name=7:0,start=512;
target_index=3,target_begin=6,target_len=2,target_name=linear,target_version=1.4.0,device_name=7:0,start=512;
```

### 2. Device resume:

When a suspended device is resumed, the device information and the hash of the data from previous load of an active table are measured.

The IMA measurement log has the following format for 'dm_device_resume':

```
EVENT_NAME := "dm_device_resume"
EVENT_DATA := <dm_version_str> ";" <device_metadata> ";" <active_table_hash> ";" <current_device_capacity> ";"

dm_version_str := As described in the 'Table load' section above.
device_metadata := As described in the 'Table load' section above.
active_table_hash := "active_table_hash=" <table_hash_alg> ":" <table_hash>
                     Rerpresents the hash of the IMA data being measured for the
                     active table for the device.
table_hash_alg := Algorithm used to compute the hash.
table_hash := Hash of the (<dm_version_str> ";" <device_metadata> ";" <table_load_data> ";")
              as described in the 'dm_table_load' above.
              Note: If the table_load data spans across multiple IMA 'dm_table_load'
              events for a given device, the hash is computed combining all the event data
              i.e. (<dm_version_str> ";" <device_metadata> ";" <table_load_data> ";")
              across all those events.
current_device_capacity := "current_device_capacity=" <N>

For instance, if a linear device is resumed with the following command,
#dmsetup resume linear1

then IMA ASCII measurement log will have an entry with:
(converted from ASCII to text for readability)

10 56c00cc062ffc24ccd9ac2d67d194af3282b934e ima-buf sha256:e7d12c03b958b4e0e53e7363a06376be88d98a1ac191fdbd3baf5e4b77f32
dm_device_resume
dm_version=4.45.0;
name=linear1,uuid=,major=253,minor=0,minor_count=1,num_targets=4;
active_table_hash=sha256:4d73481ecce5eadba8ab084640d85bb9ca899af4d0a122989252a76efadc5b72;current_device_capacity=8;
```

### 3. Device remove:

When a device is removed, the device information and a sha256 hash of the data from an active and inactive table are measured.

The IMA measurement log has the following format for 'dm_device_remove':

```
EVENT_NAME := "dm_device_remove"
EVENT_DATA := <dm_version_str> ";" <device_active_metadata> ";" <device_inactive_metadata> ";"
              <active_table_hash> "," <inactive_table_hash> "," <remove_all> ";" <current_device_capacity> ";"

dm_version_str := As described in the 'Table load' section above.
device_active_metadata := Device metadata that reflects the currently loaded active table.
                          The format is same as 'device_metadata' described in the 'Table load' section above.
device_inactive_metadata := Device metadata that reflects the inactive table.
                            The format is same as 'device_metadata' described in the 'Table load' section above.
active_table_hash := Hash of the currently loaded active table.
                     The format is same as 'active_table_hash' described in the 'Device resume' section above.
inactive_table_hash :=  Hash of the inactive table.
                        The format is same as 'active_table_hash' described in the 'Device resume' section above.
remove_all := "remove_all=" <yes_no>
yes_no := "y" | "n"
```

```
current_device_capacity := "current_device_capacity=" <N>

For instance, if a linear device is removed with the following command,
 #dmsetup remove l1

then IMA ASCII measurement log will have the following entry:
(converted from ASCII to text for readability)

10 790e830a3a7a31590824ac0642b3b31c2d0e8b38 ima-buf sha256:ab9f3c959367a8f5d4403d6ce9c3627dadfa8f9f0e7ec7899299782388de3
dm_device_remove
dm_version=4.45.0;
device_active_metadata=name=l1,uuid=,major=253,minor=2,minor_count=1,num_targets=2;
device_inactive_metadata=name=l1,uuid=,major=253,minor=2,minor_count=1,num_targets=1;
active_table_hash=sha256:4a7e62efaebfc86af755831998b7db6f59b60d23c9534fb16a4455907957953a,
inactive_table_hash=sha256:9d79c175bc2302d55a183e8f50ad4bafd60f7692fd6249e5fd213e2464384b86,remove_all=n;
current_device_capacity=2048;
```

### 4. Table clear:

When an inactive table is cleared from the device, the device information and a sha256 hash of the data from an inactive table are measured.

The IMA measurement log has the following format for 'dm_table_clear':

```
EVENT_NAME := "dm_table_clear"
EVENT_DATA := <dm_version_str> ";" <device_inactive_metadata> ";" <inactive_table_hash> ";" <current_device_capacity> ";

dm_version_str := As described in the 'Table load' section above.
device_inactive_metadata := Device metadata that was captured during the load time inactive table being cleared.
                            The format is same as 'device_metadata' described in the 'Table load' section above.
inactive_table_hash := Hash of the inactive table being cleared from the device.
                       The format is same as 'active_table_hash' described in the 'Device resume' section above.
current_device_capacity := "current_device_capacity=" <N>

For instance, if a linear device's inactive table is cleared,
 #dmsetup clear l1

then IMA ASCII measurement log will have an entry with:
(converted from ASCII to text for readability)

10 77d347408f557f68f0041acb0072946bb2367fe5 ima-buf sha256:42f9ca22163fdfa548e6229dece2959bc5ce295c681644240035827ada0e1
dm_table_clear
dm_version=4.45.0;
name=l1,uuid=,major=253,minor=2,minor_count=1,num_targets=1;
inactive_table_hash=sha256:75c0dc347063bf474d28a9907037eba060bfe39d8847fc0646d75e149045d545;current_device_capacity=1024
```

### 5. Device rename:

When an device's NAME or UUID is changed, the device information and the new NAME and UUID are measured.

The IMA measurement log has the following format for 'dm_device_rename':

```
EVENT_NAME := "dm_device_rename"
EVENT_DATA := <dm_version_str> ";" <device_active_metadata> ";" <new_device_name> "," <new_device_uuid> ";" <current_dev

dm_version_str := As described in the 'Table load' section above.
device_active_metadata := Device metadata that reflects the currently loaded active table.
                          The format is same as 'device_metadata' described in the 'Table load' section above.
new_device_name := "new_name=" <dm-device-name>
dm-device-name := Same as <dm-device-name> described in 'Table load' section above
new_device_uuid := "new_uuid=" <dm-device-uuid>
dm-device-uuid := Same as <dm-device-uuid> described in 'Table load' section above
current_device_capacity := "current_device_capacity=" <N>

E.g 1: if a linear device's name is changed with the following command,
 #dmsetup rename linear1 --setuuid 1234-5678

then IMA ASCII measurement log will have an entry with:
(converted from ASCII to text for readability)

10 8b0423209b4c66ac1523f4c9848c9b51ee332f48 ima-buf sha256:6847b7258134189531db593e9230b257c84f04038b5a18fd2e1473860e056
dm_device_rename
dm_version=4.45.0;
name=linear1,uuid=,major=253,minor=2,minor_count=1,num_targets=1;new_name=linear1,new_uuid=1234-5678;
current_device_capacity=1024;

E.g 2:  if a linear device's name is changed with the following command,
 # dmsetup rename linear1 linear=2

then IMA ASCII measurement log will have an entry with:
(converted from ASCII to text for readability)

10 bef70476b99c2bdf7136fae033aa8627da1bf76f ima-buf sha256:8c6f9f53b9ef9dc8f92a2f2cca8910e622543d0f0d37d484870cb16b95111
dm_device_rename
dm_version=4.45.0;
name=linear1,uuid=1234-5678,major=253,minor=2,minor_count=1,num_targets=1;
new_name=linear\=2,new_uuid=1234-5678;
current_device_capacity=1024;
```

## Supported targets:

Following targets are supported to measure their data using IMA:

1. cache
2. crypt
3. integrity
4. linear
5. mirror
6. multipath
7. raid
8. snapshot
9. striped

10. verity

## 1. cache

The 'target_attributes' (described as part of EVENT_DATA in 'Table load' section above) has the following data format for 'cache' target.

```
target_attributes := <target_name> "," <target_version> "," <metadata_mode> "," <cache_metadata_device> ","
                     <cache_device> "," <cache_origin_device> "," <writethrough> "," <writeback> ","
                     <passthrough> "," <no_discard_passdown> ";"

target_name := "target_name=cache"
target_version := "target_version=" <N> "." <N> "." <N>
metadata_mode := "metadata_mode=" <cache_metadata_mode>
cache_metadata_mode := "fail" | "ro" | "rw"
cache_device := "cache_device=" <cache_device_name_string>
cache_origin_device := "cache_origin_device=" <cache_origin_device_string>
writethrough := "writethrough=" <yes_no>
writeback := "writeback=" <yes_no>
passthrough := "passthrough=" <yes_no>
no_discard_passdown := "no_discard_passdown=" <yes_no>
yes_no := "y" | "n"

E.g.
When a 'cache' target is loaded, then IMA ASCII measurement log will have an entry
similar to the following, depicting what 'cache' attributes are measured in EVENT_DATA
for 'dm_table_load' event.
(converted from ASCII to text for readability)

dm_version=4.45.0;name=cache1,uuid=cache_uuid,major=253,minor=2,minor_count=1,num_targets=1;
target_index=0,target_begin=0,target_len=28672,target_name=cache,target_version=2.2.0,metadata_mode=rw,
cache_metadata_device=253:4,cache_device=253:3,cache_origin_device=253:5,writethrough=y,writeback=n,
passthrough=n,metadata2=y,no_discard_passdown=n;
```

## 2. crypt

The 'target_attributes' (described as part of EVENT_DATA in 'Table load' section above) has the following data format for 'crypt' target.

```
target_attributes := <target_name> "," <target_version> "," <allow_discards> "," <same_cpu_crypt> ","
                     <submit_from_crypt_cpus> "," <no_read_workqueue> "," <no_write_workqueue> ","
                     <iv_large_sectors> "," <iv_large_sectors> "," [<integrity_tag_size> ","] [<cipher_auth> ","]
                     [<sector_size> ","] [<cipher_string> ","] <key_size> "," <key_parts> ","
                     <key_extra_size> "," <key_mac_size> ";"

target_name := "target_name=crypt"
target_version := "target_version=" <N> "." <N> "." <N>
allow_discards := "allow_discards=" <yes_no>
same_cpu_crypt := "same_cpu_crypt=" <yes_no>
submit_from_crypt_cpus := "submit_from_crypt_cpus=" <yes_no>
no_read_workqueue := "no_read_workqueue=" <yes_no>
no_write_workqueue := "no_write_workqueue=" <yes_no>
iv_large_sectors := "iv_large_sectors=" <yes_no>
integrity_tag_size := "integrity_tag_size=" <N>
cipher_auth := "cipher_auth=" <string>
sector_size := "sector_size="  <N>
cipher_string := "cipher_string="
key_size := "key_size="  <N>
key_parts := "key_parts="  <N>
key_extra_size := "key_extra_size="  <N>
key_mac_size := "key_mac_size="  <N>
yes_no := "y" | "n"

E.g.
When a 'crypt' target is loaded, then IMA ASCII measurement log will have an entry
similar to the following, depicting what 'crypt' attributes are measured in EVENT_DATA
for 'dm_table_load' event.
(converted from ASCII to text for readability)

dm_version=4.45.0;
name=crypt1,uuid=crypt_uuid1,major=253,minor=0,minor_count=1,num_targets=1;
target_index=0,target_begin=0,target_len=1953125,target_name=crypt,target_version=1.23.0,
allow_discards=y,same_cpu=n,submit_from_crypt_cpus=n,no_read_workqueue=n,no_write_workqueue=n,
iv_large_sectors=n,cipher_string=aes-xts-plain64,key_size=32,key_parts=1,key_extra_size=0,key_mac_size=0;
```

## 3. integrity

The 'target_attributes' (described as part of EVENT_DATA in 'Table load' section above) has the following data format for 'integrity' target.

```
target_attributes := <target_name> "," <target_version> "," <dev_name> "," <start>
                     <tag_size> "," <mode> "," [<meta_device> ","] [<block_size> ","] <recalculate> ","
                     <allow_discards> "," <fix_padding> "," <fix_hmac> "," <legacy_recalculate> ","
                     <journal_sectors> "," <interleave_sectors> "," <buffer_sectors> ";"

target_name := "target_name=integrity"
target_version := "target_version=" <N> "." <N> "." <N>
dev_name := "dev_name=" <device_name_str>
start := "start=" <N>
tag_size := "tag_size=" <N>
mode := "mode=" <integrity_mode_str>
integrity_mode_str := "J" | "B" | "D" | "R"
meta_device := "meta_device=" <meta_device_str>
block_size := "block_size=" <N>
recalculate := "recalculate=" <yes_no>
allow_discards := "allow_discards=" <yes_no>
fix_padding := "fix_padding=" <yes_no>
fix_hmac := "fix_hmac=" <yes_no>
legacy_recalculate := "legacy_recalculate=" <yes_no>
journal_sectors := "journal_sectors=" <N>
interleave_sectors := "interleave_sectors=" <N>
buffer_sectors := "buffer_sectors=" <N>
yes_no := "y" | "n"

E.g.
```

```
When a 'integrity' target is loaded, then IMA ASCII measurement log will have an entry
similar to the following, depicting what 'integrity' attributes are measured in EVENT_DATA
for 'dm_table_load' event.
(converted from ASCII to text for readability)

dm_version=4.45.0;
name=integrity1,uuid=,major=253,minor=1,minor_count=1,num_targets=1;
target_index=0,target_begin=0,target_len=7856,target_name=integrity,target_version=1.10.0,
dev_name=253:0,start=0,tag_size=32,mode=J,recalculate=n,allow_discards=n,fix_padding=n,
fix_hmac=n,legacy_recalculate=n,journal_sectors=88,interleave_sectors=32768,buffer_sectors=128;
```

### 4. linear

The 'target_attributes' (described as part of EVENT_DATA in 'Table load' section above) has the following data format for 'linear' target.

```
target_attributes := <target_name> "," <target_version> "," <device_name> <,> <start> ";"

target_name := "target_name=linear"
target_version := "target_version=" <N> "." <N> "." <N>
device_name := "device_name=" <linear_device_name_str>
start := "start=" <N>

E.g.
When a 'linear' target is loaded, then IMA ASCII measurement log will have an entry
similar to the following, depicting what 'linear' attributes are measured in EVENT_DATA
for 'dm_table_load' event.
(converted from ASCII to text for readability)

dm_version=4.45.0;
name=linear1,uuid=linear_uuid1,major=253,minor=2,minor_count=1,num_targets=1;
target_index=0,target_begin=0,target_len=28672,target_name=linear,target_version=1.4.0,
device_name=253:1,start=2048;
```

### 5. mirror

The 'target_attributes' (described as part of EVENT_DATA in 'Table load' section above) has the following data format for 'mirror' target.

```
target_attributes := <target_name> "," <target_version> "," <nr_mirrors> ","
                     <mirror_device_data> "," <handle_errors> "," <keep_log> "," <log_type_status> ";"

target_name := "target_name=mirror"
target_version := "target_version=" <N> "." <N> "." <N>
nr_mirrors := "nr_mirrors=" <NR>
mirror_device_data := <mirror_device_row> | <mirror_device_data><mirror_device_row>
                      mirror_device_row is repeated <NR> times - for <NR> described in <nr_mirrors>.
mirror_device_row := <mirror_device_name> "," <mirror_device_status>
mirror_device_name := "mirror_device_" <X> "=" <mirror_device_name_str>
                      where <X> ranges from 0 to (<NR> -1) - for <NR> described in <nr_mirrors>.
mirror_device_status := "mirror_device_" <X> "_status=" <mirror_device_status_char>
                        where <X> ranges from 0 to (<NR> -1) - for <NR> described in <nr_mirrors>.
mirror_device_status_char := "A" | "F" | "D" | "S" | "R" | "U"
handle_errors := "handle_errors=" <yes_no>
keep_log := "keep_log=" <yes_no>
log_type_status := "log_type_status=" <log_type_status_str>
yes_no := "y" | "n"

E.g.
When a 'mirror' target is loaded, then IMA ASCII measurement log will have an entry
similar to the following, depicting what 'mirror' attributes are measured in EVENT_DATA
for 'dm_table_load' event.
(converted from ASCII to text for readability)

dm_version=4.45.0;
name=mirror1,uuid=mirror_uuid1,major=253,minor=6,minor_count=1,num_targets=1;
target_index=0,target_begin=0,target_len=2048,target_name=mirror,target_version=1.14.0,nr_mirrors=2,
   mirror_device_0=253:4,mirror_device_0_status=A,
   mirror_device_1=253:5,mirror_device_1_status=A,
handle_errors=y,keep_log=n,log_type_status=;
```

### 6. multipath

The 'target_attributes' (described as part of EVENT_DATA in 'Table load' section above) has the following data format for 'multipath' target.

```
target_attributes := <target_name> "," <target_version> "," <nr_priority_groups>
                     ["," <pg_state> "," <priority_groups> "," <priority_group_paths>] ";"

target_name := "target_name=multipath"
target_version := "target_version=" <N> "." <N> "." <N>
nr_priority_groups := "nr_priority_groups=" <NPG>
priority_groups := <priority_groups_row>|<priority_groups_row><priority_groups>
priority_groups_row := "pg_state_" <X> "=" <pg_state_str> "," "nr_pgpaths_" <X>  "=" <NPGP> ","
                       "path_selector_name_" <X> "=" <string> "," <priority_group_paths>
                       where <X> ranges from 0 to (<NPG> -1) - for <NPG> described in <nr_priority_groups>.
pg_state_str := "E" | "A" | "D"
<priority_group_paths> := <priority_group_paths_row> | <priority_group_paths_row><priority_group_paths>
priority_group_paths_row := "path_name_" <X> "_" <Y> "=" <string> "," "is_active_" <X> "_" <Y> "=" <is_active_str>
                            "fail_count_" <X> "_" <Y> "=" <N> "," "path_selector_status_" <X> "_" <Y> "=" <path_selector
                            where <X> ranges from 0 to (<NPG> -1) - for <NPG> described in <nr_priority_groups>,
                            and <Y> ranges from 0 to (<NPGP> -1) - for <NPGP> described in <priority_groups_row>.
is_active_str := "A" | "F"

E.g.
When a 'multipath' target is loaded, then IMA ASCII measurement log will have an entry
similar to the following, depicting what 'multipath' attributes are measured in EVENT_DATA
for 'dm_table_load' event.
(converted from ASCII to text for readability)

dm_version=4.45.0;
name=mp,uuid=,major=253,minor=0,minor_count=1,num_targets=1;
target_index=0,target_begin=0,target_len=2097152,target_name=multipath,target_version=1.14.0,nr_priority_groups=2,
   pg_state_0=E,nr_pgpaths_0=2,path_selector_name_0=queue-length,
       path_name_0_0=8:16,is_active_0_0=A,fail_count_0_0=0,path_selector_status_0_0=,
```

```
        path_name_0_1=8:32,is_active_0_1=A,fail_count_0_1=0,path_selector_status_0_1=,
    pg_state_1=E,nr_pgpaths_1=2,path_selector_name_1=queue-length,
        path_name_1_0=8:48,is_active_1_0=A,fail_count_1_0=0,path_selector_status_1_0=,
        path_name_1_1=8:64,is_active_1_1=A,fail_count_1_1=0,path_selector_status_1_1=;
```

### 7. raid

The 'target_attributes' (described as part of EVENT_DATA in 'Table load' section above) has the following data format for 'raid' target.

```
target_attributes := <target_name> "," <target_version> "," <raid_type> "," <raid_disks> "," <raid_state>
                     <raid_device_status> ["," journal_dev_mode] ";"

target_name := "target_name=raid"
target_version := "target_version=" <N> "." <N> "." <N>
raid_type := "raid_type=" <raid_type_str>
raid_disks := "raid_disks=" <NRD>
raid_state := "raid_state=" <raid_state_str>
raid_state_str := "frozen" | "reshape" |"resync" | "check" | "repair" | "recover" | "idle" |"undef"
raid_device_status := <raid_device_status_row> | <raid_device_status_row><raid_device_status>
                      <raid_device_status_row> is repeated <NRD> times - for <NRD> described in <raid_disks>.
raid_device_status_row := "raid_device_" <X> "_status=" <raid_device_status_str>
                          where <X> ranges from 0 to (<NRD> -1) - for <NRD> described in <raid_disks>.
raid_device_status_str := "A" | "D" | "a" | "-"
journal_dev_mode := "journal_dev_mode=" <journal_dev_mode_str>
journal_dev_mode_str := "writethrough" | "writeback" | "invalid"

E.g.
When a 'raid' target is loaded, then IMA ASCII measurement log will have an entry
similar to the following, depicting what 'raid' attributes are measured in EVENT_DATA
for 'dm_table_load' event.
(converted from ASCII to text for readability)

dm_version=4.45.0;
name=raid_LV1,uuid=uuid_raid_LV1,major=253,minor=12,minor_count=1,num_targets=1;
target_index=0,target_begin=0,target_len=2048,target_name=raid,target_version=1.15.1,
raid_type=raid10,raid_disks=4,raid_state=idle,
   raid_device_0_status=A,
   raid_device_1_status=A,
   raid_device_2_status=A,
   raid_device_3_status=A;
```

### 8. snapshot

The 'target_attributes' (described as part of EVENT_DATA in 'Table load' section above) has the following data format for 'snapshot' target.

```
target_attributes := <target_name> "," <target_version> "," <snap_origin_name> ","
                     <snap_cow_name> "," <snap_valid> "," <snap_merge_failed> "," <snapshot_overflowed> ";"

target_name := "target_name=snapshot"
target_version := "target_version=" <N> "." <N> "." <N>
snap_origin_name := "snap_origin_name=" <string>
snap_cow_name := "snap_cow_name=" <string>
snap_valid := "snap_valid=" <yes_no>
snap_merge_failed := "snap_merge_failed=" <yes_no>
snapshot_overflowed := "snapshot_overflowed=" <yes_no>
yes_no := "y" | "n"

E.g.
When a 'snapshot' target is loaded, then IMA ASCII measurement log will have an entry
similar to the following, depicting what 'snapshot' attributes are measured in EVENT_DATA
for 'dm_table_load' event.
(converted from ASCII to text for readability)

dm_version=4.45.0;
name=snap1,uuid=snap_uuid1,major=253,minor=13,minor_count=1,num_targets=1;
target_index=0,target_begin=0,target_len=4096,target_name=snapshot,target_version=1.16.0,
snap_origin_name=253:11,snap_cow_name=253:12,snap_valid=y,snap_merge_failed=n,snapshot_overflowed=n;
```

### 9. striped

The 'target_attributes' (described as part of EVENT_DATA in 'Table load' section above) has the following data format for 'striped' target.

```
target_attributes := <target_name> "," <target_version> "," <stripes> "," <chunk_size> ","
                     <stripe_data> ";"

target_name := "target_name=striped"
target_version := "target_version=" <N> "." <N> "." <N>
stripes := "stripes=" <NS>
chunk_size := "chunk_size=" <N>
stripe_data := <stripe_data_row>|<stripe_data><stripe_data_row>
stripe_data_row := <stripe_device_name> "," <stripe_physical_start> "," <stripe_status>
stripe_device_name := "stripe_" <X> "_device_name=" <stripe_device_name_str>
                      where <X> ranges from 0 to (<NS> -1) - for <NS> described in <stripes>.
stripe_physical_start := "stripe_" <X> "_physical_start=" <N>
                         where <X> ranges from 0 to (<NS> -1) - for <NS> described in <stripes>.
stripe_status := "stripe_" <X> "_status=" <stripe_status_str>
                 where <X> ranges from 0 to (<NS> -1) - for <NS> described in <stripes>.
stripe_status_str := "D" | "A"

E.g.
When a 'striped' target is loaded, then IMA ASCII measurement log will have an entry
similar to the following, depicting what 'striped' attributes are measured in EVENT_DATA
for 'dm_table_load' event.
(converted from ASCII to text for readability)

dm_version=4.45.0;
name=striped1,uuid=striped_uuid1,major=253,minor=5,minor_count=1,num_targets=1;
target_index=0,target_begin=0,target_len=640,target_name=striped,target_version=1.6.0,stripes=2,chunk_size=64,
   stripe_0_device_name=253:0,stripe_0_physical_start=2048,stripe_0_status=A,
   stripe_1_device_name=253:3,stripe_1_physical_start=2048,stripe_1_status=A;
```

## 10. verity

The 'target_attributes' (described as part of EVENT_DATA in 'Table load' section above) has the following data format for 'verity' target.

```
target_attributes := <target_name> "," <target_version> "," <hash_failed> "," <verity_version> ","
                     <data_device_name> "," <hash_device_name> "," <verity_algorithm> "," <root_digest> ","
                     <salt> "," <ignore_zero_blocks> "," <check_at_most_once> ["," <root_hash_sig_key_desc>]
                     ["," <verity_mode>] ";"

target_name := "target_name=verity"
target_version := "target_version=" <N> "." <N> "." <N>
hash_failed := "hash_failed=" <hash_failed_str>
hash_failed_str := "C" | "V"
verity_version := "verity_version=" <verity_version_str>
data_device_name := "data_device_name=" <data_device_name_str>
hash_device_name := "hash_device_name=" <hash_device_name_str>
verity_algorithm := "verity_algorithm=" <verity_algorithm_str>
root_digest := "root_digest=" <root_digest_str>
salt := "salt=" <salt_str>
salt_str := "-" <verity_salt_str>
ignore_zero_blocks := "ignore_zero_blocks=" <yes_no>
check_at_most_once := "check_at_most_once=" <yes_no>
root_hash_sig_key_desc := "root_hash_sig_key_desc="
verity_mode := "verity_mode=" <verity_mode_str>
verity_mode_str := "ignore_corruption" | "restart_on_corruption" | "panic_on_corruption" | "invalid"
yes_no := "y" | "n"

E.g.
When a 'verity' target is loaded, then IMA ASCII measurement log will have an entry
similar to the following, depicting what 'verity' attributes are measured in EVENT_DATA
for 'dm_table_load' event.
(converted from ASCII to text for readability)

dm_version=4.45.0;
name=test-verity,uuid=,major=253,minor=2,minor_count=1,num_targets=1;
target_index=0,target_begin=0,target_len=1953120,target_name=verity,target_version=1.8.0,hash_failed=V,
verity_version=1,data_device_name=253:1,hash_device_name=253:0,verity_algorithm=sha256,
root_digest=29cb87e60ce7b12b443ba6008266f3e41e93e403d7f298f8e3f316b29ff89c5e,
salt=e48da609055204e89ae53b655ca2216dd983cf3cb829f34f63a297d106d53e2d,
ignore_zero_blocks=n,check_at_most_once=n;
```