SELinux -FAQ

Q 1 . **What is selinux ? do I need to care about it as a developer ?**

A. Security-Enhanced Linux (SELinux) is an Linux kernel security module (LSM) that provides a mechanism for supporting access control security policies which is based on mandatory access controls (MAC).

As a developer, you need to understand that malicious actors can abuse a program to gain control of the device. SELinux ensures that a program only uses a set of functionality that is needed and this mitigates a larger threat to confidentiality and integrity of the device and data.

Concepts is driver owner defines /documents set of permission which are needed for working (called as policy) . This policy become part of the build and during runtime anything beyond pre-document permission will hit Denial-of-Service .

For additional reading you can refer to go/selinux presentations.

Q 2 .What all mode are possibe in SELinux ?

A. SELinux in on OWRT has 3 mode , Disabled mode , Permissive Mode , Enforce Mode

Disable mode : No restriction /checks kernel and stack is not aware of selinux or policy files .

Permissive mode : In this mode it will not block any calls but still show logs on missing permission we call it denials .

Enforced mode : In this mode it will block call which dont have policy /permission pre-document and made part of the build.

Permissive mode is only for debugging or development .

Q 3 .Why type of speolicy is used in Openwrt ?

Openwrt can be configured with dssp (**Data Security Service Policy**) or refpolicy , Currently OWRT.PRODUCT.xx is using dssp model.

As the activity on this set for openwrt usecase is hight we prefered to go with this .

Q4 .Why type of Software Image (SI) are using dssp model ?

All SI like OWRT.PRODUCT.XX are using dssp model . JFI IPQ are not enabled with SELinux .

Q 5 .How to move to permissive mode /disable selinux for testing ? want to validate fast and add policy later ?

End user device and product devices are expected be always in enforce mode there is no exception to this . .
There are 3 ways to change into different modes .

1. Using " setenforce 0" in root shell (not persistence over reboot this will move to permissive from that time (t0) any thing before (t0-x) will be in enforce mode .
2. Using the /etc/selinux/config file , edit the SELINUX to permissive and reboot (Persistent across reboot)
3. Using the kernel cmdline (currently its disabled as we don't see the need)

We recommend to use the second method which is mostly handy .

Note : on overlay enabled build you might see issue that /etc/selinux/config changes to permissive is not reflecting on reboots .

For such cases please use the following steps (umount and /etc has to be done)

update /etc/selinux/config to permissive -> this updates upper /etc (overlay)
umount -l /etc
update /etc/selinux/config to permissive -> this updates lower /etc (actual rootfs)
reboot

Q 6 . What is selinux denial ? how to read it ? What happens if we dont write sepolicy file ?

A. As said policy files are predefined permission which are to be made part of build , If we are not adding this rule it will hit a denial-of-service (DoS) . Where you features might not work in enforcemode .

A denial is the event generated anytime that a service, application, file, etc. is denied access by the SELinux system. When this happens, the denial is cached in the Access Vector Cache (AVC). You will sometimes see a denial message referred to as an AVC denial.

Let's consider an example:

avc: denied { getattr } for pid=250 comm="mount_root" path="/dev/ram14" dev="tmpfs" ino=9985 scontext=u:r:mountroot.subj tcontext=u:r:tmp.fs tclass=blk_file permissive=0

A typical SELinux denial message consists of a few parts:

1. {...} --> Type of permission denied.
2. scontext --> Which object requires this access
3. tcontext --> The resource for which the permission is requested
4. tclass --> The type of resource (in this case - blk_file)

To summarise this example, mount process (scontext) is trying to access tmp.fs (tcontext) with the permission getattr.

Q 7 . **How to add my own service with new CIL file .**

Currently the tree is organized based on the type of resource you are adding for bin/services you need to add in execute folder , dor devnode you need to add ing devnode .

Project that is used for adding CIL file is owrt-qti-sepolicy , (under owrt) we expect to create cil file for service/binary which need access to resource (socket /file/folder /mountpoint / ...)
service - owrt-qti-sepolicy/cil/files/execute
dev - owrt-qti-sepolicy/cil/files/dev

- cil file general format :

https://review-android.quicinc.com/#/c/4149478/

Q 8 . **How to just compile selinux cil files ? Fast way of checking sepolicy changes for compilation ?**

A. To compile selinux .cil files, the command is (assuming your current location is: openwrt/owrt):

Where first one will clean and second will compile the policy file.

make –j8 V=sc package/system/selinux-policy/clean

make –j8 V=sc package/system/selinux-policy/compile

Once done you can following the normal process of building the images .

Q 9. **How to have variable /macros shared across the cil files?**

create macro def inside block

decclaration : (macro <macro-name> ((type ARG1))
(allow rules or predefind macro calls)

example (macro xyz ((type ARG1))
(allow arg1 arg2 dir (search))

use macro (.call.<macro-name> ((type ARG1))

example : (.call.xyz(arg1))

Q 10. **How to get support on writing CIL files ?**

First thing is we want you to go throught slides /presentations on the home page and try it out .

confluence page for understanding openwrt selinux : go/openwrt-selinux

If you still have issues drop a email to owrtsepolicy.support or seandroid.support

POC : risidoji/ arukmang / hkhokhar /

Q 11 . **PoC and workflow in getting the CIL file changes approved .**

Once the change is validated we want the team to

- tag a CR , CR should be on the tech team area on which the policy is added for .
- DV should be added .
- Need to send a email to owrt.sepolicy.approvers <owrt.sepolicy.approvers@qti.qualcomm.com>/seandroid.support

Q 12. **What is dac_override and dac_search or dac _... permission are they allowed ?**

A. Any file /folder will have its own user and group permission (ls -l will show the details) .

You service / bin might be access a resource (file/folder/socket) where you are not part of the user /group and still want to access this then you will hit dac_override /dac_serarch /dac ....

We as a policy are not allowing any Qualcomm service /process to have this policy . It should be added /update with right user /group/supplement group . On how to add supplement group

you can visit following links

Supplementary groups mechanism in OpenWrt

Enabling service Capabilities in OWRT

FR52909:Deprivilege root processes

Q 13. **How to address opensource services denails ? How are opensource service sepolicy extended ?**

For handling opensource service denials we can inherit or extend opensource service cil file and add rules

example :

suppose xyz is a upstream service throwing selinux denial in dmesg logs

To address xyz upstream service denial

```
(in .xyz
(block xyz
    ;; policy
    (blockinherit .agent.base_template))
      (allow subj self (capability (<permission>)))
)
```

Q 14 . How CIL files folder tree organized ?

As of now all Qualcomm added cil files (which are needed for qualcomm driver + extension to upstream) are added to the ***owrt-qti-sepolicy*** project .

The tree is as below with 3 major folder dev – for dev nodes, file - for service /bin /exefiles , files – which are needed for selinux build and working .

```
cil
  dev
    nodedev
      gpsnodedev.cil
      qseecomnodedev.cil
      smcinvoke.cil
      smdnodedev.cil
  file
    execfile
      adbd.cil
      diag-router.cil
      init_mss.cil
      irsc_util.cil
      kmodloader-append.cil
      qcmap_cm.cil
      qexec.cil
      qlibs.cil
      qrtr-cfg.cil
      qrtr-lookup.cil
      qrtr-ns.cil
      rcbootaks.cil
      rclogaks.cil
      readbear.cil
      rmt_storage.cil
      tftp_server.cil
      ubustmp.cil
files
  selinux-labeldev.init
  selinux-labeldev.sh
README
```

Q 15. **Adding capabilities to my services? adding wakelock for my service ?**

Enabling service Capabilities in OWRT

Q 16. **How to see if uJail is active on my services?**

if you added ujail related code in your service init file , we can see ujail process running as a root corresponding

to service

example

```
066 root      2304 S    {QCMAP_Connectio} /sbin/ujail -n QCMAP_ConnectionMan
2069 radio    15268 S    /usr/bin/QCMAP_ConnectionManager /etc/data/mobileap
```

Q 17 **Memory and performance impact with Openwrt .**

- Memory requirement – showing 1.2 MB with SELinux (this is with initial data done during early days) adding the new policy files also should not go muchas 1.5MB with current policy database.

Q 18. **How to create a new UID and Group and use it to my service ?**

Default user/groups like (system/adb/logd /wakelocks) are expected to be created by platform team and they are already been added .

If you service/bin need a new userid/group you can refer

FR52909:Deprivilege root processes

Q 19. How to check if selinux is enabled and in which mode its running ?

A. you can get this details using "getenforce" utility from adb shell / serial console

Q 20. Pushed the .Ko to the /data/… and tried to do insmod its not working what is wrong with this why is not successful I am in root still it doesn't work and works in permissive ?

A. SElinux expect each file to be labeled (called secontext of the file) and only certain labeled files are going to be allowed for insmod . In this case we are pushing the .ko which will not have right label and will not work .
Only module which are part of /etc/modules/<kernel version > are expected to be insmod .

Q 21. How do I Just build sepolicy ?

A. To validate selinux changes for compile time you can use the following process

rm –rf /build_dir/target-arm_cortex-a7+neon-vfpv4_musl_eabi/selinux-policy-0.8

make –j8 V=sc packages/system/selinux-policy/clean
make –j8 V=sc packages/system/selinux-policy/compile
above will just compile the sepolicy and will not reflect the image , you have to trigger the respective image /targe/Linux/sdx….. build again for images

Q 22. How to decode inod from avc denails ? how to know the path of file /folder/socket/resouce from denial ?

Openwrt 2.0 (for sdx75) has find utilty with inode support .

for the denials you can see the ino use this in find utilty

find -inum <ino show in the denails >

If you know the folder /fs you can pass that also like if its in sys then

find /sys -inum <ino >

find /proc -inum <ino>

Q 23. **How to create sockets** ?

A. Advise to create sockets :

1. general way is use the startupscript (.init (openwrt) or .service (in OE)) to create sockets
   .

https://opengrok.qualcomm.com/source/xref/OWRT.PRODUCT.1.0/owrt/owrt-qti-data-prop/owrt-packages/data/netmgr/files/netmgrd.init#13

https://opengrok.qualcomm.com/source/xref/OWRT.PRODUCT.1.0/owrt/owrt-qti-location-prop/recipes/gps/feeds/loc-launcher/files/loc-launcher.init#22

1. Use any startupscripts or preinit script to create the sockets which will be running in root and they will create requested folder and socket .

Q 24. **How to add rules based on some dependency/ how to make sure my file will not break compile as dependency is not there** ?

There are different option available ,like bool , optional , tuneable sort of keyword for handling such cases .

- Booleanif

;;Example for booleanif,

```
(booleanif EXPR
        (true
                (allow a b (file (read))))
        (false
                (allow a b (file (read write)))))
```

- Optional rules

used to define a set of rules that might (expectantly) fail because types are defined in another CIL policy file that doesn't exist. If at any point the declaration for a symbol cannot be found, the optional rule is disabled.

Example for optional rules

; automatic dependencies ; Always has a name associated with the block. (optional apache (allow foo bar (file (write)))) (optional foobar (type whatever) (allow whatever foo (file (write))))

Q 25 How to allow neverallow rules written in upstream selinux code ?

example - In selinux version 1.1 CIL file implementation read/write on stordev blocks is not allowed

```
 (neverallow not_subj_typeattr obj_typeattr (blk_file (read)))
(neverallow not_subj_typeattr obj_typeattr (blk_file (write)))
```

let's say rmt_storage wants read/write permission for mmc.stordev block

when we directly write allow rule we can see neverallow check fail for rmt_storage service

Rule : (call .mmc.readwrite_stordev_blk_files (subj))

NeverAloow hits during compilation

```
Compile post process
Building policy binary
Checking Neverallows
neverallow check failed at src/dev.cil:512
  (neverallow dev.stor.write.not_subj_typeattr dev.stor.obj_typeattr (blk_file (append write)))

<root>
    block at src/rmt_storage.cil:14
    allow at src/rmt_storage.cil:74
      (allow rmt_storage.subj mmc.stordev (blk_file (read write)))
<root>
    block at src/rmt_storage.cil:14
    call at src/rmt_storage.cil:80
    allow at src/dev.cil:251
      (allow rmt_storage.subj mmc.stordev readwrite_blk_file)
```

How to avoid neverallow check for rmt_storage block?

To bypass neverallow read/write permission , add below macro

(call .dev.stor.readwrite.subj_type (subj)) - using this macro we are allowing rmt_storage to read/write on mmc.stordev

Similar macros are written for other permissions to avoid neverallow checks , see example change below

Gerrit: owrt-sepolicy: added r/w to mmc block for rmt_storage (Ie77aec3b) · Gerrit Code Review (quicinc.com)

Q 26 Seeing denails with sys.rootfile can we add any run-on sys.rootfile ?

As per desing we want the /) to be read only there should be any new folder or file added to this.
Which could lead to adding following rule for file/directory which is not going to be allowed.
(allow subj .sys.rootfile (dir(open read write)))
if the folder /file is needed create this during build time. if this is not the case and needed at run time
we want your init file to create this under your own tech team folder.

Q 27. How to move a particular service to permissive ?

(typepermissive subj) – instead of full system only this domain will be in permissive mode