*Best practice for applying Program Temporary Fixes (PTFs) for kernel in SUSE Edge*

## Environment

SUSE Linux Micro

## Situation

SUSE Technical Support provided a set of packages to fix a given situation (so called PTF = Program Temporary Fix).

For more information on downloading and applying PTF files in SUSE Linux, please see the "Best practice for applying Program Temporary Fixes (PTFs) article".

## Resolution

How should a PTF be applied?

SUSE Edge runs on top of SUSE Linux Micro operating system which uses an immutable filesystem. Therefore we are presented with a few different options on how to apply the PTF, depending on the use case.

| Procedure | Pros | Cons |
|---|---|---|
| Install the PTF on a running system | Easiest<br>No need to redeploy | Reboot required |
| Create a new image with EIB | Embedded into the image | Reboot required<br>Redeploy needed |
| Create a new image with Kiwi | Embedded into the image | Redeploy needed<br>More elaborated<br>Only option if PTF addresses boot issues |

### *1. Install the PTF on a running system via a transactional update.*

*NOTE:* The system will need to be rebooted for the PTF kernel to be running.

A simple workflow for this would be like:

- Download the PTF RPMs
- Copy the RPMs to the host, for example using scp:

```
scp *PTF*rpm user@slmicrohost:/tmp/
```

- Install the PTFs:

```
ssh user@slmicrohost
sudo transactional-update pkg install /tmp/*PTF*rpm
```

*NOTE:* In the case “Signature verification failed” messages are observed, please use “i” to ignore, otherwise the PTF will not be installed.

- Reboot the node manually:

```
sudo reboot
```

Once the system is back up, verify the rpm is installed with :

```
sudo rpm -qa | grep -i PTF
```

### *2. Use Edge Image Builder (EIB) to build a new image:*

Build a new image for your cluster with the same base image and the PTF kernel applied on it, same as before with Edge Image Builder. You will need to reboot though as applying a kernel on an existing image happens on combustion stage. A simple workflow for this would be:

- Create a `rpms` directory in your main Edge Image Builder directory and inside it store the rpm(s), a directory for the `gpg-keys` (more info on this can be found on the Edge Image Builder documentation) and the `suse_ptf_key.asc` file as:

```
rpms/
├── gpg-keys
│   └── suse_ptf_key.asc
└── kernel-default-6.4.0-150600.23.33.1.XXXXX.1.PTF.XXXXXXX.x86_64.rpm
```

The gpg key for SUSE PTF files is:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.15 (GNU/Linux)

mQENBGIYzssBCADJN/hpDHueuhfWHTik9KrUjQKZv/kwByISSQIWyuXW9WcmUbo/
pYiPgCjq1xNX58Yw+Rh0BQet9QFUYWFhJqG/3lc38UHByM3Wq/3dEXHtNoQGosBb
zVQfDznD6EHuikU89+ErcX05suscA7og2YNMNl/Yt4v2VoFOqeKuvYGtRxcTzgHi
8xBbj6EN2X3XnsubddrCiwdoXKq/It0vGRcmez+1EVbAjQEasoIQkxtRzLDbwcfK
eM4vKyhQ+ziUe4jadp58VsvIbKxA+kfNegaHsYLn11cdUjc2o5eiPFWbERz1x2pE
gUUaVD8M8RJdllZbWKhmaI06+RZYbmiYTRlPABEBAAG0J1NVU0UgUFRGIFNpZ25p
bmcgS2V5IDxzdXBwb3J0QHN1c2UuY29tPokBPgQTAQIAKAUCYhjOywIbAwUJB4TO
AAYLCQgHAwIGFQgCCQoLBBYCAwECHgECF4AACgkQRt+gXG9dpiuboQgAuKKA7JmH
F4tiW5dacBPbNF3hjX4AH4amFn09WyvpjLodANpr5XI4guzykzaxPqeWUQNKC1gs
+Qg0NtskJ4sHHH1V7xR1OU0NFL/RTX9q8K1ICk0WaQE+GhLQm+IwCmF6SKnEGfro
iXbcHTDKtv4VlYRO9mYnRU2BJQpi7T82kWg5ipG2StRDJ355GPqp4ciul4hmSIuH
yX5MMc8yXZa7Wf/VALQTbcIva3oePaz4QJeg0sGL3SgoBIp+XCm1fNGfdoUpamLH
f3svFdbT1XlgXmhgU9K6hTiqD+qUIx2HWRq/15imY31cAeGVM8fBnr5s4UrPRjXH
jFIZZtawT40NLA==
=Du7L
-----END PGP PUBLIC KEY BLOCK-----
```

- Then proceed with building the image with Edge Image Builder as you would ordinarily do.
- The package upgrade happens at the combustion stage, so the reboot will be necessary:

```
combustion[1392]: The following package requires a system reboot:
combustion[1392]:   kernel-default
combustion[1392]: 1 package to upgrade, 5 new.
combustion[1392]: Overall download size: 115.0 MiB. Already cached: 0 B.
After the operation, additional 7.4 MiB will be used.
combustion[1392]:     Note: System reboot required.
```

*3. Use the Kiwi tool to build an image with the PTF kernel embedded:*

For more info on this please see Building Updated SUSE Linux Micro Images with Kiwi

- An example on using kiwi:

Create a folder to host the assets and the output:

```
mkdir -p ~/ptf/
mkdir -p ~/output/
```

Create a new custom.kiwi file copying the original SL-Micro.kiwi one:

```
podman create --name kiwi-builder
registry.suse.com/edge/3.2/kiwi-builder:10.1.16.0
podman cp kiwi-builder:/micro-sdk/defs/SL-Micro.kiwi ~/ptf/custom.kiwi
podman rm kiwi-builder
```

Perform the following changes into the `custom.kiwi` file: Create a new `ptf-1234-SelfInstall` profile that also requires a new `ptf-1234-x86-self_install` profile.

```
<profile name="ptf-1234-SelfInstall" description="PTF-1234 Custom SL Micro
with Podman and KVM as raw image with uEFI boot - SelfInstall" arch="x86_64">
    <requires profile="full"/>
    <requires profile="ptf-1234-x86-self_install"/>
    <requires profile="self_install"/>
</profile>
```

Add also a new packages section for that profile to include the PTF kernel needed to install:

```
    <packages type="image" profiles="ptf-1234-x86-self_install">
        <package
name="kernel-default-6.4.0-150600.23.33.1.XXXXX.1.PTF.XXXXXXX.x86_64"/>
        <package name="kernel-firmware-all"/>
    </packages>
```

The bootloader profile needs to be also explicitly satisfied:

```
        <profile name="ptf-1234-x86-self_install" description="Raw disk for
x86_64 - uEFI" arch="x86_64">
            <requires profile="bootloader"/>
        </profile>
```

For the rest of profile definitions, we can just add the new `ptf-1234-x86-self_install` profile to them:

```
    <preferences
profiles="x86-self_install,x86-rt-self_install,ptf-1234-x86-self_install">
        <version>6.0</version>
        <packagemanager>zypper</packagemanager>
        <bootsplash-theme>SLE</bootsplash-theme>
...


    <packages type="image"
profiles="x86,x86-encrypted,x86-rt-encrypted,x86-self_install,x86-legacy,x86-
vmware,x86-rt,x86-rt-self_install,x86-qcow,aarch64-qcow,rpi,aarch64-self_inst
all,ptf-1234-x86-self_install">
        <package name="dracut-kiwi-oem-repart"/>
        <package name="dracut-kiwi-oem-dump"/>
    </packages>
```

Create a folder to store the PTF rpm:

```
mkdir -p ~/ptf/repo/
```

Copy the rpm over there and run `createrepo`

```
cp kernel-default-6.4.0-150600.23.33.1.XXXXX.1.PTF.XXXXXXX.x86_64.rpm
~/ptf-repo/
cd ~/ptf/repo/
createrepo .
```

Create a repofile and save it to the `ptf` folder:

```
cat << EOF > ~/ptf/ptf-1234.repo
[ptf-1234]
name=PTF-1234
baseurl=file:///ptf-1234/
autorefresh=1
EOF
```

This is how the folder should look like:

```
/home/opensuse/ptf
├── custom.kiwi
├── ptf-1234.repo
└── repo
    ├── kernel-default-6.4.0-150600.23.33.1.XXXXX.1.PTF.XXXXXXX.x86_64.rpm
```

```
    └── repodata
        ├──
0eb2e2b1301bf60518ac005d415071c5f53b6d969e764339ccf4c92f9c859469-other.xml.zs
t
        ├──
2a4f80987670af90bfcbe880ae35fa130100463a3bc212802b70c29c0e73f543-primary.xml.
zst
        ├──
c6e176ac091a7846958b99fad998f08b614b5b8198e47df8bac149bbe662a6de-filelists.xm
l.zst
        └── repomd.xml
```

Run the podman command with the proper paths for the ptf-repo folder and the proper profile (ptf-1234-SelfInstall)

```
sudo podman run --privileged \
                -v ${HOME}/ptf/repo:/ptf-1234 \
                -v ${HOME}/ptf/custom.kiwi:/micro-sdk/defs/SL-Micro.kiwi \
                -v /etc/zypp/repos.d:/micro-sdk/repos/ \
                -v ${HOME}/ptf/ptf-1234.repo:/micro-sdk/repos/ptf-1234.repo \
                -v ${HOME}/output:/tmp/output \
                -it registry.suse.com/edge/3.2/kiwi-builder:10.1.16.0 \
                build-image -p ptf-1234-SelfInstall
```

If running this on a non Micro 6 host, the `/etc/zypp/repos.d/` folder can be copied over the host and adjust the folder like:

```
sudo podman run --privileged \
                -v ${HOME}/ptf/repo:/ptf-1234 \
                -v ${HOME}/ptf/custom.kiwi:/micro-sdk/defs/SL-Micro.kiwi \
                -v ${HOME}/ptf/repofiles/:/micro-sdk/repos/ \
                -v ${HOME}/ptf/ptf-1234.repo:/micro-sdk/repos/ptf-1234.repo \
                -v ${HOME}/output:/tmp/output \
                -it registry.suse.com/edge/3.2/kiwi-builder:10.1.16.0 \
                build-image -p ptf-1234-SelfInstall
```

As you can see the last option is more elaborate, but it can be useful if you can’t reboot the nodes and prefer to rebuild them,or in the case that the PTF is fixing boot related issues where it’s your only option.