ONIE Project Status Update 2021

Alex Doyle, ONIE Project Lead, Nvidia Corporation
Overview

• Project lead (or, “What would you say ya do here?”)
• Introduction, statistics and thanks.
• ONIE project goals.
  ○ The stereotype of the “Lazy Pro gramer”
• What’s new since the last Project Update.
  ○ Component upgrades have broken most platforms.
  ○ Improved emulation, Secure Boot, and security extensions.
• Future plans.
  ○ A secure installer.
  ○ New network edge applications for ONIE and Wi-Fi support.
• Question Time
ONIE Project Lead

• I’ve been project lead for the last three years.
• Started with Cumulus Networks which was bought by Nvidia.
  ○ So same job, different email.
• Responsibilities include:
  ○ ONIE Releases
  ○ Processing pull requests
  ○ Documentation
  ○ QA
  ○ Setting the direction of the project.
  ○ ...and everything else managing ONIE requires.
What’s ONIE?

- **Open Network Install Environment**
- A small, open source Linux operating system for installing other operating systems.
- Built by manufacturers to support their particular hardware and is installed on the devices they ship.
- Typically used on network switches.
- Allows the end user to install their choice of Network Operating System on a switch.
- It has become an industry standard for NOS install.
Statistics Since Last Year

• Over 144 sign-offs.
• Over 244 platforms supported.
• Thanks to many individual contributors and others from:

<table>
<thead>
<tr>
<th>Accton Technology</th>
<th>Alpha Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centec</td>
<td>Delta Networks</td>
</tr>
<tr>
<td>Inventec Corporation</td>
<td>Marvell</td>
</tr>
<tr>
<td>Netberg</td>
<td>Nvidia Corporation</td>
</tr>
<tr>
<td>Pegatron Corporation</td>
<td>Quanta Cloud Technology</td>
</tr>
</tbody>
</table>
ONIE Goals For the Last Three Years

• The Stereotypical “Lazy Programmer” knows:
  ○ One has to maximize the reward for effort.
    ■ i.e.: Work smarter, not harder.
  ○ Making the task easier for oneself makes it easier for everyone.
  ○ Do it right the first time and you won’t need a second time.
  ○ Setting up your workspace properly saves time later.
    ■ ...and counts as progress.
    ■ Painters know this.

• Question: with limited resources, what provides the most benefit?
• Answer: I went with setting up the workspace properly.
Setting up the workspace properly

• Focused on infrastructure improvements.
• Component and build tool upgrades are finally in place
  ○ Kernel is 5.4.x, Grub is 2.04, etc...
  ○ ONIE builds on Debian 10 (soon to be Debian 11) systems.
• The open source tool Dedicated User Environment provides ONIE build containers for Docker or Podman to help with:
  ○ Build configuration and reproducibility issues.
  ○ “Making it easier for everybody.”
• A number of pull requests and bug fixes have been processed. However, growth means change and change means some bad news...
What’s New: The Bad News

• No physical platforms currently build from the master branch.
• This is an expected consequence of component upgrade, as:
  ○ Newer compilers are more strict.
  ○ Patches don’t apply to new code.
  ○ Library interfaces change.
  ○ ...etc.
• Manufacturers will have to update their code to take advantage of the new features in the master branch.
  ○ Nobody else is qualified or has the resources.

That said, there is some good news...

OPEN POSSIBILITIES.
What’s New: The Good News

• Every platform that built in the **2021.08** release will always build.
  ○ ...because we have a “properly set up workspace”
• Docker images are available to provide Debian 9 build environments
• ONIE’s onie/build-config/scripts/`onie-build-targets.json` file maps build targets to known good build environments for ease of reproduction.

So… the workspace is set up. Now what?
Improved ARM64 Emulation

- The `qemu_arm64` target has been broken for the last few years.
- Fixing this was prioritized as more is being done with ARM64.
  - Shout out to Shi Lei from Centec for help fixing this.
- Builds:
  - ONIE
  - The installer .iso
  - The DemoOS
- It could use more testing.
  - “Don’t let the perfect be the enemy of the good.”
- The QEMU command line to run it is (predictably) a nightmare.
  - So it would be nice to have...
Easier Emulation

Simplify running emulation targets with **onie-vm.sh**

```
onie/emulation/
  emulation-files
    └── onie-kvm_x86_64-demo.qcow2 <- virtual hard drive
    └── uefi-bios
        └── x86
            └── OVMF_CODE.fd <- UEFI BIOS code
            └── OVMF_VARS.fd <- Store user set UEFI variables
    └── usb
        └── usb-drive.qcow2 <- virtual “USB Drive” for storage
    └── onie-vm.lib <- functions used by onie-vm.sh
    └── onie-vm.sh <- run a virtual machine
```

...and when I say “easier”, I really mean...
Much Easier Emulation

- Run: `onie-vm.sh run --m-bios-uefi --m-embed-onie --machine-name qemu_armv8a`
- Or:

```sh
qemu-system-aarch64 -machine virt -cpu cortex-a57 -drive if=pflash,format=raw,readonly, file=/onie/emulation/emulation-files/uefi-bios/arm-flash-files/flash0.img
-drive if=pflash,format=raw, file=/onie/emulation/emulation-files/uefi-bios/arm-flash-files/flash1.img
-smp 2 -m 2048 -name onie
-cdrom /onie/emulation/../../build/images/onie-recovery-arm64-qemu_armv8a-r0.iso
-drive index=0,if=none, file=/onie/emulation/emulation-files/onie-qemu_armv8a-demo.qcow2,id=hd
-device virtio-blk-pci,drive=hd,bootindex=0 -vnc 0.0.0.0:128
-device virtio-net,netdev=onienet,mac=52:54:00:13:34:1E
-netdev user,id=onienet,hostfwd=tcp::4022-:22 -nographic
-serial telnet:localhost:9300,server
```
Build Integration With Emulation

• With a known runtime environment, the build can take steps rather than asking the user to do so.
  ○ Copy cryptographic keys for installation.
  ○ Provide a UEFI install script to demonstrate key addition.
  ○ Provide step by step instructions during run time.
  ○ And **show**, not tell, how things should work.
  ○ This is *extremely* useful when dealing with security features that use multiple encryption keys to sign multiple files.

Speaking of keys and security, it would be nice to have...
Easier Encryption

- Key generation is...complicated.
- The encryption directory holds things “Left as an exercise for the reader.”
- A place holder for the manufacturer’s key handling

```
onie/encryption/
├── machines
│   └── kvm_x86_64
│       ├── keys
│       │   └── write-keys.nsh  <- UEFI script to add keys
│       │       └── safe-place  <- Files “stored elsewhere”
│       │           └── shimx64.efi  <- ONIE’s self signed shim
│       │                   └── shimx64.efi.unsigned
│   └── onie-encrypt.lib  <- Key generation functions
│       └── onie-encrypt.sh  <- Handle key generation
```

OPEN POSSIBILITIES.
Security Enhancements

• **Secure Boot** for *kvm_x86_64* virtual machine target.
  ○ Uses new emulation to demonstrate key deployment.
  ○ DemoOS is signed as well.

• **ONIE Password**
  ○ Requires username/password for login.

• **Secure GRUB**
  ○ Validates configuration files.
  ○ Requires a password.
  ○ ONIE KVM build demonstrates configuration file signing.

Example code is great, but how about some tips on porting features?
Security Feature Porting Tips

• The build target requires a `machine-security.make` Makefile.
  ○ Just like it already requires a machine.make Makefile.
  ○ It defines all security settings in one place for easy porting.
  ○ 1:1 mapping of keys with what they sign.
  ○ `kvm_x86_64` build target is the example.
• See the ONIE Git commit history to add features.
  ○ Commits are well documented, and are (mostly) one per feature.
• Shout out to Andriy Dobush and Michael Shych from Nvidia for doing most of the development work on these features.
What’s Next: The Immediate Future

- Bug fixes.
- Pull requests.
- Work on secure installer.
  - Have ONIE validate install images against its BIOS keys.
  - Will require yet more component upgrades.
  - ...and functional specifications.
  - …so that’ll be a bit.
  - ...but whoever gets good code in first, wins.
Beyond The Immediate Future

• Working with the Open Compute Project Enterprise Connectivity Services Group.
  ○ Using ONIE in network edge devices:
    ■ Upgrading ONIE’s build system.
    ■ Improved security.
    ■ Image upgrades outside of the datacenter.
    ■ Adding Wi-Fi support.
Call to Action

• Status calls are every third Wednesday
• Tutorial demo: https://www.youtube.com/watch?v=Oq4FWw9lkwQ
• Further information:

GitHub: https://github.com/opencomputeproject/onie
Mailing List : https://ocp-all.groups.io/g/OCP-ONIE
Documentation: https://opencomputeproject.github.io/onie/
OCP Enterprise Connectivity Solutions Mailing List: https://ocp-all.groups.io/g/OCP-ECS
ONIE Docker build environment: https://github.com/CumulusNetworks/DUE/tree/master/templates/onie
Open Discussion