Hewlett Packard Enterprise

UEFI HTTP/HTTPS Boot

LinuxCon China 2017

Keng-Yu Lin <kengyu@hpe.com>

June 20, 2017

Agenda

- UEFI HTTP(s) Boot introduction
- HPE UEFI HTTP Boot PoC based on `GRUB2`
- Share obstacles
- Open discussion

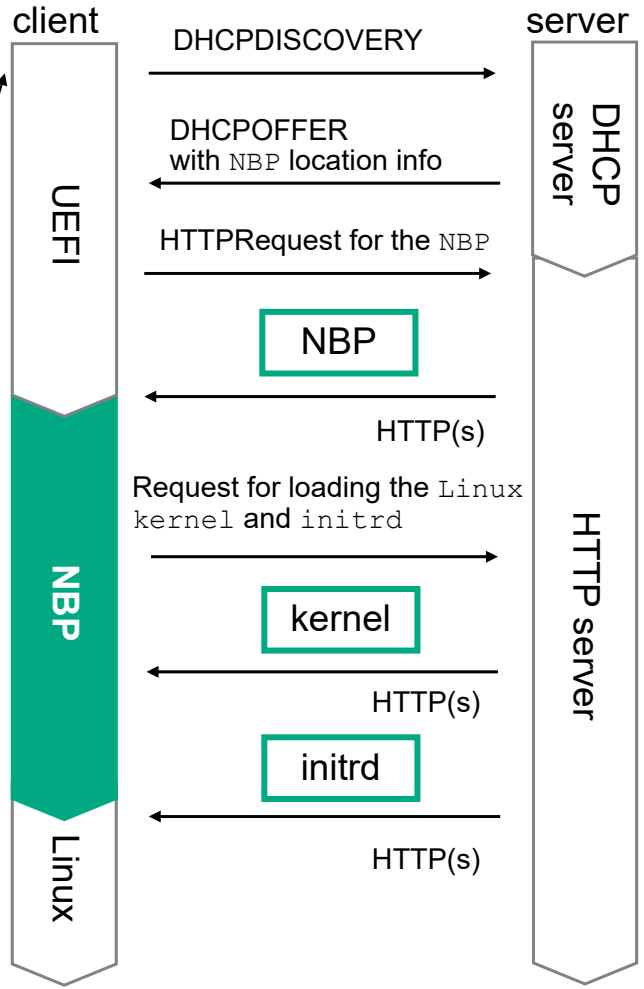
Comparison

UEFI HTTP(s) Boot	PXE
IPv4 & IPv6	IPv4 only
UEFI 2.5 plus + DHCP + HTTP server	UEFI or legacy BIOS + DHCP + TFTP server
Standard DNS setup	`dnsmasq` as the DNS forwarder
HTTP server has a variety of access control	TFTP has no access control
HTTP uses TCP → Reliable connection	TFTP uses UDP → Potential packet loss
SSL/TLS support (HTTPS)	N/A

Example

```
Boot Manager Menu

UEFI Floppy
UEFI Floppy 2
UEFI QEMU DVD-ROM QM00003
UEFI QEMU HARDDISK QM00001
UEFI PXEv4 (MAC:525400123456)
UEFI HTTPv4 (MAC:525400123456)
EFI Internal Shell
```

(screenshot from EDK2/OVMF)

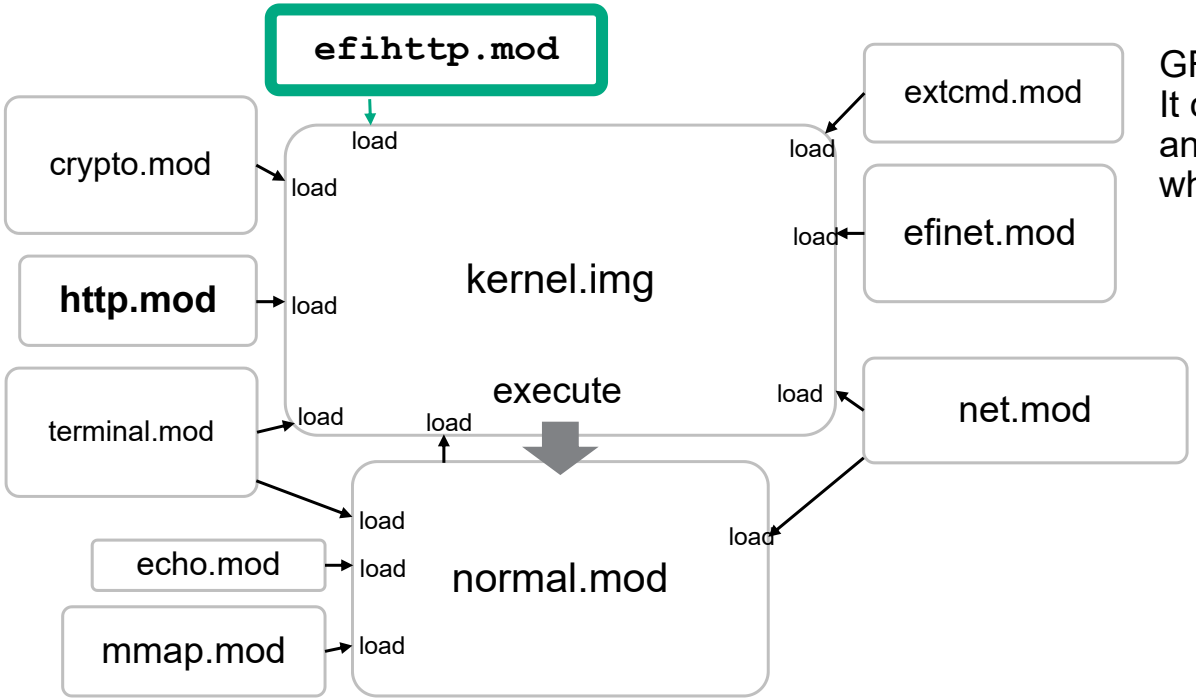


/etc/dhcp/dhcpd.conf

```
option domain-name "cloudboot.com";

option domain-name-servers 192.168.10.20;

option routers 192.168.10.1;

option vendor-class-identifier "HTTPClient";

option bootfile-name "http://www.cloudboot.com:8080/EFI/Shell.efi";
```

Hewlett Packard Enterprise

GRUB2 Modular Architecture

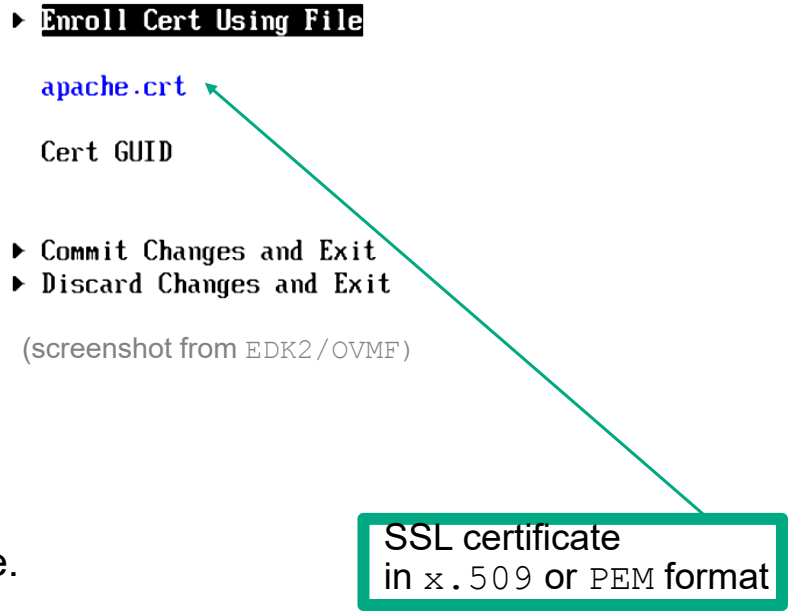


GRUB2 is not a single binary.
It contains a lot of separate modules and they are only loaded when needed.

net/drivers/efi/efihttp.c:193: Before grub_efihttp->request(), url:https://192.168.111.1/boot/grub/x86_64-efi/extcmd.mod

Software-Based Implementation

- Patches* from **HPE** & **SuSE**
  - Only use `http.mod` from `GRUB2`
- Obstacles of HTTPS
  - No `https.mod` in `GRUB2`
  - `GRUB2` to use `openssl` or `GnuTLS` is *error-prone*
  - Saving the certificates in software is dangerous
  - UEFI already provides good and simple APIs to use.
    - Disadvantages: only works on UEFI-enabled machines



* https://lists.gnu.org/archive/html/grub-devel/2016-08/msg00000.html
https://lists.gnu.org/archive/html/grub-devel/2016-12/msg00088.html

UEFI-Based HTTP Implementation

- **HPE** *PoC* works on `OVMF/QEMU`
- Preliminary test works on **HPE ProLiant Gen10** servers
- RFC patchset sent to the `GRUB2` upstream
- `GRUB2` maintainers' comments:
  - Prefers the *software-based* solution with `GnuTLS` library
    - Works on non-UEFI arches
  - Need *MNP* NIC driver rather than *SNP* for UEFI HTTP(s) protocols

* http://lists.gnu.org/archive/html/grub-devel/2017-01/msg00016.html

UEFI HTTP Protocol

- **DNS support**
  - EFI_DNS4_SERVICE_BINDING_PROTOCOL
  - EFI_DNS6_SERVICE_BINDING_PROTOCOL
  - EFI_DNS4_PROTOCOL
  - EFI_DNS6_PROTOCOL
  - EFI_IP4_CONFIG2_PROTOCOL
- **HTTP support**
  - EFI_HTTP_SERVICE_BINDING_PROTOCOL
  - EFI_HTTP_PROTOCOL
  - EFI_HTTP_UTILITIES_PROTOCOL
  - HTTP Boot Wire Protocol
- **TLS support**
  - EFI_TLS_SERVICE_BINDING_PROTOCOL
  - EFI_TLS_PROTOCOL
  - EFI_TLS_CONFIGURATION_PROTOCOL

EFI_HTTP_PROTOCOL

Protocol GUID

```
#define EFI_HTTP_PROTOCOL_GUID \
  {0x7A59B29B, 0x910B, 0x4171,\
   {0x82, 0x42, 0xA8, 0x5A, 0x0D, 0xF2, 0x5B, 0x5B}}
```

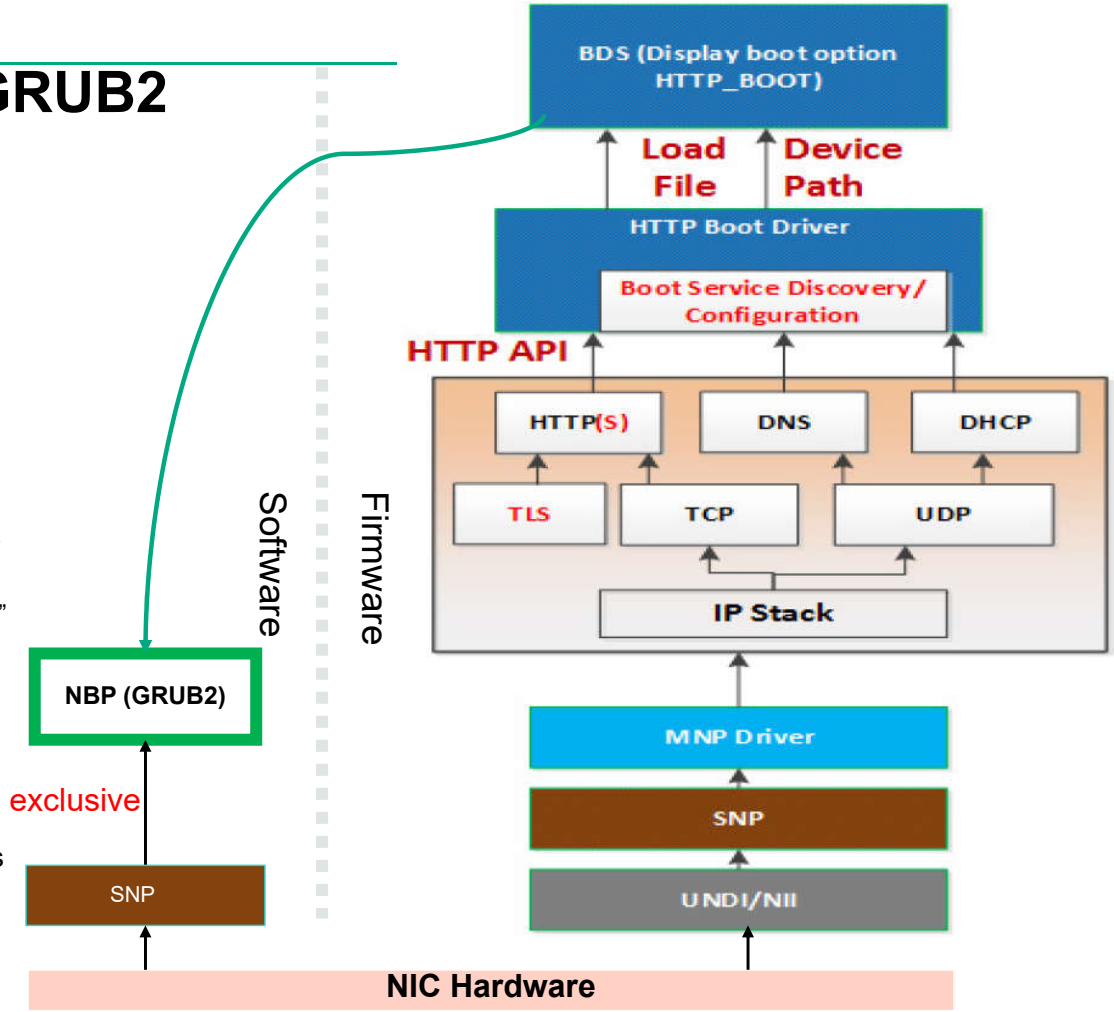
Parameters

GetModeData	Gets the current operational status. See the `GetModeData()` function description.
★ *Configure*	Initialize, change, or reset operational settings in the EFI HTTP protocol instance. See `Configure()` for function description.
★ *Request*	Queue a request token into the transmit queue. This function is a non-blocking operation. See `Request()` for function description.
Cancel	Abort a pending request or response operation. See `Cancel()` for function description.
★ *Response*	Queue a response token into the receive queue. This function is a non-blocking operation. See `Response()` for function description.
Poll	Poll to receive incoming HTTP response and transmit outgoing HTTP request. See `Poll()` for function description.

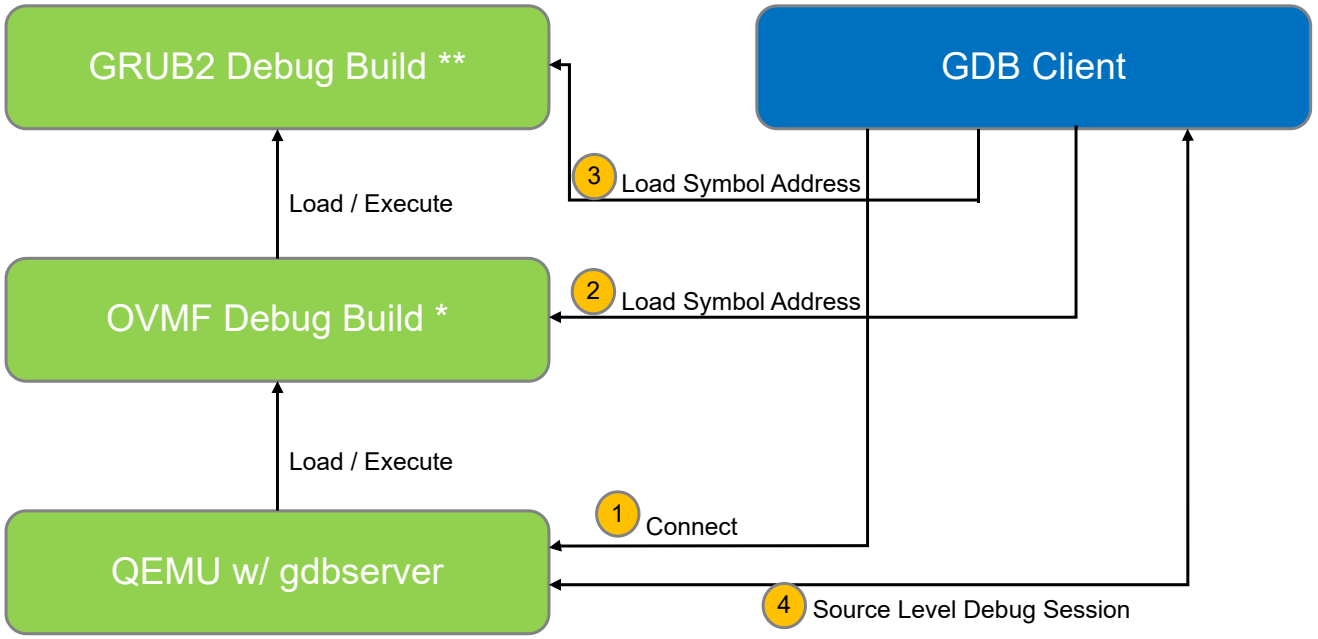
Obstacle of NIC Driver in GRUB2

- SNP & MNP
  - Simple Network Protocol, SNP (*UEFI 2.6 spec. 23.1*)
    - "This protocol can be used to as a building block in a full UDP and TCP/IP implementation that can produce a variety of application level network interface"
  - Managed Network Protocol, MNP (*UEFI 2.6 spec. 24.1*)
    - "MNP provides raw (unformatted) asynchronous network packet I/O services. The services make it possible for multiple-event-driven drivers and applications to access and use the system network interfaces at the same time."
- In short
  - `GRUB2` only implements SNP network driver
  - SNP has no "multiplex access" ability
  - HTTP(s) are an application-level protocols
  - If `GRUB2` and UEFI firmware issue HTTP requests

  at the same time, there could be *race conditions*



Interactive Source Level GRUB2 / OVMF Co-Debugging



* http://feishare.com/efimail/messages/20140619-0534- edk2 source level debugging of OVMF with gdb-Laszlo Ersek.html
** http://wjhowto.googlepages.com/grub2-r2300-2340-elfgdb.patch
** http://svnweb.mageia.org/packages/cauldron/grub2/current/SOURCES/10005-Make-a-gdb-dprintf-that-tells-us-load-addresses.patch

Summary

– Works on **HPE ProLiant Gen10** servers & `EDK2/OVMF + QEMU`

– **HPE** is the major contributor of UEFI HTTP(s) Boot in `EDK2`

– **HPE** is driving the support in `Linux bootloaders`

Hewlett Packard
Enterprise

Open Discussion

Slide downloadable from http://sched.co/AVBT