Wireshark OTG, Extend your Wireshark with extcap, iPad and Pi
---TIPS and tricks of extcap and make use of Wireshark everywhere, any capture sources
Megumi Takeshita, packet otaku

- Founder, ikeriri network service co., ltd
- Reseller of CACE technologies in 2008
- Worked SE/IS at BayNetwork, Nortel
- Wrote 10+ books about Wireshark
- Instruct Wireshark to JSDF and other company
- Reseller of packet capture / wireless tools
- One of contributors of Wireshark
- Translate Wireshark into Japanese
Session Details
Do you imagine your tablet can run Wireshark, Yes you can get Wireshark OTG. Megumi show you TIPS and tricks to use Wireshark with iPad Pro and Pi. You may not install many extcap interface that is not installed in default settings, It's time to make use of extcap interface such as sshdump. We can create our own customized extcap interface in easy way on Windows environment. Actual demonstration extend your Wireshark's extcap interface!!

Note
Megumi uses iPad Pro, Raspberry Pi and Windows10 environment. Linux bash and Windows command prompt programming skills help you understand the session well.
#1 Wireshark OTG
Bring your Wireshark without PC
Wireshark everywhere with iPad Pro

We need
- An iPad or other tablet
- A Raspberry pi 4 or zero
- A USB-C to USB-C cable transferring both power and data
Burn the latest raspberry pi os using official imager application to microSD card.

- We use official raspberry pi imager to install the latest OS

Using wired Ethernet connection to setup raspberry pi at the first

- I recommend to use Raspberry pi 4 because there are 1 RJ-45 as well as wireless lan (supports monitor mode) and many USB ports
Set raspberry pi as USB Gadget mode

- USB Gadget mode is a kind of USB OTG (On-The-Go).
- Your Pi works as USB host instead of USB devices.
- Note: Raspberry Pi zero and 4 support gadget mode.
  I recommend to choose Pi4 because Wired LAN, many USB port and fast
Edit /boot/config.txt and /cmdLine.txt

- Set up Pi as USB Gadget mode, that supports USB-C as power and network
- ls /boot To find boot option setting file /boot/config.txt and /boot/cmdLine.txt
- At the last line of /boot/config.txt, add dtoverlay=dwc2
- /boot/cmdLine.txt is a long one line file. We need to insert string after “rootwait quiet” Find “rootwait quiet” and insert string “modules-load=dwc2,g_ether”
SSH/DHCP server setting

- We need to set up Pi4 act as SSH/DHCP server
- “touch /boot/ssh” to create blank file for ssh login
- “nano /etc/modules” to open modules file and add “libcomposite” to define USB3 device.
- Install DHCP server “apt-install isc-dhcp-server” and add “denyinterfaces usb0”
- Install dnsmasq with “sudo apt-get install dnsmasq”
- Create /etc/dnsmasq.d/usb and edit
- Create /etc/network/interfaces.d/usb0 and edit
/etc/dnsmsq.d/usb

interface=usb0
dhcp-range=10.55.0.2,10.55.0.6,255.255.255.248,1h
dhcp-option=3
leasefile-ro

/etc/network/interfaces.d/usb0

auto usb0
allow-hotplug usb0
iface usb0 inet static
  address 10.55.0.1
  netmask 255.255.255.2
Create initialize script "/root/usb.sh" (1)

https://www.hardill.me.uk/wordpress/2019/11/02/pi4-usb-c-gadget/

#!/bin/bash

cd /sys/kernel/config/usb_gadget/
mkdir -p pi4

cd pi4

echo 0x1d6b > idVendor # Linux Foundation

echo 0x0104 > idProduct # Multifunction Composite Gadget

echo 0x0100 > bcdDevice # v1.0.0

echo 0x0200 > bcdUSB # USB2

echo 0xEF > bDeviceClass

echo 0x02 > bDeviceSubClass

echo 0x01 > bDeviceProtocol

mkdir -p strings/0x409/configuration

Create initialize script “/root/usb.sh” (2)

https://www.hardill.me.uk/wordpress/2019/11/02/pi4-usb-c-gadget/

```bash
echo 250 > configs/c.1/MaxPower
# see gadget configurations below
mkdir -p functions/ecm.usb0
HOST="00:dc:c8:f7:75:14" # "HostPC"
SELF="00:dd:dc:eb:6d:a1" # "BadUSB"
echo $HOST > functions/ecm.usb0/host_addr
echo $SELF > functions/ecm.usb0/dev_addr
ln -s functions/ecm.usb0 configs/c.1/
udevadm settle -t 5 || :
ls /sys/class/udc > UDC ifup usb0
service dnsmasq restart
```
Autorun USB initial script

- We want to start up USB gadget mode, DHCP Server and other service every time we start up Pi4
  
  There are nice initial script from “Ben’s Place”
  
  https://www.hardill.me.uk/wordpress/2019/11/02/pi4-usb-c-gadget/

- Make /root/usb.sh executable with chmod +x /root/usb.sh
  
  Add /root/usb.sh before exit 0

- Set this script every time we power on Pi4 so I use S01cron start script in /etc/rc3.d(CLI) and /etc/rc5.d (GUI)
“sudo halt” to power off raspberry pi,
change connection from SSH via wired LAN to USBC

○ “sudo halt” to power off raspberry pi
○ USE a USBC-USBC cable,
  connect between raspberry Pi4 and iPad Pro
○ Pi4 start up with USB Gadget mode and DHCP server
Install "ish" from AppStore and packages

- open AppStore to look for "ish" app (free)
- "Ish" is a command line shell of iPad OS
- Open "ish" app and install openssh packages
- "apk add openssh" and other packages if you need ssh pi@10.55.0.1 to login Raspberry Pi 4 via SSH
SSH connection via USB Gadget mode
install RDP server

- Connect raspberry pi via USB Gadget mode
  Pi uses 10.55.0.1, and iPad gets 10.55.0.2 from dhcpd
  ssh 10.55.0.1, and enter username and password
  pi/raspberry and confirm the connection via USBC

- Also recommend to install XRDP (Remote Desktop Protocol server) if there are not installed
  apt-get install xrdp

- Now you can use your Wireshark cli tools such as
  dumpcap, tshark, mergecap, editcap, capinfos etc.
Connect via Microsoft RDP client

- Install Microsoft RDP iOS client app
- Open 10.55.0.1 and login as the same as CLI
Bring your Wireshark everywhere

- Install Microsoft RDP client
- Create shortcut of 10.55.0.1 via USB-C
- Username pi
- Password raspberry
- Wireshark works at reasonable speed. (Pi4 with 8GB RAM)
Change Wireshark permission settings

- We need administrative privilege to capture packet, though Raspberrian OS works as user mode.
- We need to change Wireshark permission to be able to capture packets in user mode.
- "sudo dpkg-reconfigure wireshark-common"
- Choose YES to capture packets in user mode
- "sudo adduser wireshark pi" to add user pi into wireshark group
- Restart Raspberry Pi4 and login as pi again
Enjoy capturing wired and wireless network with full-featured Wireshark, tshark and so on

• Under 15 seconds, pi starts up
• Capturing wired network using Raspberry Pi4 RJ-45 port
For wireless capturing

- Capturing wireless network using external WiFi adapter that supports monitor mode (unfortunately Pi internal Wireless LAN card cannot be changed into monitor mode at default setting)
- You may use KaliLinux ARM image instead of Rapsberian.
- Open ish and check wireless card “sudo iwconfig”
- “sudo airmon-ng check kill” to stop all wireless related process
- “sudo airmon-ng start wlan1” to change into monitor mode
- ”iwconfig” again to check wlan1 was changed into wlan1mon (monitor mode interface)
- Choose “View>Wireless tool bar” to show wireless settings
#2 Extcap brings the external capture source

- The extcap interface is a versatile plugin interface that allows external binaries to act as capture interfaces directly in Wireshark.

- It is used in scenarios, where the source of the capture is not a traditional capture model (live capture from an interface, from a pipe, from a file, etc). The typical example is connecting esoteric hardware of some kind to the main Wireshark application and data.

- Extcaps may be any binary or script within the extcap directory. Please note, that scripts need to be executable without prefacing a script interpreter before the call.
Example extcap interface: SSH remote capture

- Let’s try sample extcap interface, SSH remote capture
- SSH remote capture is provided by SSHDump, is option component with Wireshark, so you need to check Tools>SSHDump dialog during Wireshark
Example extcap interface: SSH remote capture

- There are default “SSH remote capture” extcap in Wireshark (you may check Tools>SSH remote capture during install process)
- Double click extcap icon (left side) to edit option
Test SSH remote capture Extcap interface

1. Start Wireshark
2. Choose “SSH remote capture” interface
3. Click option icon
4. Set Remote SSH server address as some Linux host this time we use Raspberry Pi IP address 10.0.0.201
5. Set Remote SSH port number as 22 in the Server Tab
Test SSH remote capture Extcap interface

6. Click Authentication tab, enter Remote SSH server username (this time we use “pi”)

7. Enter Remote SSH server password (this time “raspberry”) Note: you may fail at the first time to connect to save the host’s public key as known host
Test SSH remote capture Extcap interface

8. Click Capture tab, enter Remote interface
   Enter Remote capture command (dumpcap -w -)
   (this time we use dumpcap command, output pcap
   not to file but to standard output "-w -" option)

9. Check “Use sudo on the remote machine”
   Note Remote capture filter is set automatically
Test SSH remote capture Extcap interface

11. You can set debug file in case of failure check “Run in debug mode” and set path in “Use a file for debug” text box

![Image of Wireshark interface with debug settings]
Test SSH remote capture Extcap interface

12. Click Start to capture packet at remote SSH host

13. You can get the trace at the Pi’s side remote LAN interface
Then let’s test your own extcap interface

- Extcap is useful so you can extend capture source.
- There are some nice hardware, such as Bluetooth dongle and open source capture devices to capture via extcap interface.
- At first, let’s test your own extcap interface.  
- We do not need to create binary, but just a bit of batch file to test extcap interface
Check your extcap path
Help>About Wireshark>Folder

- Personal Extcap Path in Windows Default
  C:¥Users¥user¥AppData¥Roaming¥Wireshark¥extcap

- Global Extcap Path in Windows Default
  C:¥Program Files¥Wireshark¥extcap
<table>
<thead>
<tr>
<th>elements</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arg (options)</td>
<td>argument for CLI calling</td>
</tr>
<tr>
<td>number</td>
<td>Reference # of argument for other values, display order</td>
</tr>
<tr>
<td>call</td>
<td>Literal argument to call (--call=...)</td>
</tr>
<tr>
<td>display</td>
<td>Displayed name</td>
</tr>
<tr>
<td>default</td>
<td>Default value, in proper form for type</td>
</tr>
<tr>
<td>range</td>
<td>Range of valid values for UI checking (min,max) in proper form</td>
</tr>
<tr>
<td>type</td>
<td>Argument type for UI filtering for raw, or UI type for selector:</td>
</tr>
<tr>
<td></td>
<td>integer</td>
</tr>
<tr>
<td></td>
<td>unsigned</td>
</tr>
<tr>
<td></td>
<td>long (may include scientific / special notation)</td>
</tr>
<tr>
<td></td>
<td>float</td>
</tr>
<tr>
<td></td>
<td>selector (display selector table, all values as strings)</td>
</tr>
<tr>
<td></td>
<td>boolean (display checkbox)</td>
</tr>
<tr>
<td></td>
<td>radio (display group of radio buttons with provided values, all values as strings)</td>
</tr>
<tr>
<td></td>
<td>fileselect (display a dialog to select a file from the filesystem, value as string)</td>
</tr>
<tr>
<td></td>
<td>multichck (display a textbox for selecting multiple options, values as strings)</td>
</tr>
<tr>
<td></td>
<td>password (display a textbox with masked text)</td>
</tr>
<tr>
<td></td>
<td>timestamp (display a calendar)</td>
</tr>
<tr>
<td>value (options)</td>
<td>Values for argument selection</td>
</tr>
<tr>
<td></td>
<td>arg Argument # this value applies to</td>
</tr>
</tbody>
</table>
Create example.bat and copy into Personal Extcap Path

```bat
echo interface {value=test}{display=Test Extcap Interface}
echo dlt {number=147}{name=test}{display=Layer2 DLT}
echo arg {number=1}{call=--host}{display=Filter Hostname}{type=string}{tooltip=hostname}{required=true}{default=10.0.0.201}{group=Host}
echo arg {number=2}{call=--port}{display=Filter Port number}{type=unsigned}{tooltip=port}{range=1,65535}{default=22}{group=Port}
```
Open Wireshark and Capture>Option to check Extcap Interface

interface {value=test} {display=Test Extcap Interface}

You can find your Extcap interface at Interface list in your Wireshark startup screen

dlt {number=147}{name=test}{display=Layer2 DLT}

Select Capture>Option and look the Link layer header Column to check your extcap data link type value
Click option button ( ) to find your extcap dialog

You can create your Extcap Option GUI using script
Set number to set reference number and display order
Set call to call function (this time do nothing)
Set display to set the display name
Set type to set the type definition (this time is string)
Set tooltip to set tooltip string
Set required to set this value is necessary
Set default to set default value
Set group to set the tab name

```
echo arg {number=1}{call=--host}{display=Filter Hostname}{type=string}{tooltip=host}{required=true}{default=10.0.0.201}{group=Host}
```
Click option button ( ) to find your extcap dialog(2)

You can create your Extcap Option GUI using script

```
echo arg {number=2}{call=--port}{display=Filter Port number}{type=unsigned}{tooltip=port}{range=1,65535}{default=22}{group=Port}
```

Set number to set reference number and display order
Set call to call function (this time do nothing)
Set display to set the display name
Set type to set type definition (this time is unsigned)
Set tooltip to set tooltip string
Set range to set the range of the value (this time is from 1 to 65535)
Set default to set default value
Set group to set the tab name
Then check SSHDump command

- Open Global Extcap Path in explorer in Windows Default C:¥Program Files¥Wireshark¥extcap
- Check sshdump.exe is in Global Extcap path
- Open command prompt and execute sshdump.exe
sshdump.exe command

- Check online help of sshdump.exe

C:\Program Files\Wireshark\extcap>sshdump.exe

Wireshark - sshdump.exe v1.0.0

Usage:
sshdump.exe --extcap-interfaces
sshdump.exe --extcap-interface=sshdump.exe --extcap-dlts
sshdump.exe --extcap-interface=sshdump.exe --extcap-config
sshdump.exe --extcap-interface=sshdump.exe --remote-host myhost --remote-port 22222 --remote-username myuser --remote-interface eth2 --remote-capture-command 'tcpdump -U -i eth0 -w' --fifo=FILENAME --capture

- There are many options for sshdump
sshdump.exe options

--extcap-interfaces: list the extcap Interfaces
--extcap-dlts: list the DLTs
--extcap-interface <iface>: specify the extcap interface
--extcap-config: list the additional configuration for an int
--capture: run the capture
--extcap-capture-filter <filter>: the capture filter
--fifo <file>: dump data to file or fifo
--extcap-version: print tool version
--debug: print additional messages
--debug-file: print debug messages to file
--help: print this help
--version: print the version
--remote-host <host>: the remote SSH host
--remote-port <port>: the remote SSH port
--remote-username <username>: the remote SSH username
--remote-password <password>: the remote SSH password.
If not specified, ssh-agent and ssh-key are used

--sshkey <public key path>: the path of the ssh key
--sshkey-passphrase <public key passphrase>: the passphrase to unlock public ssh

--proxycommand <proxy command>: the command to use as proxy the the ssh connection
--remote-interface <iface>: the remote capture interface
--remote-capture-command <capture command>: the remote capture command
--remote-sudo: use sudo on the remote machine to capture
--remote-noprom: don't use promiscuous mode on the remote machine
--remote-filter <filter>: a filter for remote capture (default: don't listen on local interfaces IPs)
--remote-count <count>: the number of packets to capture
Check debug.txt created by sshdump.exe

- We make use of sshdump.exe to create our own extcap interface
sshdump.exe command

- we need to create the command like below

"C:\Program Files\Wireshark\extcap\sshdump.exe" -capture
--extcap-interface sshdump.exe --fifo %fifo% --remote-host 10.0.0.201 --remote-port 22 --remote-password raspberry
--debug --remote-sudo --remote-capture-command
"dumpcap -P -w -" --debug-file
C:\Users\TakeshitaMegumi\Desktop\debug.txt
--remote-username pi --remote-interface eth0 true

So let’s create example2.bat file to create your own extcap interface to make use of SSHDump
example2.bat (initialization)

set "capture=0"
set "extcap_interfaces=0"
set "extcap_interface="
set "extcap_dlts=0"
set "fifo="

Initialization of command variables such as capture, extcap_interfaces, extcap_interface, extcap_dlts, fifo

Flag 0:off 1:on
example2.bat (parse)

:parse
REM check command line parameters
if "%~1"=="" goto :main
if /i "%~1"=="--capture" set "capture=1" & shift & goto :parse
if /i "%~1"=="--extcap-interfaces" set "extcap_interfaces=1" & shift & goto :parse
if /i "%~1"=="--extcap-interface" set "extcap_interface=%~2" & shift & shift & goto :parse
if /i "%~1"=="--fifo" set "fifo=%~2" & shift & shift & goto :parse
if /i "%~1"=="--extcap-dlts" set "extcap_dlts=1" & shift & goto :parse
shift
goto :parse

Check command line parameters, and if the parameter matches the option, set the flag as 1, use shift to adjust parameter and jump parse again (shift decrease the position of the parameter and save.)

%~1 is the first parameter without quotes, %~2 is the second.

If there are no parameter, jump to the main function.
example2.bat (main)

:main

REM – Process request for interface list from Wireshark
if ”%extcap_interfaces%”==”1″ call :extcap_interface_func & goto :end

REM – Process request for dlts list from Wireshark
if ”%extcap_dlts%”==”1″ call :extcap_dlts_func & goto :end

REM – Process capture request
if ”%capture%”==”1″ call :capture_func & goto :end

exit /B 1

Main function check each flags and jump at corresponding labels

Check command line parameters, and if the parameter matches the option, set the flag as 1 and jump parse again
example2.bat (extcap_interface_func)

:extcap_interface_func

echo interface {value=test2}{display=Capture from Pi}

exit /B 0

Show extcap interface as the request for interface list from Wireshark and exit

example2.bat (extcap_dlts_func)

:extcap_dlts_func

echo dlt {number=147}{name=test2}{display=Layer2 DLT}

exit /B 0

Show datalink header type as the request for dlts from Wireshark and exit
example2.bat (capture_func)

:capture_func
"C:\Program Files\Wireshark\extcap\sshdump.exe"
   --capture --extcap-interface sshdump.exe --fifo %fifo%
   --remote-host 10.0.0.201 --remote-port 22
   --remote-password raspberry --debug --remote-sudo
   --remote-capture-command "dumpcap -P -w -"
   --debug-file C:\Users\TakeshitaMegumi\Desktop\debug.txt
   --remote-username pi --remote-interface eth0 true
exit /B

Call sshdump.exe with adequate parameters

example2.bat (end)

:end

exit /B

“exit /B” do not terminate the command but just quit preserving variables.
Let’s try your customized extcap interface

1. Copy example2.bat to the personal extcap path (C:¥Users¥username¥AppData¥Roaming¥Wireshark¥extcap)

2. Close and open Wireshark and check your own extcap interafaces ( Capture from Pi )

3. Double click “Capture from Pi” and get the trace
Let’s try your customized extcap interface

Just a double click of your own extcap interface, you can get traces from remote host using Wireshark
Reference: Windows 10 Wireshark Plug-in for WLANPi Wireless Captures (wlan-extcap-win)

◉ Instead of batch file programming, you may also use nice batch file, wlan-extcap-win by wifinigel
https://github.com/wifinigel/wlan-extcap-win

◉ It is a nice batch file script based on Adrian Granados' original Python scripts on the wlan-extcap project (macOS)

◉ We can use Raspberry Pi’s monitor mode Wi-Fi interface as one of extcap interfaces to make use of SSHDump
Download wlanpidump.bat and save to Extcap path

◉ Download batch file “wlanpidump.bat” and save to personal Extcap path (C:¥Users¥user¥AppData¥Roaming¥Wireshark¥extcap)

◉ Edit wlanpidump.bat and find “set capture_cmd” section

```
set capture_cmd="%kill_old_instances_cmd% %time_cmd% %if_down% %set_monitor% %if_up% sudo /usr/sbin/iw %remote_interface% set channel %remote_channel% %remote_channel_width% > /dev/null && /usr/sbin/tcpdump -i %remote_interface% %filter_statement% -s %frame_slice% -U -w - "

call "%sshdump_path%" --extcap-interface sshdump --remote-host %host% --remote-port %port% --remote-capture-command %capture_cmd% --remote-username %username% --remote-password %password% --fifo %fifo% --capture
```

The batch file create GUI of wireless settings, set parameters, set wireless interface as monitor mode, create capture command and call SSHDdump to capture wireless packet via ssh connection from Raspberry Pi
Use wlan-extcap-win extcap interface

1. Close and open Wireshark again
2. You can find WLAN Pi extcap interface
3. Click Option icon
4. Choose Channel and Channel Width in Capture TAB (this time we use Channel 1 and 20MHz bandwidth)
5. Set Remote host IP address and Port in WLAN Pi Address and WLAN Pi Port fields in Server tab (used for SSHDump parameter)

6. Set Remote host username and password in Authentication tab
Use wlan-extcap-win extcap interface

7. Set remote interface, capture filter (option), Frame Slice, and Sync WLANPi Time option in advanced tap (we use wlan1 as remote interface, no capture filter, capture all frames using 0 as Frame Slice and disabled Sync WLANPi Time)

8. Click Start to capture wireless packet via Raspberry Pi
USE WIRESHARK

Thank you for watching!!

Please complete the SharkFest Europe app-based survey

Supplemental file
http://www.ikeriri.ne.jp/sharkfest

ikeriri network service
http://www.ikeriri.ne.jp