Politecnico di Torino

Protocolli e Architetture di Routing

Capture and Parsing of HTTP Packets

Fulvio Risso

May 1, 2013

Introduction

Given a personal computer equipped with a packet capture library, write a program in C language that:

- Captures all the packets generated and received by the host
- Writes, per each packet, a single line on screen reporting the following information (in case some information are not available, such as the PORT in case of a packet that is neither UDP nor TCP, please leave the field blank):

  `timestamp  MAC_src -> MAC_dst  IP_src -> IP_dst  Protocol  PORTsrc -> PORTdst`

- Check if the TCP the destination port of the packet is equal to '80'; in this case:
  - Check if the packet contains an HTTP request (e.g., a `POST/GET` command)
  - In this case, extract the URL contained in the packet (e.g., `http://www.cnn.com`) and print it on screen, after the data mentioned before.

Hints

Capture library

In order to capture the packets on your host, you need to have a packet capture library. e.g., either `libpcap` on Unix or `WinPcap` on Windows. However, if you want to develop some software based on this library, you need to install also the development libraries, e.g. `libpcap-dev` on Linux or the `WinPcap Developer's Pack` on Windows.

Please check the the instructions related to your operating system in order to install those packages. For instance, Linux lists them in its software repository, while in Windows you have to download (and install) both packages (the run-time library and the Developer's Pack) from `http://www.winpcap.org`.

Documentation

A rather complete documentation of the capture library (either `libpcap` on `WinPcap`) is available on the WinPcap website, `http://www.winpcap.org`, including programming samples. Since `libpcap` and `WinPcap` share the same API, the documentation that you can find on the WinPcap website applies also to `libpcap`, excluding some OS-specific topics such as how to compile/link your software under different operating systems.

If you operate on Windows, please note that the WinPcap Developer's Pack includes also some working examples, complete with source and project files (for Microsoft Visual Studio). It is strongly suggested to start with those files in order to avoid compilation/linking issues (e.g., due to required files located in the wrong folder).

Reading the packet data

The packet capture library exports a set of primitives that allow the user software to receive the full packet, as it is received by the network interface card. This data is formatted as a plain buffer; you need to know the format of each protocol in order to parse the packet and check what is written inside.

Please refer to the proper documentation (e.g., RFCs) for the protocol you need; a brief summary is available at the following website: `http://www.networksorcery.com/`.

Byte ordering

Please note that the information contained in the packet buffer is written in *network byte order* (which is *big-endian*), while Intel machines work the opposite way (*little-endian*). Therefore, all the fields that need to be read as numbers (e.g., TCP/UDP ports) need to be translated in the proper byte order before being able to operate on them (e.g., checking their value). In this case, it is stongly suggested to use the functions `ntoh()` (available in the C standard library) in order to convert numbers in the right format.

More information on byte ordering is available at `http://en.wikipedia.org/wiki/Endiannes`.