Open Access License Notice

Dissecting Industrial Control Systems Protocol for Deep Packet Inspection

Abstract - *The nation's critical infrastructures, such as those found in industrial control systems (ICS), are increasingly at risk and vulnerable to internal and external threats. One of the traditional ways of controlling external threats is through a network device called a firewall. However, given that the payload for controlling the ICS is usually encapsulated in other protocols, the tendency is for the firewall to allow packets that appear to be innocuous. These seemingly harmless packets can be carriers for sinister attacks that are buried deep into the payload. The purpose of this paper is to present the different ICS protocol header signatures for the purpose of devising deep packet inspection strategies that can be implemented in network firewalls.*

Keywords: Industrial Control Systems (ICS), Network Protocols, Deep Packet Inspection, Firewall, Intrusion Prevention, SCADA.

1 Introduction

The nation's critical infrastructures, such as those found in industrial control systems (ICS), are increasingly at risk and vulnerable to internal and external threats. One of the traditional ways of controlling external threats is through a network device called a firewall. However, given that the payload for controlling the ICS is usually encapsulated in other protocols, the tendency is for the firewall to allow packets that appear to be innocuous. These seemingly harmless packets can be carriers for sinister attacks that are buried deep into the payload. Thus, defense-in-depth techniques, such as Deep Packet Inspection (DPI), are needed to counter these potentially damaging activities. Our goal is to provide the reader with a deeper understanding of ICS protocol packets which could be useful in packet analysis and in the development of tools for DPI.

The rest of the paper is organized into four parts. First, we present a review of industrial control system protocols and detailed depiction of their network frames. Second, we cover various devices and systems for network packet filtering. In the third section, we illustrate packet dissection by analyzing a handful of industrial network protocol packets that were captured by an open source network packet analysis tool. Finally, we conclude our paper by presenting possible research avenues that can be pursued as extensions to this seminal work.

2 Industrial Control System Protocols

Industrial control system protocols range from wired to wireless. Wired protocols include Ethernet/IP, Modbus, Modbus/TCP, Distributed Network Protocol version3 (DNP3), PROFIBUS, CANOpen, and DeviceNet. The wireless variety include WirelessHART, 802.15 (Bluetooth), 802.16 (Broadband) and Zigbee. Some of these protocols are briefly described in the following subsections.

2.1 DNP3

The Distributed Network Protocol Version 3 (DNP 3.0) is a protocol standard to define communications between Remote Terminal Units (RTU), master stations, and Intelligent Electronic Devices (IEDs). It was originally a proprietary model developed by Harris Controls Division and designed for SCADA systems. DNP 3.0 is an open protocol standard and is an accepted standard by the electric, oil & gas, waste/water, and security industries [Clarke, Reynders & Wright, 2004].

DNP 3.0 is a four-layer subset of the OSI 7 layer model. The layers are the application, data link, physical, and pseudo-transport layers. The pseudo-transport layer includes routing, flow control of data packets, and transport functions such as error-correction and disassembly and assembly of packets. The transport layer takes an APDU (application protocol data unit) from the application layer and breaks it down into smaller units of TPDU (transport protocol data unit), which consists of a one byte header followed by a maximum of 249 bytes of data. The header contains a bit that identifies the start of the sequence of TPDU frames and another bit that identifies the end. Following these two bits is a six-bit sequence counter. The Link Protocol Data Unit (LPDU), also known “DNP3 Frame” has a maximum limit of 292 bytes [Clarke, Reynders, & Wright, 2004]. The DNP3 link layer frame is shown in Figure 1.

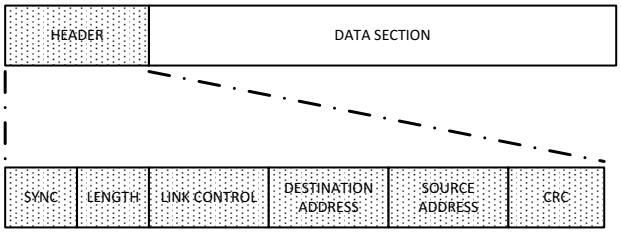

Figure 1. The DNP3 Link Layer Frame

2.2 CAN

Controller area network (CAN) protocols operate at the physical and the data link layer of the OSI model. The protocol supports at most 110 nodes on a half-duplex network and is based on the Ethernet Carrier Sense Multiple Access with Collision Detection (CSMA/CD) model. Because the specific transmission times across the network cannot be guaranteed, CAN provides transmission priorities using arbitration on message priority (AMP) with CSMA/CD to compensate for this issue [Krutz, 2006].

The CSMA/CD + AMP scheme allows for priority rating to be included in a message. The identifier in the message determines the message's priority, which has a length of 11 bits. A CAN data frame consists of a start of frame, an arbitration field, a control field, a data field, a CRC field, an ACK field, and an end of frame field [CAN in Automation, 2013].

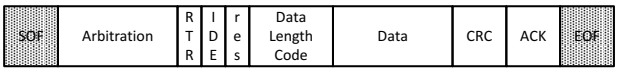

Figure 2. A Standard CAN Data Frame

In the standard CAN data frame depicted in Figure 2, SOF is the start of frame, followed by the 11-bit Arbitration field, a 1-bit Remote Transmission Request (RTR), a 1-bit Identifier Extension (IDE) a 1-bit reserve flag, a 4-bit Data Length Code (DLC), 0-8 Byte Data payload, a 15-bit Cyclic Redundancy Check (CRC), a 1-bit CRC delimiter, a 2-bit Acknowledgement (ACK) field (first bit for slot and the second bit for the delimiter), and finally, the seven "recessive" bits called the End of Frame (EOF). In data communication, a "recessive" bit has the logical value 1.

2.3 Modbus

Perhaps the most widely deployed ICS protocol is the Modbus. It is based on the master/slave principle, where transactions can be a query/response type, where only a single slave addressed, or a broadcast/no response type where all slaves is addressed. Data in transmission can be of two modes: American Standard Code for Information Interchange (ASCII) or Remote Terminal Unit (RTU). The selection mode defines the bit contents of the message fields which are transmitted serially and determines how the data will packed and decoded [Modicon, 2000].

The functions which the Modbus protocol support are listed below [Clarke, Reynders, & Wright, 2004]:

- Coil control commands for reading and setting a single coil or a group of coils
- Input control commands for reading input status of a group of inputs
- Register control commands for reading and setting one or more holding registers
- Diagnostic tests and report functions
- Program functions
- Polling control functions
- Reset

The Modbus protocol specifies the Protocol Data Unit (PDU), which is comprised by the function code and the data field, and is independent of the underlying communication layers. Additional fields such as the address and error-check fields augment the PDU to complete the application Data Unit (ADU) [Modbus.org, 2012]. The generic Modbus frame (ADU) is depicted in Figure 3.

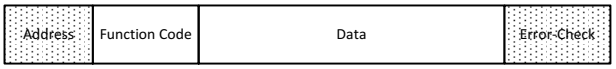

Figure 3. Generic Modbus Frame

The single octet Address field identifies the controller/device to which the request/response is being directed. The single octet function code can be any code from one of these three categories: Public, User-defined, and Reserved function codes. The data field contains the information that is being requested, the exception code, or the information, such as the addresses and number of registers, that is being passed to the server. The Error-Check field contains the Longitudinal Redundancy Check (LRC) information for the ASCII mode and for the RTU mode, the Cyclic Redundancy Check (CRC) information.

2.4 Modbus TCP

The Modbus organization, Modbus.org, extended the Modbus protocol to work over the Transmission Control Protocol (TCP) by encapsulating the Modbus PDU with the Modbus TCP ADU [Thomas, 2008]. This protocol is registered to utilize port 502 and is realized by augmenting the standard Modbus PDU with a Modbus Application Protocol (MBAP) header as shown in Figure 4.

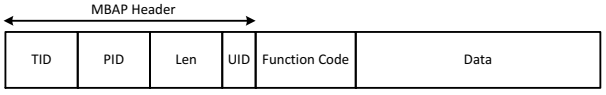

Figure 4. Modbus TCP ADU

The MBAP Header is made up the following: a two-octet Transaction Identifier (TID), a two-octet Protocol Identifier (PID), a two-octet total length (Len) in bytes of the remaining fields, and an octet indicating the Unit Identifier (UID) which identifies the remote slave connected by a serial line [Modbus.org, 2012b].

2.5 Profibus

Process Fieldbus (Profibus) is an open fieldbus serial network standard, and is mainly used for real-time control applications. The protocol operates on the application, data link, and physical layers of the OSI model. Profibus comes in three forms: Profibus Process Automation (PA), Decentralized Peripherals (DP), and Profibus Fieldbus Message Specification (FMS) [Krutz, 2006].

Profibus Process Automation (PA) uses a common serial bus to connect both data acquisition devices and control devices. This implementation gives this form intrinsic safety and reliability features. This form also provides power to devices through the bus.

The *Profibus Decentralized Peripherals (DP)* provides high-speed communication between Programmable Logic Controllers (PLCs) in a decentralized environment.

The *Profibus Fieldbus Message Specification(FMS)* supports a large number of applications. It is also used for general automation and for average transmission rates.

The *Profibus DP* telegram header (11 bytes) and data field (variable length—maximum of 244 bytes) is depicted in Figure 5. The header consists of the start delimiter (SD), the net data length (LE), the length repeated (LEr), destination address (DA), source address (SA), function code (FC), the destination service access point (DSAP), the source service access point (SSAP), the frame checking sequence (FCS), and the end delimiter (ED) [Acromag, 2002].

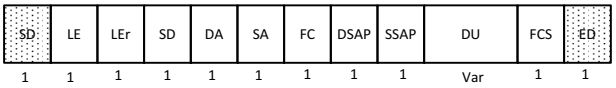

Figure 5. Profibus-DP (Message) Telegram Structure

2.6 ControlNet

ControlNet is a token-passing bus control network protocol that is based on the IEEE 802.4 standard. The nodes in the token bus network are configured into a ring topology, and in particular, in ControlNet, each node knows the address of the preceding and succeeding nodes [Lian, Moyne, & Tilbury, 2001]. The ControlNet protocol became a part of the Common Industrial Protocol (CIP) family of protocols in 1997.

The Medium Access Control (MAC) frame format transmitted on ControlNet is shown in Figure 6 [ODVA, 2006]. The transmitted data, which could be as many as 510 bytes, is carried by a series of link packets (LPackets). A link packet can either be a Fixed Tag or a Generic Tag, each of which is shown in Figures 7 and 8, respectively. The Fixed Tag LPackets are used for Unconnected Messaging and network administration while the Generic Tag LPackets are used for Connected Messaging [ODVA, 2006]. The link data field occupies 506 bytes in the fixed tag and 505 bytes in the generic tag.

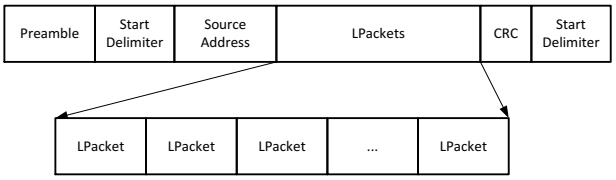

Figure 6. ControlNet MAC Frame Format

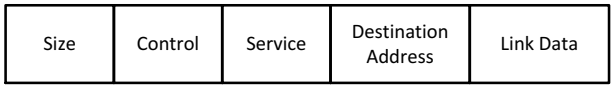

Figure 7. Fixed Tag LPacket Format

Figure 8. Generic Tag LPacket Format

2.7 EtherNet/IP

Another member of the CIP protocol family is the EtherNet/Industrial Protocol or EtherNet/IP. This protocol runs over TCP/IP or UDP/IP. TCP/IP uses the reserved port **0xAF12** for transmitting Explicit messages while UDP/IP uses the reserved port **0x08AE** for transmitting I/O messages. A typical Ethernet frame with the encapsulated EtherNet/IP data is shown in Figure 9 [ODVA, 2006].

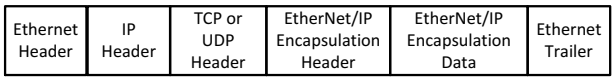

Figure 9. Ethernet Frame with Encapsulated EtherNet/IP

The encapsulation header and data fields are shown in Figure 10. The two-octet command (Cmd) field represents various types of commands such as broadcast, session opening and closing, and receiving and sending data for connected and unconnected messaging.

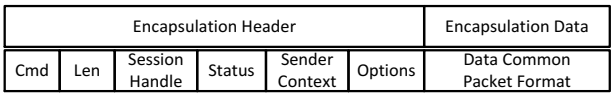

Figure 10. Encapsulation Packet Format

2.8 EtherCAT

EtherCAT stands for Ethernet for Control Automation Technology. It is a control system protocol that is mainly used to satisfy very high performance requirements in the manufacturing environment. For a typical 1000 distributed points the update time is approximately 30 microseconds [Digital Bonds, 2013]. The format of an EtherCAT UDP frame is shown in Figure 11. The EtherCAT telegram consist of one or more datagrams, each serving a particular memory area of the logical process images of up to 4 GB in size. The EtherCAT frame header consists of a length field, a reserve field, and a type field which indicates the nature of the data carried by the EtherCAT telegram [EtherCAT, 2013].

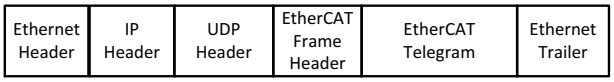

Figure 11. EtherCAT UDP Frame

3 Network Filters

Network filters found in devices such as routers, gateways, bridges, firewalls, and intrusion prevention systems are the first line of defense against malicious packets. In this section, we examine the different techniques that are used to realize their filtering functionalities. We also introduce the concept of deep packet inspection to usher our on-going work on ICS protocol packet dissection.

3.1 Filtering Techniques

Deep packet inspection is small part of the filtering techniques that are adopted by security providers in their commercial products. A non-exhaustive list, which is shown below, is presented in [Franz & Pothamsetty, 2004].

- Layer 2 filtering using intelligent switches/bridges;
- Access Control Lists on Layer 3 and 4 using routers;
- Stateful Firewall filtering;
- Application Proxy filtering; and
- Deploying DPI and Intrusion Preventing Systems.

In describing an implementation of a Modbus firewall for DPI, Franz & Pothamsetty showed some simple firewall rules [Franz & Pothamsetty, 2004]:

 # iptables –A INPUT –p tcp –m modbus –funccode 8 –allowtcp 1 –j DROP
 # iptables –A input –p tcp –m modbus –funccode !16 –allowtcp 1 –j DROP

The first rule drops a Modbus packet with a diagnostic function code; the second rule drops a Modbus packet whose function code is NOT a write multiple registers (16). The rules are simple examples of filtering Modbus packets but are not very useful in practice. Obviously there is a great need for developing filters that are creatively constructed using DPI and intelligent mechanisms. These filters can then be applied on fields and values in control systems. These fields might include register read and write commands, controlled objects, and service requests [Byres, 2012].

3.2 Deep Packet Inspection(DPI)

Deep Packet Inspection is the process of allowing packet inspecting devices, such as firewalls and Intrusion Prevention Systems (IPS), to perform an in-depth analysis of packet contents. This in-depth analysis is much broader than common technologies in that it combines protocol anomaly detection and signature scanning to realize its potential [Ramos, 2009]. The interested reader is referred to Antonello, et al. (2012) for a comprehensive literature review on various tools and techniques for the development of DPI systems.

4 ICS Protocol Packet Dissection

In this section, we present snapshots of captured industrial protocol packets that are encapsulated by Ethernet. We believe that, due to the ubiquity of the adaptation of industrial protocol on this networking technology, network attacks on industrial control equipment will most likely come with this encapsulation.

A cursory analysis of each packet is presented with the aim of stimulating the interest of the reader in exploring the other ICS protocol packets that are not shown in this paper. Further, a packet dissection using freely accessible packet capture and analysis software such as Wireshark [Wireshark, 2013] or Capsa Network Analyzer [Colasoft, 2013] provides tremendous boost towards securing industrial networks. Foremost among the derived benefits are 1) the facilitation of deep packet inspection; 2) the enhancement of data collection for predictive analytics; and 3) the provision of a deeper understanding of the protocol architecture and behavior.

4.1 Modbus/TCP Packet

A sample captured Modbus/TCP packet dissection is shown in Figure 12. Note how perfect the Modbus and Modbus/TCP segments map directly with the APDU fields shown in Figure 4. With this information at hand, the DPI designer can now judiciously put forth more

sensible IPS or firewall rules that are tuned to the secure operations of the ICS. As an additional note, after a quick perusal on how the protocol is assembled, a trained security analyst can easily recognize a security vulnerability: the lack of an authentication mechanism.

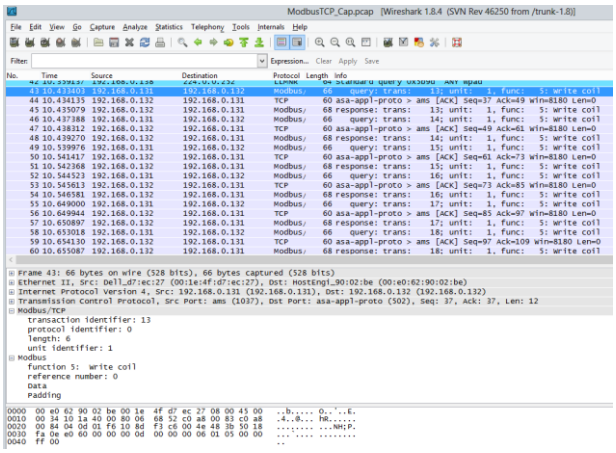

Figure 12. Sample Modbus/TCP Packet

4.2 EtherNet/IP Packet

A sample captured EtherNet/IP packet dissection is shown in Figure 13. A perusal of the packet dissection shows the encapsulated EtherNet/IP header and the command specific data fields. In a similar vein as what is described above, the information gathered here can be used for intelligent monitoring and securing an EtherNet/IP enable device.

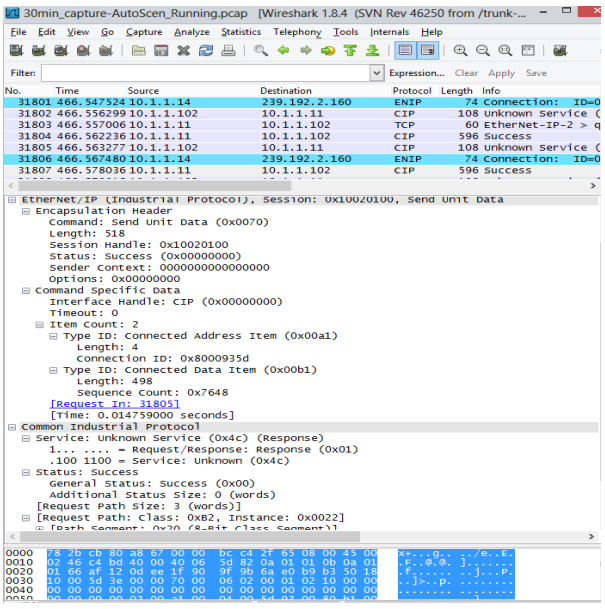

Figure 13. Sample EtherNet/IP Packet

4.3 EtherCAT Packet

A sample captured EtherCAT packet dissection is shown in Figure 14. Again, it can easily be seen the mapping of the dissected fields to those shown in Figure 11. The EtherCAT frame header is carrying “command” types of data that are defined in each of the datagrams (telegrams).

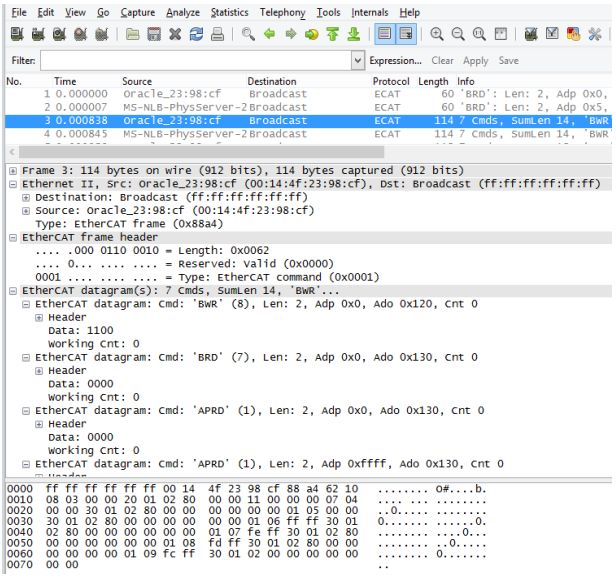

Figure 14. Sample EtherCAT Packet

5 Conclusions and Future Plans

This paper presented a review of industrial control protocols and the need for an in-depth defense mechanism in form of a Deep Packet Inspection system. We also presented our on-going work on the dissection of various ICS packets as a springboard for the development of a DPI system dedicated to the protection of industrial controls. Further, we have to emphasize the fact that a solid understanding of the ICS protocols and architectures is critical to securing these systems. This seminal work on scrutinizing ICS packets is a small contribution to that end.

The challenge for the authors will be in the continual development of the DPI for control systems. Future plans include:

- Development of a proof-of-concept Linux based DPI; and
- Expansion of the ICS frame signatures to include those that utilizes the wireless network protocols.

6 Acknowledgements

This paper is based upon a project partly supported by the National Science Foundation under grant award

XXXXX-XX. Opinions expressed are those of the authors and not necessarily of the Foundation.

7 References

Acromag Incorporated. (2002). "Introduction to Profibus DP." Busworks 900PB Series. Profibus/RS485 Network I/O Modules. Technical Reference. Retrieved: http://www.diit.unict.it/users/scava/dispense/II/Profibus.pdf. March 13, 2013.

Antonello, R., et al. (2012), Deep Packet Inspection Tool and Techniques in Commodity Platforms: Challenges and Trends." *Journal of Network and Computer Applications*, November 2012, 35(6):1863-1878. Elsevier, Ltd.

Byres, Eric (2012). "Understanding Deep Packet Inspection for SCADA Security." White paper--Tofino Security. December 12, 2012. Retrieved January 10, 2013 from http://www.tofinosecurity.com.

CAN in Automation (2013). *CAN Specification 2.0, Part B.* Retrieved March 08, 2013 from CAN in Automation: http://www.can-cia.org/fileadmin/cia/specifications/CAN20B.pdf

Clarke, G., Reynders, D., & Wright, E. (2004). *Practical Modern SCADA Protocols.* Oxford: Elsevier Ltd.

ColaSoft (2013). "Capsa Network Analyzer." Retrieved February 28, 2013 from http://www.colasoft.com.

Digital Bond (2013). "EtherCAT". Retrieved March 14, 2013 from http://www.digitalbond.com/scadapedia/protocols/ethercat/.

Franz, M. & Pothamsetty, V. (2004), "ModbusFW Deep Packet Inspection for Industrial Ethernet,' Cisco Systems. Retrieved March 01, 2013 from http://blogfranz.googlecode.com/files/franz-niscc-modbusfw-may04.pdf.

EtherCAT Technology Group (2013). "EtherCAT-the Ethernet Fieldbus Technical Introduction and Overview". Retrieved March 15, 2013 from http://www.ethercat.org/en/technology.html#3.1.

Krutz, R. L. (2006). *Securing SCADA Systems.* Indianapolis: Wiley.

Lian, F., Moyne, J., & Tilbury, D. "Performance Evaluation of Control Networks: Ethernet, ControlNet, and DeviceNet" *IEEE Control Systems Magazine*, February, 2001. Pp. 66-83.

Modbus.org (2012). "Modbus Application Protocol Specification" v1.1b3. Retrieved March 10, 2013 from http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.

Modbus.org (2012b). "Modbus Messaging on TCP/IP Implementation Guide" v1.0b. Retrieved March 10, 2013 from http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf.

Modicon (2000). "Modbus Protocol" v1.1b3. Retrieved March 13, 2013 from http://irtfweb.ifa.hawaii.edu/~smokey2/software/about/sixnet/modbus/modbus_protocol.pdf.

Open DeviceNet Vendor Association, Inc. (ODVA) (2006). "The Common Industrial Protocol (CIP) and the Family of CIP Networks". Retrieved March 15, 2013 from http://www.odva.org/portals/0/library/publications_numbered/pub00123r0_common%20industrial_protocol_and_family_of_cip_netw.pdf. Ann Arbor, MI.

Ramos, Anderson (2009), "Deep Packet Inspection Technologies," in *Information Security Management Handbook*, Harold Tipton & Micki Krause, eds. 6th Edition Vol 3. Auerbach Publications, NY.

Thomas, George (2008). "Introduction to Modbus Serial and Modbus TCP," The Extension—A Technical Supplement to Control Network. Contemporary Control Systems, Inc. Retrieved March 01, 2013 from http://www.ccontrols.com/pdf/Extv9n5.pdf.

Wireshark (2013). "Wireshark." Retrieved February 28, 2013 from http://www.wireshark.org.