Proceedings of the 14th Colloquium for Information Systems Security Education
Baltimore Marriott Inner Harbor
Baltimore, Maryland June 7 - 9, 2010

TCP Three-way Handshake as a Pedagogical Tool

Yu "Andy" Wu, *University of North Texas*

Abstract – The TCP three-way handshake can be used as a pedagogical tool when teaching network security in an introductory course in information security. I use it as a common theme that runs through various network security topics so that it is easier for students to grasp new concepts while reinforcing old knowledge. This paper describes the rationale for so doing and shares examples of some learning tools I created in this respect.

Index terms – TCP three-way handshake, sniffing, packet capture, port scanning, Wireshark, Nmap, security lab

I. Introduction

A typical introductory course in information security includes both (a) the necessary coverage of basic security principles and (b) more technically oriented topics such as network security threats and malware. The instructor may also enhance the conceptual contents with some hands-on labs. Thus, managerial and human behavioral topics, the theoretical portion of technical contents, and hands-on exercises all compete for the limited time available in a semester. The time and content coverage that can be devoted to network security may be quite scarce. On the other hand, network security is such an important foundation for understanding other areas in security that student mastery of its basic concepts should be one key objective of the course. Therefore, an instructor often faces the challenge – How do I select and design network security contents effectively and economically to optimize student learning of network security with limited time and coverage?

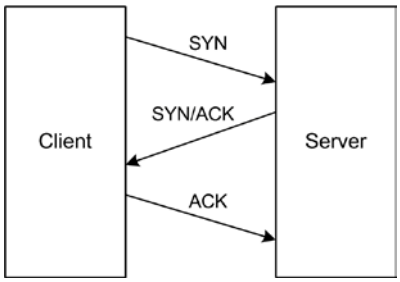
I personally find the TCP three-way handshake an interesting and significant networking concept with high pedagogical value. The TCP three-way handshake is the process by which two hosts establish a TCP connection for communication. It is much abused by a number of information security threats, including port scanning (if used for nefarious purposes) and distributed denial-of-service (DDoS) attacks. From the perspective of information security education, the handshake can serve as an effective pedagogical tool to organize and illustrate a number of key concepts in network security with a clear theme and focus. As the result of the repeated, focused coverage, student knowledge is reinforced effectively. Based on my experience teaching introductory security courses at two large public universities in southeastern and southwestern U.S., this common thread effectively leverages the limited time for maximum mastery. In this paper I describe the rationale for using the handshake as a pedagogical tool and provide related examples such as lab exercises and exam questions.

II. Handshake and Network Security

The TCP three-way handshake can be utilized as a common theme that runs through the networking portion of an information security course, from the basics of the TCP protocol to network defense mechanisms such as firewalls and intrusion detection systems (IDSes). Below, I introduce the handshake briefly and then describe how it plays a role in five subject areas in network security: TCP protocol, sniffing, port scanning, distributed denial of service (DDoS), and network defense.

A. TCP Three-way Handshake

The TCP three-way handshake is a process by which two hosts establish a connection-oriented TCP session.



The requester for information (the client) initializes session establishment by sending a synchronization (SYN) packet to the provider of information (the server). In TCP jargon, "synchronization" means to start communication. Upon receipt of this packet, the server returns a packet that serves two purposes: (a) acknowledging (with the ACK flag) the receipt of the SYN packet from the client; and (b) indicating (with the SYN flag) the server's intention to synchronize with the client as well. The client responds to this SYN/ACK packet from the server by acknowledging it with its own ACK packet. In principle, each host must send a SYN packet and receive an ACK packet to establish the connection. However, since the acknowledgment and synchronization request coming from the server are combined in one packet for efficiency, this process in

effect takes three instead of four packets, hence the name "TCP three-way handshake." [1-2].

Once the handshake finishes, data can be transmitted between the client and the server. Each subsequent packet will bear only the ACK flag, but not the SYN flag, the latter being used only for synchronization and appearing only during the handshake stage [3].

B. Handshake and TCP Protocol

Among the seven layers in the OSI model, Layer 3 and Layer 4 are of particular relevance to an introductory course in information security. Students need to acquire working knowledge of the TCP protocol so that they can understand how security attacks and defense devices, such as firewalls and IDSes, work. To achieve this, an instructor typically introduces the concept of port numbers and how services use them. I found it beneficial to also add coverage of TCP headers and how the fields are used in TCP flow control and error correction, e.g., forward acknowledgment. The TCP handshake proves to be an excellent instructional apparatus for this because it uses two of the control bits (SYN and ACK). Complemented by my later coverage of port scanning (see below for details) and sometimes, the four-way process of tearing down a connection, students will eventually see five out of the six flags (except PSH) in action. The three-way handshake also provides a vivid example of how SEQ numbers are created (in the second and third steps) and how ACK numbers are used to acknowledge receipt of packets (in the second and third packets).

C. Handshake and Network Sniffing

Performing network sniffing is an essential skill for security administrators and attackers alike. A trend observed over the recent years is that Wireshark has become the preferred sniffing tool for many. Wireshark [4] features a familiar graphic user interface (GUI) and therefore is an excellent choice for novices and students, especially those majoring in information systems rather than computer science. An instructor needs to bring students up to speed with two essential skills – packet capture and analysis. The former is relatively easy, thanks to the highly user-friendly GUI of Wireshark. The challenge lies more in the latter. How can we develop student skills in navigating the GUI, drilling down to the right level of details, and finding and analyzing the right information?

Even capturing a visit to a simple page like the Google home page can generate quite a number of packets. This can become overwhelming for students. Therefore, the exercise usually involves a simple task such as capturing ping packet. However, analysis of ping packets only deals with Layer 3 information. For students to understand Layer 4 header fields, TCP three-way handshake is a handy sample. Handshake traffic is simple, has clear starting and ending points, but provides rich examples of the use of flags and SEQ/ACK numbers. Use of handshake traffic, therefore, controls the scope of both capture and analysis so that student will find the tasks manageable.

D. Handshake and Network Scanning

Another essential network security skill is port scanning. Nmap [5] is the most widely used port scanner [6]. The basic strategy is for the scanning host ("scanner" hereinafter) to repetitively initiate connections (sending SYN packets) to the target host, each to a different port. Whereas the scanner is not sincere in using connections for data transmission, the target will faithfully respond with a SYN/ACK packet, if a port is open and no firewall sits between the two hosts. A reset (RST) response or absence of responses from a port signifies to the scanner that the port is closed or filtered by a firewall. In this way, the scanner analyzes the replies from the server to determine the status of the scanned ports [6]. This is a blatant abuse of the TCP handshake because the scanner capitalizes on the rules of the protocol (Requests for Comments, or RFCs), knowing that an unprotected target is bound to respond in predictable ways despite the scanner's insincerity to communicate.

There are a number of scan types to choose from. In a connection scan, the scanner actually finishes the connection with the target, although it does not exchange any other data with it. A connection scan has the advantage of being more accurate but is more susceptible to detection. In a SYN scan, the scanner does not complete the handshake with the target. If a SYN/ACK packet is received from the target, the scanner simply terminates the premature handshake by sending an RST packet to the target. A SYN scan is not as easy to be detected, will not be logged, but is not as accurate as a connection scan.

Capturing Nmap traffic therefore makes a good learning tool for both understanding how port scanning works and how TCP handshake works and can be abused.

E. Handshake and DDoS Attack

DDoS attack can be pulled off by abusing the TCP handshake. During the handshake, once a server replies to a SYN packet with a SYN/ACK, the connection is said to be "half-open." In addition, the server allocates to this half-open connection some memory buffer, which will be tied up until the client replies with an ACK packet or resets the connection. If nothing is heard from the client,

the half-open connection will be closed after a time-out period and the resources freed. However, during a SYN flood DDoS attack, the attacker repetitively initiates an exorbitant number of requests for connection. For each request, the attacker simply ignores the SYN/ACK packet from the server, hence holding the connection half open for as long as possible. This trick is done with such a quantity and frequency that the rate at which new requests are sent largely outpaces the rate at which these half-baked connections time out. As a result, the resources on the server are exhausted to the extent that the server is unable to respond to legitimate requests for connection.

Similar to port scanning, the SYN flood attack abuses the handshake process, knowing that the server cannot distinguish legitimate from malicious SYN packets [7].

F. Handshake and Network Defense

Two of the major perimeter security mechanisms in use today are firewalls and IDSes. Both mechanisms can be implemented with stateless and stateful variations. A stateless firewall or IDS simply matches certain fields in the packet headers to predefined rules. For example, a stateless firewall or IDS assumes that any packet with the ACK flag is part of an existing connection. In reality, such a packet can be generated by a port scanner performing an "ACK scan" or "Windows scan" [3, 5]. Without the knowledge of any previous three-way handshake, it is basically impossible for a stateless mechanism to determine the intention of the ACK packet: Is it a packet carrying legitimate data after a handshake is finished and a connection established? Or is it an ACK scan packet? An ACK scan can be more difficult to filter than a SYN scan [5].

A stateful inspection mechanism, in contrast, remembers the state of all connections. It maintains a historical record of all packets related to an active connection until the connection terminates. When it receives a packet, it looks up its state table and decides which connection the packet belongs to. Continuing with the previous example, a stateful inspection firewall will be able to tell whether a packet with ACK flag is part of an established connection or comes "out of the blue" without a pre-established three-way handshake [3, 5]. If the latter is the case, it will not let the packet through. Similarly, the "stream4" preprocessor in Snort allows stateful inspection and generates alerts on port scan packets that are not preceded by three-way handshakes.

III. Sample Instructional Materials

As the above section shows, the TCP three-way handshake is relevant to a good variety of network security areas. Given the limited time that can be allocated to network security in a typical introductory course, I use it as a strategic pedagogical tool. Whenever a network security topic is covered and the handshake can be brought in as an example, I will do that. Thus, students will see a familiar concept multiple times from various new perspectives. This not only reduces students' anxiety with the new topic, but the repetition also helps student to reinforce old knowledge.

In addition to PowerPoint slides, the main learning tools I created are lab exercises with open-ended questions, handouts on packet analysis, diagram problems, and exam questions. For each of the above, students also have access to suggested answers so that they can assess their mastery of the topics.

A. Lab Exercises

For labs, synergy can be achieved by generating handshake traffic with port scanning task. The following are the steps from one of the labs I created:

(1) Start Wireshark.
(2) Select Capture | Interfaces…
(3) Locate the correct network interface. Click "Options." Then select the capture filter that you created in Part 1 of Lab 3 for only capturing traffic to and fro this computer (*In a previous lab the student captured ping traffic and created a filter to capture only traffic coming into and going out of the local virtual machine*).
(4) Do not click the "Start" button yet.
(5) Start Zenmap.
(6) Inside the "Command" textbox, replace the existing command, if any, with this command text: `nmap -sT -v -p 131-150 172.19.204.1xx`. The "Target" and "Profile" textboxes will be cleared automatically once you type over the default command.
(7) Go back to Wireshark and click the "Start" button.
(8) Move back to Zenmap and click the "Scan" button.
(9) Once Nmap scan finishes, wait a few seconds and then stop capture in Wireshark.
(10) Save the capture as `Lab03-Nmap-ConnScan.cap`.

These steps capture the traffic to scan 20 ports. By using the `-sT` and –p switches this way, I limit the total number of packets generated to be around 60 (normally, only three packets are generated for each port). This is a connection scan and the port range includes two of the Windows NetBIOS ports (135 and 139) that usually are open in a Windows networking environment. As the result, it is certain that students will obtain at least two TCP handshakes in the capture file.

To develop students' skills in analyzing the packets, after asking students to report Nmap results, I further instruct the students to open the capture file, apply the display filter: `tcp.port == 139`, and then answer the question:

Has a three-way handshake been established between the two machines? Why or why not? Support your answer with the screenshot AND analyses of SYN and ACK bits and SEQ and ACK numbers.

To answer the second half of the question correctly, students must expand the nodes in the middle pane of the Wireshark GUI (the "Protocol Tree" pane) until the SEQ and SYN numbers and the control bits are visible. They then must report in their answer the relationships they discover among those numbers and control bits. To accomplish this they often have to refer to the textbook, my handouts, and PowerPoint slides from my lectures to review the rules governing the handshake and TCP forward acknowledgment. This learning process solidifies student knowledge of TCP handshake.

By default, the "Enable transport name resolution" option is turned on in Wireshark. To make the analysis exercise more illustrative, I require students to turn off the option before they start the analysis. As the result, the port numbers are displayed in numeric form rather than translated into service names. Similarly, the "Relative sequence numbers and window scaling" option is turned on by default and Wireshark in fact performs SEQ-ACK number correlation for the user. By requiring students to turn it off before analysis, the SEQ and ACK numbers are in their original, full format, forcing the students to manually correlate the numbers.

If I want to make the analysis more challenging, in an alternative version of this lab, I do not provide the display filter `tcp.port == 139` to the students. Instead, I ask them to go though the captured packets and locate at least one TCP handshake. For students with higher skill levels, the alternative version proves to be an effective exercise.

The above lab steps guide students through capturing packets involved in a connection scan. For students to learn the variation in port scanning techniques, the lab continues with a SYN-scan task:

> Repeat Steps (6) through (10) in Section 1.1 above but make these minor changes:
> Nmap scan command: `nmap -sS -v -p 131-150 172.19.204.1xx`
> Save capture as: `Lab03-Nmap-SYNScan.cap`

Students are then required to repeat the analysis work on this set of captured packets and answer the same question above. The main difference they are expected to find is that in a SYN scan, the three-way handshake does not complete because in the scanner's reply, the RST bit instead of the ACK bit is turned on.

After performing these two sets of capture and analysis tasks, students have had the opportunity to conduct sniffing at least twice and to analyze two sets of packets, one comprising a full handshake while the other an aborted handshake. After the lab is graded, they also can download my solution file and read suggested answers (see Appendix I for an example). This process of repetition and contrast reinforces students' understanding of scanning and of the TCP handshake.

As a further exercise, an alternative version of the lab requires the student to perform yet another scanning task, a Xmas scan. Students then are required to identify packets used to scan a port and, in the Protocol Tree pane, drill down to the sufficient level of detail to report what control bits are turned on. This exercise not only improves students' skill in scanning and packet capture and analysis, but also prevents a misconception of the Xmas scan that can be seen in some publications. Only three control bits are turned on in a Xmas scan, not the six that some authors describe without checking [e.g., 8]. This exercise enhances understanding of the TCP protocol because additional control bits besides SYN and ACK are used.

B. Packet Analysis Handout

I created a 25-page handout [9] that shows the use of Windump and Wireshark to capture FTP and HTTP traffic. It contains a section analyzing the handshake traffic captured by Windump. This enhances learning by using a novel presentation format (Windump versus the familiar Wireshark). The following is an excerpt on the handshake:

> *One thing characteristic of TCP is the three-way handshake that must be done to establish a connection between two hosts, before data transfer can begin. The handshake uses "forward acknowledgement" in that the receiving host acknowledges its receipt of a packet not by returning the SEQ number of the packet last received. Instead, it sends out the SEQ number of the next packet it expects to receive. This will be the number that appears in the ACK number field in the header. In the case of three-way handshake, the ACK number is simply the SEQ number of the received packet plus 1 (for packets containing data, the calculation is more complicated; refer to the PowerPoint slides for our networking class). … The three-way handshake between the client and the Microsoft FTP server (207.46.133.140) is carried out as follows (Packets 5 through 7):*

```
14:37:41.255022 IP (tos 0x0, ttl 128, id
10061, offset 0, flags [DF], length: 48)
yvr2kpro.1324 > 207.46.133.140.21: S [tcp
sum ok] 4027325053:4027325053(0) win 65535
<mss 1460,nop,nop,sackOK>
      0x0000:  4500 0030 274d 4000 8006
0650 ac13 cc5c  E..0'M@....P...\
      0x0010:  cf2e 858c 052c 0015 f00c
1a7d 0000 0000  .....,.....}....
      0x0020:  7002 ffff a629 0000 0204
05b4 0101 0402  p....).........

14:37:41.381176 IP (tos 0x0, ttl  51, id
7642, offset 0, flags [none], length: 48)
207.46.133.140.21 > yvr2kpro.1324: S [tcp
sum ok] 3950319304:3950319304(0) ack
4027325054 win 16384 <mss
1460,nop,nop,sackOK>
      0x0000:  4500 0030 1dda 0000 3306
9cc3 cf2e 858c  E..0....3.......
      0x0010:  ac13 cc5c 0015 052c eb75
16c8 f00c 1a7e  ...\...,.u.....~
      0x0020:  7012 4000 63da 0000 0204
05b4 0101 0402  p.@.c...........

14:37:41.381514 IP (tos 0x0, ttl 128, id
10062, offset 0, flags [DF], length: 40)
yvr2kpro.1324 > 207.46.133.140.21: . [tcp
sum ok] ack 3950319305 win 65535
      0x0000:  4500 0028 274e 4000 8006
0657 ac13 cc5c  E..('N@....W...\
      0x0010:  cf2e 858c 052c 0015 f00c
1a7e eb75 16c9  .....,.....~.u..
      0x0020:  5010 ffff d09e 0000
P.......
```

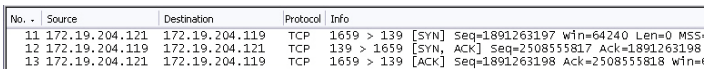
C. Diagram Problems

Two versions of a diagram problem test students' understanding of the handshake process. The one shown in Appendix II is the simpler version. Student should be able to indicate the direction of the packets, show which control bit(s) are turned on in each packet, and calculate the SEQ and ACK numbers involved. In a more challenging version, students first must answer the conceptual question – *What is the TCP three-way handshake*? They then are required to use one sentence to describe what each step accomplishes and to indicate the flags that are on. Also, they have to come up with their own fictitious SEQ and ACK numbers. Each semester I use one of the two versions based on the skill level of the students in that particular semester. The results have been fairly satisfactory. Also, I created a similar problem on the four-step process of terminating a connection and used it as a bonus question.

D. Exam Questions

In my exams, the three-way handshake may appear as the basis for packet analysis or Nmap questions. There are also questions that straightforwardly ask students to identify a handshake. Two examples are shown below.

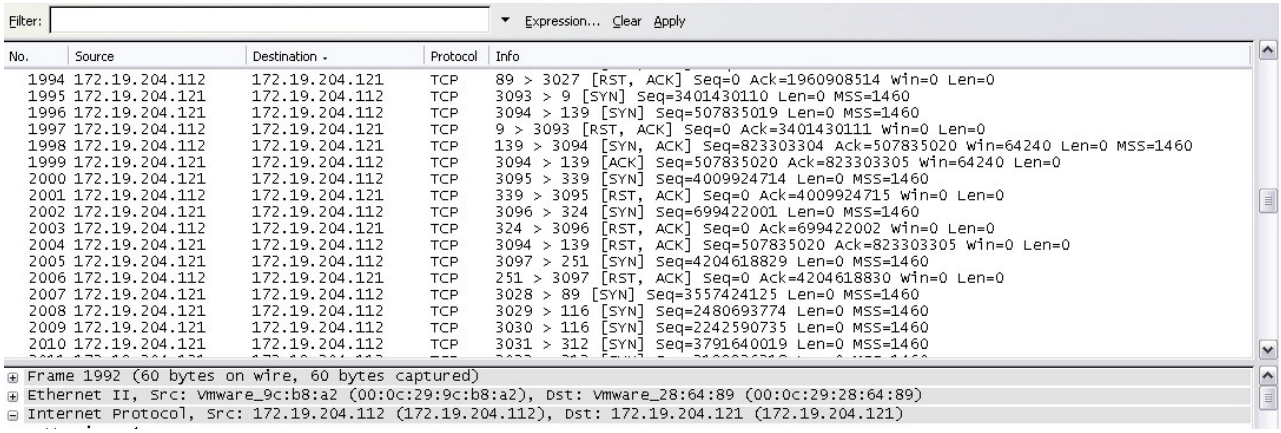
Question 1: If an Nmap scan was performed from



`172.19.204.121` and generated the following three packets when it scanned an open port:
These packets must be part of the results from running:

A. `Nmap -sT 172.19.204.121`
B. `Nmap -sS 172.19.204.121`
C. `Nmap -sT 172.19.204.119`
D. `Nmap -sS 172.19.204.119`

Correct answer: C



Question 2: "In the above figure, packets 1996, 1998, and 1999 completes a three-way handshake on Port 139"? This statement is true or false?

Correct answer: True

IV. Conclusion

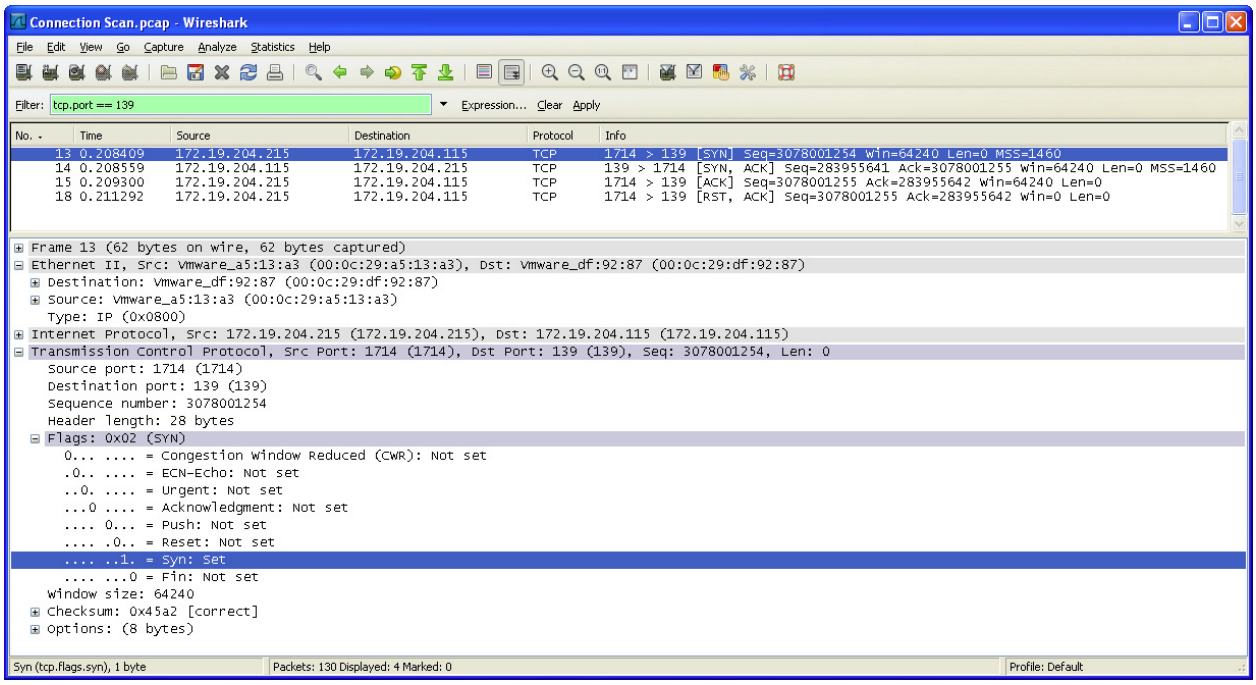
Since the TCP three-way handshake is involved in a number of subject areas in network security, instructional tools that center on this process promote synergy among various topics and make better use of limited time allocated to network security in a typical introductory course in information security.

V. References

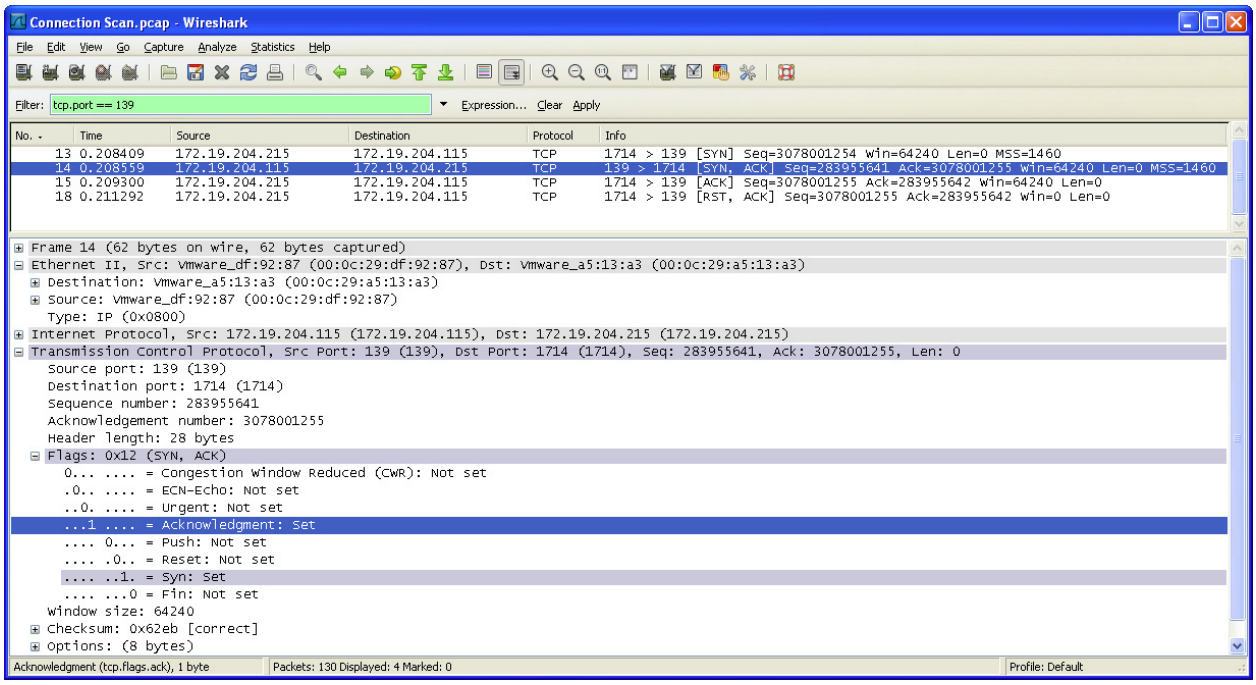
[1] C. M. Kozierok, *The TCP/IP Guide*, San Francisco, CA: No Starch Press, 2005.
[2] T. W. Ogletree, *Upgrading and Repairing Networks*, Boston, MA: Que, 2003.
[3] A. R. Baker, and J. Esler, *Snort IDS and IPS Toolkit*, Rockland, MA: Syngress Publishing, Inc., 2007.
[4] A. Orebaugh, *Wireshark and Ethereal Network Protocol Analyzer Toolkit*, Rockland, MA: Syngress Publishing, Inc., 2007.
[5] Fyodor, *Nmap Network Scanning*, Sunnyvale, CA: Insecure.com, LLC, 2008.
[6] M. Gregg, *Hack the Stack*, Rockland, MA: Syngress Publishing, Inc., 2006.
[7] J. Mirkovic, S. Dietrich, D. Dittrich *et al.*, *Internet Denial of Service: Attack and Defense Mechanisms*, Upper Saddle River, NJ: Prentice Hall, 2005.
[8] T. Howlett, *Open Source Security Tools*, Upper Saddle River, NJ: Prentice Hall, 2005.
[9] Y. Wu, *Sniffing and Ethereal*, unpublished teaching notes, 2006.

Appendix I. Sample Answer to Lab Questions

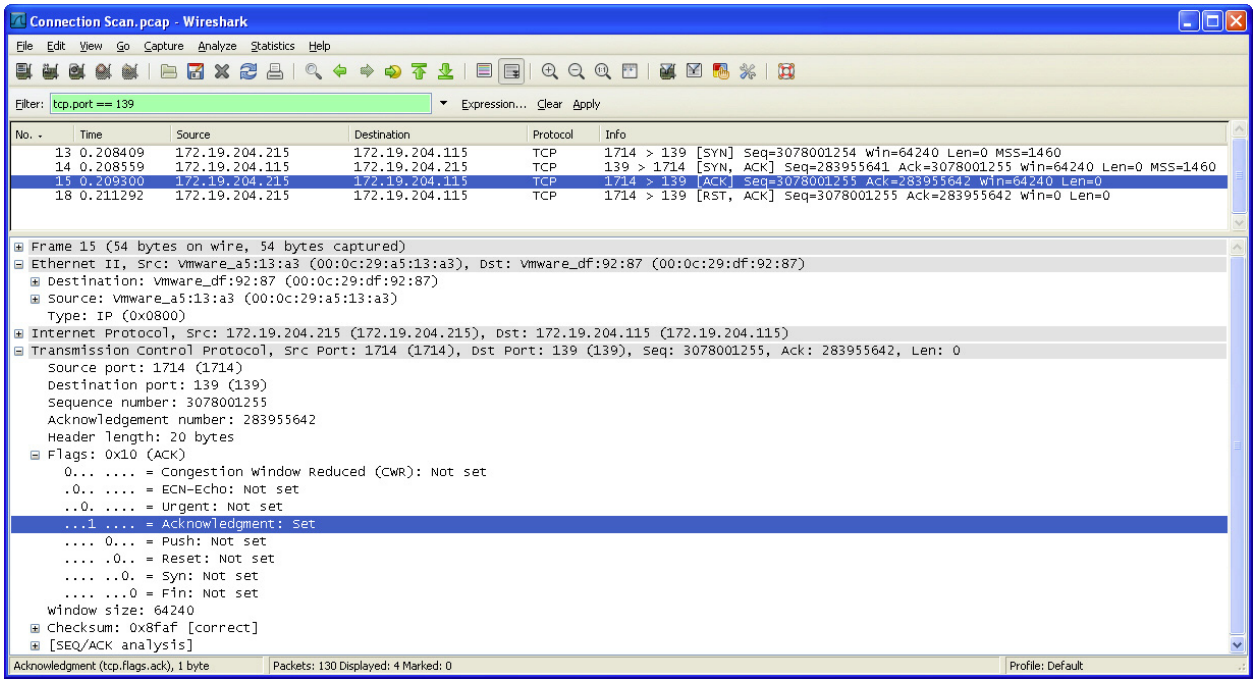
We know the following packet (#13) is the first packet because the SYN flag in the TCP header is set. This packet goes from the scanner host (172.19.204.215) to the target (172.19.204.115). The destination port, of course, is 139 because this is one of the ports the scanner wants to scan. As for the port from which this packet goes out, in Nmap scans the scanner randomly opens up a high-number port for the scanning job. For this scan, Port 1714 is the source port. The ISN is 3078001254. There is no ACK number because this is the first packet in the handshake.



The next packet is the response from the target machine. As the second packet in the handshake, both the ACK and SYN flags are set. The ACK bit is on because this is how the target host acknowledges the receipt of the first packet. In addition, the target host increments the scanner's ISN by 1 and then uses it as the ACK number (3078001255). Since both parties in a handshake have to explicitly express its intention to communicate, this same packet also doubles as the indication of the target host's such intention. Therefore, just like in the first packet, the SYN bit is turned on and the SEQ number field shows the target host's ISN – 283955641. This packet goes from Port 139 on the target to Port 1714 on the scanner.



The third packet is the scanner host's reply to the second packet. Therefore, only the ACK bit is turned on. The ACK number is the target host's ISN + 1, namely 283955642. The SEQ number of this packet is 3078001255, because the target host's ACK in the second packet in effect says, "I have received packet #3078001254. Now please send me packet #3078001255".



APPENDIX II. SAMPLE DIAGRAM PROBLEM

The following diagram illustrates the TCP three-way handshake. Currently the contents of the first packet is partially shown. It has a sequence number of 1234001 and goes from Host A to Host B. Please complete the diagram by filling in correct information, which includes:

Circling the arrow that indicates the correct *direction a packet is being sent*;

If a control bit should be turned on in a particular packet, marking the value that should appear in that bit;

Indicating the correct *sequence and/or acknowledgment numbers* (you may come up with fictitious numbers if necessary but the arithmetic relationships between the numbers should be correct).

Please be exact. You may lose points if you mark extraneous bits/numbers excessively.