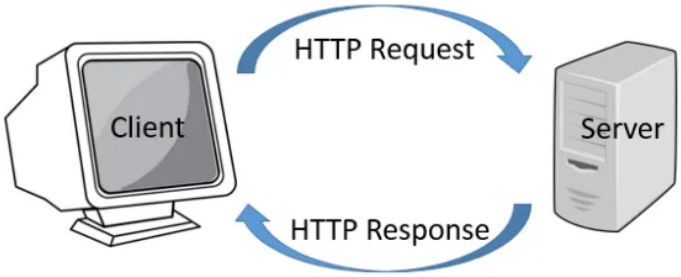
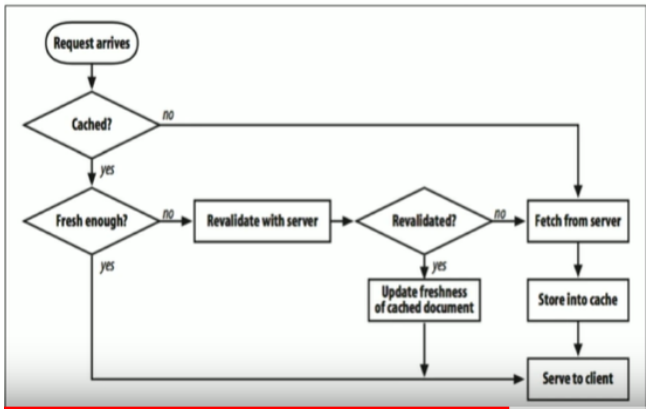
1. A title indicating what application-layer protocol is being used.

The application layer protocol being used is HTTP.

2. 1-2 paragraphs explaining how that protocol works and what it is used for. Include images and diagrams as necessary. Explain the steps taken in the communication of the protocol. - Include an image of the back-and-forth communication the protocol did in the pcap to help illustrate your explanation.

The HTTP protocol is a protocol that helps with actually getting data from/to client and server. In an analogy to the real world HTTP is kind of like a car that travels on TCP roads. The roads are confirmed and exist and therefore HTTP can travel on them to perform one of the four methods: put, get, post, and delete.





I will explain these diagrams briefly and if more information is needed let me know. The general idea is that the client sends HTTP request and server send HTTP response and communication takes place. This is beyond the transport layer where something like TCP already established a connection with a handshake. The HTTP protocol is actually sending and retrieving what you want from the server. An example of this would be an HTTP..GET “picture of a kitten” is an HTTP request from the client to the server to send a “picture of a kitten”. The example we are working with had nothing to do with kittens but you can see and POST and GET between ports 1989 and 80. This has to do with the content discussed below.

The diagram on the right might be a little bit deeper than what you’re looking for but from what I understand the request arrives and then is checked to see if the requested “item” is cached. That means it asks if it is stored closer or if it has to go to the main source. If it’s cached it can be checked for freshness, revalidated/updated if needed and then sent to the client.

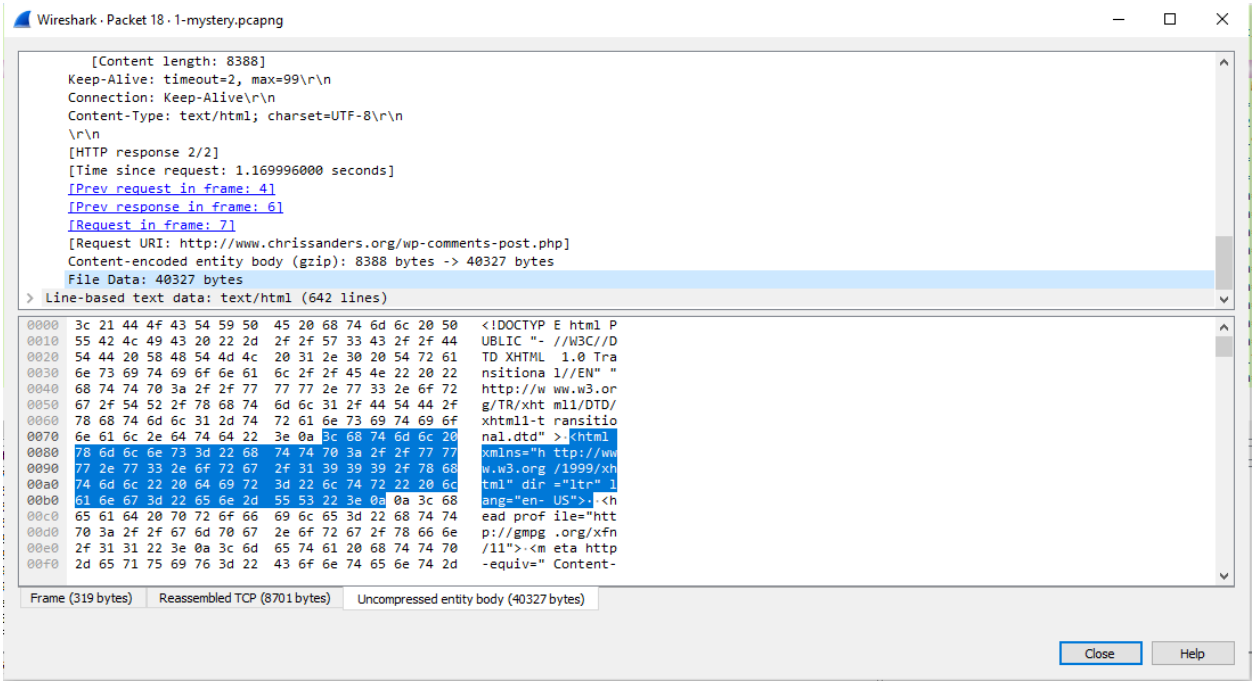
3. 2-3 paragraphs explaining what happened in the communication across the network.

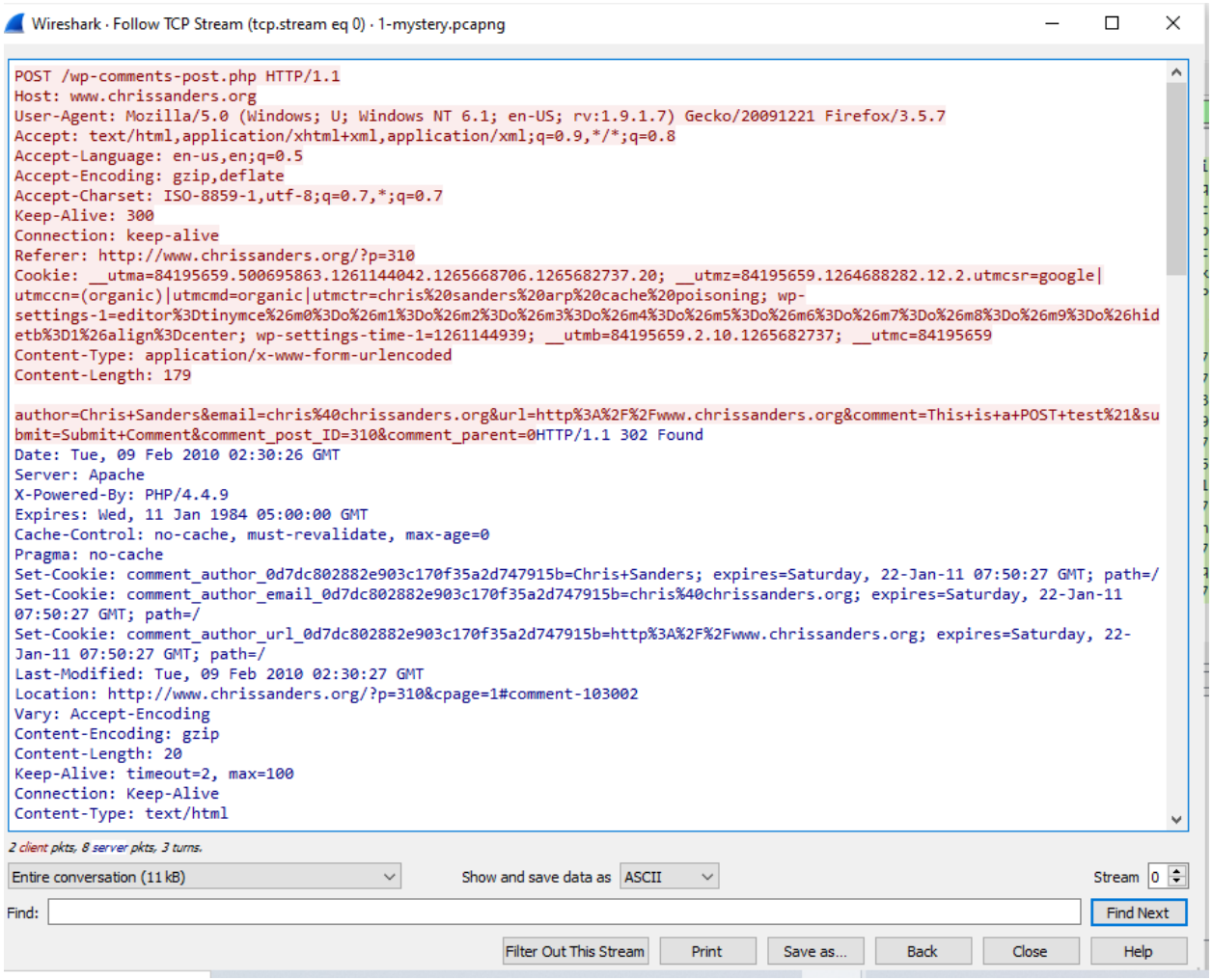
You can see in the .pcapng that someone wants to post something to “chrissanders.org”. The TCP protocol in the transport layer is lightening up with Ack, SYN-ACK, and eventually with a FIN. This tells me that a connection is being made. The next thing I see is the HTTP protocol which tells me actual information is being communicated. Since it is HTPP and not HTTPS the data was not encrypted and could be read in plain text. It looks like to me that Chris Sanders is

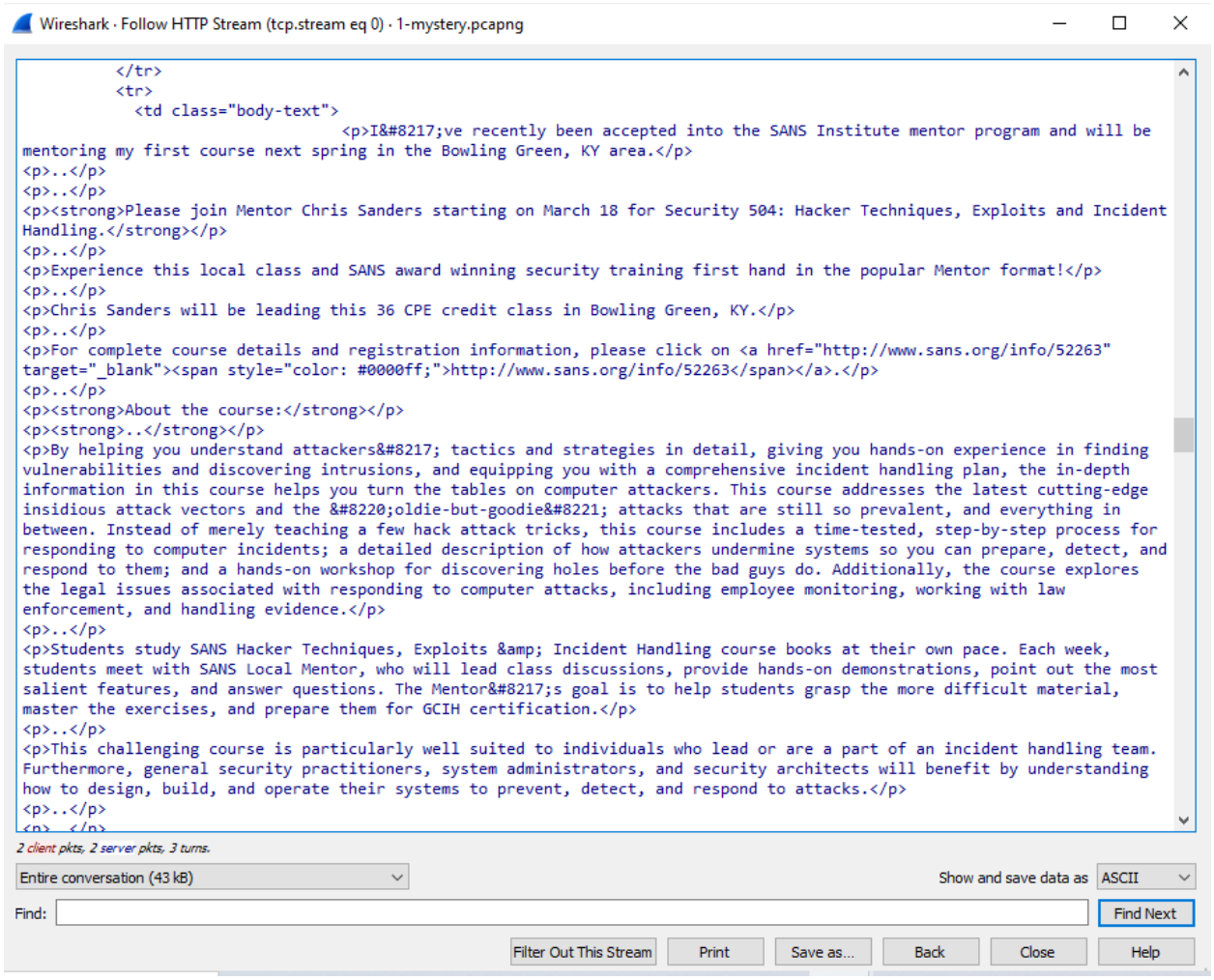
an expert on network security. I noticed that links to what I'm thinking are his literature are posted. I am also seeing some stuff about SANS Hacker Techniques so I am thinking this particular post is about training perhaps. It also looks like a lot of text for images and coding of location, much like you would expect to see on a website that you could post material.

The more technical information to provide is that this the communication was between port 1989 on the server and port 80 on the client. It appears to have four different packets using HTTP protocol and within those, there were individual transfers of 179 bytes and 40327 bytes for a total of 40,506 bytes.

I found this information by looking at the TCP Stream, the HTTP Stream, and the individual packet screen in Wireshark. I included screenshots below.




Wireshark · Follow TCP Stream (tcp.stream eq 0) · 1-mystery.pcapng
POST /wp-comments-post.php HTTP/1.1
Host: www.chrissanders.org
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.7) Gecko/20091221 Firefox/3.5.7
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.chrissanders.org/?p=310
Cookie: __utma=84195659.500695863.1261144042.1265668706.1265682737.20; __utmz=84195659.1264688282.12.2.utmcsr=google|utmccn=(organic)|utmcmd=organic|utmctr=chris%20sanders%20arp%20cache%20poisoning; wp-settings-1=editor%3Dtinymce%26m0%3Do%26m1%3Do%26m2%3Do%26m3%3Do%26m4%3Do%26m5%3Do%26m6%3Do%26m7%3Do%26m8%3Do%26m9%3Do%26hidetb%3D1%26align%3Dcenter; wp-settings-time-1=1261144939; __utmb=84195659.2.10.1265682737; __utmc=84195659
Content-Type: application/x-www-form-urlencoded
Content-Length: 179
author=Chris+Sanders&email=chris%40chrissanders.org&url=http%3A%2F%2Fwww.chrissanders.org&comment=This+is+a+POST+test%21&submit=Submit+Comment&comment_post_ID=310&comment_parent=0HTTP/1.1 302 Found
Date: Tue, 09 Feb 2010 02:30:26 GMT
Server: Apache
X-Powered-By: PHP/4.4.9
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Pragma: no-cache
Set-Cookie: comment_author_0d7dc802882e903c170f35a2d747915b=Chris+Sanders; expires=Saturday, 22-Jan-11 07:50:27 GMT; path=/
Set-Cookie: comment_author_email_0d7dc802882e903c170f35a2d747915b=chris%40chrissanders.org; expires=Saturday, 22-Jan-11 07:50:27 GMT; path=/
Set-Cookie: comment_author_url_0d7dc802882e903c170f35a2d747915b=http%3A%2F%2Fwww.chrissanders.org; expires=Saturday, 22-Jan-11 07:50:27 GMT; path=/
Last-Modified: Tue, 09 Feb 2010 02:30:27 GMT
Location: http://www.chrissanders.org/?p=310&cpage=1#comment-103002
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive
Content-Type: text/html
2 client pkts, 8 server pkts, 3 turns.
Entire conversation (11 kB)
Show and save data as
ASCII
Stream
0
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help


Wireshark · Follow HTTP Stream (tcp.stream eq 0) · 1-mystery.pcapng
</tr>
<tr>
<td class="body-text">
I’ve recently been accepted into the SANS Institute mentor program and will be mentoring my first course next spring in the Bowling Green, KY area.
..
..
Please join Mentor Chris Sanders starting on March 18 for Security 504: Hacker Techniques, Exploits and Incident Handling.
..
Experience this local class and SANS award winning security training first hand in the popular Mentor format!
..
Chris Sanders will be leading this 36 CPE credit class in Bowling Green, KY.
..
For complete course details and registration information, please click on http://www.sans.org/info/52263.
..
About the course:
..
By helping you understand attackers’ tactics and strategies in detail, giving you hands-on experience in finding vulnerabilities and discovering intrusions, and equipping you with a comprehensive incident handling plan, the in-depth information in this course helps you turn the tables on computer attackers. This course addresses the latest cutting-edge insidious attack vectors and the “oldie-but-goodie” attacks that are still so prevalent, and everything in between. Instead of merely teaching a few hack attack tricks, this course includes a time-tested, step-by-step process for responding to computer incidents; a detailed description of how attackers undermine systems so you can prepare, detect, and respond to them; and a hands-on workshop for discovering holes before the bad guys do. Additionally, the course explores the legal issues associated with responding to computer attacks, including employee monitoring, working with law enforcement, and handling evidence.
..
Students study SANS Hacker Techniques, Exploits & Incident Handling course books at their own pace. Each week, students meet with SANS Local Mentor, who will lead class discussions, provide hands-on demonstrations, point out the most salient features, and answer questions. The Mentor’s goal is to help students grasp the more difficult material, master the exercises, and prepare them for GCIH certification.
..
This challenging course is particularly well suited to individuals who lead or are a part of an incident handling team. Furthermore, general security practitioners, system administrators, and security architects will benefit by understanding how to design, build, and operate their systems to prevent, detect, and respond to attacks.
..
2 client pkts, 2 server pkts, 3 turns.
Entire conversation (43 kB)
Show and save data as
ASCII
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help