Conner Mattingly
Cpts 455
HW2

1. The kliks URL for this class is http://kliks.eecs.wsu.edu:7070. In networking terms, what is the
“7070” here? (Section 2.1.2)
- The 7070 indicates the port number.

2. CptS 355 is also using kliks, but the URL for that class is http://kliks.eecs.wsu.edu:8080. Why do
you suppose there is a different URLs for the two classes?
- There has to be so that different data can be accessed by the different users for the different classes.

3. Explain succinctly the difference between a port and a socket.
- A port is location on a computer where data can be sent when received from a network.
- A socket is an IP address along with a Port and is a way for two computers to connect with each other.

4. What is the importance of well-known ports such as port 80 for web, port 25 for smtp, and port 22
for secure shell? Can these services be run on other ports? What reasons might one have for doing that?
What issues have to be dealt with when services are run on non-standard ports?
- Well known ports allow for easy access since the programmer does not have to search for which
port to use because most reserved ports are reserved because or convention. Services can be run on
other ports and one reason to do that would be to make it a little more difficult for people and bots that
scan for vulnerabilities on specific ports.

5. Figure 2.4 says that downloading web documents requires a few kbps. Do you think that a few kbps
of download speed gives a satisfactory user experience? What might have changed since Figure 2.4
was created?
- I think a few kbps may have been satisfactory but now there may be more pictures and videos
on web pages which probably require more data per second.

6. The reliable data transfer guarantee provided by TCP is actually a bit stronger than stated in section
2.1.4: an application reading from a TCP socket is assured of receiving the same stream of bytes sent
by the sender with no loss or duplication and in the same order that the bytes were sent. This additional
guarantee is a source of problems for some time-sensitive applications. Why? (Hint: consider the effect
of lost data (at a layer below the transport layer) on what the transport layer must do in order to meet its
guarantees).
- This can lead to a reduced throughput since packet loss would cause the message to be retransmitted
with TCP

7. Another characteristic of TCP is that its service model is reliable delivery of

a stream of bytes. In
particular it does not preserve segment boundaries from the sender to the receiver. That is, the sender
may send a segment of 1000 bytes followed by a segment of 500 bytes. The receiver may receive a
single segment of 1500 bytes, or 3 segments of 500 bytes, or even 1500 segments of 1 byte each! What
implications does this behavior have for application protocols? As a hint, refer to figure 2.8 and
consider why the blank line is required between the header lines and the request body and could not be
replaced by sending the headers and body separately.
-This behavior leads to the necessity of headers attached to bodys so the sender can
notify the receiver as to how many bytes of data to expect.

8. A similar issue to that of problem 7 governs the issue of “where does the entity body end” in an
HTTP request or response. Why doesn't a blank line suffice to terminate the entity body? What
mechanism is used instead to allow the receiver of a request (or response) to determine the end of the
entity body part of the request (or response). (I do not believe the answer to this can be found in the
book – you will have to do a little research. Cite your sources.)
-Content Length given by the sender in the header of an HTTP request seems to be a common way for
the receiver to know when the body has been fully read. There is also Chunked transfer encoding.
- Source - http://stackoverflow.com/questions/4824451/detect-end-of-http-request-body

9. Section 2.3 concerns FTP, the file transfer protocol. Why is FTP infrequently used today? (Again,
some research is warranted)
- I found that it is not as secure as other protocols due to its weaknesses. It is weak against
certain types of attacks such as FTP bounce attack and Brute Force attack and it is vulnerable to
Packet sniffing while on a network due to its non encrypted transmissions.
_ Source https://en.wikipedia.org/wiki/File_Transfer_Protocol#Security

10. SMTP, POP3, and IMAP are all protocols related to email. Of the three of them, which is essential
to providing email connectivity between essentially all Internet users? Why? Why could we get by
without the other two?
- SMTP (simple mail transfer protocol) is the standard protocol for sending emails across the Internet.
The other two, POP3 and IMAP, are just protocols for accessing emails on a server from a client.
SMTP is the one that connects everything together.

11. The DNS service described in the book is not secure as described on p. 143. Explain in your own
words in a couple of paragraphs how the DNSSEC specifications go about reducing DNS-related
vulnerabilities. Cite any sources you use and DO NOT COPY directly from the sources.

- DNSSEC (Domain Name System Security Extensions) Reduce DNS related vulnerabilities by adding extensions
that secure certain information that would be open to the public otherwise. The main security feature it
adds to the DNS system is origin authentication which essentially allows for clients to be certain that
data received is complete and has not been modified. However DNSSEC does not provide data confidentiality.
All response traffic is authenticated but not encrypted. There are other protocols for encrypting data.

The goal of DNSSEC was to protect against forged and or manipulated Domain Name data. One way this is
accomplished is through the use of digital signatures on answers given by DNSSEC protected network zones.
This allows a DNS resolver to check the signature to see it the information received was tampered with or not.
DNSSEC can also enable other security systems such as SSH fingerprints and IPSec public keys.

Sources:
https://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions
https://en.wikipedia.org/wiki/Message_authentication
https://en.wikipedia.org/wiki/Public-key_cryptography
https://en.wikipedia.org/wiki/DNS_spoofing
https://en.wikipedia.org/wiki/Digital_signature