Samantha Rack
CSE 30264
Homework 10

P1 8.17

(a) When PuTTY is used to SSH to a server that has not been previously connected to, PuTTY will generate a dialog box, warning that the server's key is not yet in PuTTY's cache, displaying the public key of the server, and asking the user if the server can be trusted. If the user responds affirmatively, then that server's key is added to PuTTY's cache and there will be no prompt the next time the user uses SSH to connect to that server.

(b) Clients are authenticated to the server through the use of usernames and passwords. On the opening of the connection, the user using SSH is prompted for her username and then password. If these are valid, then the server concludes that the client is authentic. An alternative method outlined in the PuTTY documentation for a client to verify itself is through the use of public-private keys and signatures.

(c) The list of ciphers supported by PuTTY is available in the initial PuTTY window. When making an SSH connection, PuTTY traverses this list, beginning at the top, and checks if the server has the cipher algorithm until it finds one. The first cipher in the list that is also supported by the server is the cipher that is chosen for the encryption.

(d) When PuTTY determines that the cipher being used is too weak, it displays a warning message to the user, signalling that the cipher is below a threshold of security, and then asks the user if she wants to continue with the session using this cipher. Ciphers can be too weak when the keys are short enough that they could be discovered using brute force methods. Ciphers that are too weak may be supported by PuTTY to provide a less secure option that is faster than stronger ciphers.

(e) Rekeying is implemented by PuTTY approximately every hour to improve the security of a session. The longer the same keys are used for communication, the greater the chance someone will be able to discover the key using methods such as brute force, known plaintext, and chosen plaintext. When the keys are reset, a hacker trying to crack an encryption has to start over.

P2 8.18

(a) When a host initiates a connection using Telnet, it is the client of the communication and will be connecting on port 23 of the server, who will be receiving the connection. When this host connects, it binds to an available port number so the server can send data back to it. So, to enable outbound Telnet connections (ie. those that have been initiated by my host), the firewall must permit the TCP port that my Telnet client has bound to.

(b) Already established TCP connections have the ACK bit set to 1, but the initial connection request packet has the ACK bit set to 0. So, the firewall would allow Telnet activity, but disallow new TCP connections if it blocked all packets with the ACK bit set to 0.

P3 8.19

(a) The PORT command is used on the client with a firewall to allow FTP data transfer to take place over a port. The PORT command is used to specify a port number that will be allowed through the client's firewall. It initiates a second connection with the FTP server so that the data can pass through the firewall. To limit the number of ports that the firewall must allow inbound access, an FTP client could be written so that communication of commands and transfer of data can take place via only one port. This implementation would cause blocking behavior while a file is transferring, and so a user would have to wait until a command has been carried out completely before she could type another command.

(b) FTP PASV command can be used to solve the firewall problem by causing the initiation of the data transfer connection to come from the FTP client instead of the server. Since the client is the establisher, the port will be enabled by the firewall for data transfer.

P4 9.14

(a) This can be implemented through the addition of an HTTP MESSAGE_HEADER option. Because the browser will have to be prepared to handle a new type of option, the browser would have to be upgraded. This option can use some kind of mechanism to indicate which server it found closest to itself. This is filled in at the establishment of the connection.

(b) This can be implemented by adding a hierarchy to the domain name that is used for the website. The client chooses the server's name that corresponds to the one that is closest to it. This would not require an upgraded browser.

P5 9.22

ARP cache timeouts are much shorter than DNS caches because it provides the mapping of IP addresses to hardware addresses as opposed to the mapping of domain names to IP addresses that is provided by DNS. The data in ARP caches is likely to change more quickly, so there should be more periodic updates to the cache.

The consequence of having too long of a DNS cache entry lifetime is the danger of having an expired mapping of a domain name to IP. With a long DNS cache lifetime, nodes have a greater chance of having out of date data.