Homework 3

John Carlyle

February 15, 2014

1. (a) http://pgp.mit.edu/ "John Carlyle"

   (b) Spreads out the web of trust further. Which goes hand in hand with it being able to accuratly indicate your identity. It would be a lot easier to convince one community I am another person. If however I have signatures from multple communities that I knew at different times it is much less likely I have convinced all those people I knew at different times in different geographical locations that I am a different person.

2. Whether or not it is correct, it has a lot of influence. It is currently in the list of 6 links (at least for me) that appear under xkcd on the google search page for "xkcd". I also have not read this comic siunce it came out and I still remember the example password. Even if the security claims are wrong, it certainly has a good point about memorability. I agree with the statement overall, although there are some drawbacks of long passwords. The general principle is that instead of generating long passwords from a small alphabet, we generate short passwords from a much longer alphabet. More concretly, in the XKCD proposed scheme the dictionary is essentially used as an alphabet, and has 2^{11} entries. I think this estimate is actually an underestimate. I did a little bit of research on how word lists are compiled for dictionary attacks and found some ineresting tools. http://www.digininja.org/projects/cewl.php can crawl a website and collect unique words on that site and add them to your wordlist.

   ```
   grep -v \' /etc/dictionaries-common/words | awk 'length > 5' | wc -l
   ```

   yields 62652 words, which makes me think with all the words specific to the target we are interested in added to the wordlist the dictionary size might be closer to 2^{12} or 2^{13} than 2^{11}. Regardless, the entropy of the system using words as the alphabet symbols requires only three words to exceed the entropy of the super confusing password scheme. The only argument I could see for the opposing viewpoint is that perhaps the password scheme is harder to guess which I suppose is why he added an extra three bits of entropy too it in the comic to allow for other password schemes. However if we adhere Kerckhoffs's principle we must assume that the attacker

knows the format of each password. Another reason to argue against large passwords is that they can be misstyped and cost a company money in the form of customer support. Even if it is just a personal password, the longer it is the more prone to annoyance and typing errors it is. For example: my friend and I have a few VPSs that we manage, all of the passwords are 8 words at least, which means I never forget them since they are memorable, but I also can't type them to save my life. Luckily public private key saves the day for SSH connections. This question was a very interesting rabbit-hole that lead me to disable single-user mode on my mac laptops.

3. (a) Alice is being authenticated to Bob in this case. This is being done by the following process:
   i. Alice sends to Bob "Hey I'm Alice!"
   ii. Bob encrypts a Nonce with Alice's public key and sends it back to Alice.
   iii. Alice uses her private key to decrypt the nonce sent over by Bob, and sends it back to prove she is in fact Alice (or at least has Alice's private key).

   The idea here is that no one other than Alice should be able to figure out what n_0. Bob is satisfied that the person on the other end really is Alice if they nonce comes back correct.

4. (a) **Man-in-the-middle attack on the key exchange** Certificates are required for the server to be authenticated to the client. So the man in the middle would be unable to convince the client that he was the server since he would not have a certificate.

   (b) **Password Sniffing** The handshake at the beginning of the session defines a symmetric key to be used by both the client and server for encrypting all communications. Hence any password shared will be encrypted and the evesdropper will be unable to determine the password without the key.

   (c) **IP Spoofing** The MAC will be incorrect because the user was not part of the handshake. This means that SSL will terminate the connection or at least not accept it.

   (d) **IP hijacking** Since the new host does not know the secret key, any messages he recieves will be useless as they cannot be decrypted. Similarly all messages he attempts to send to the other host will not be read since they will have an incorrect MAC and the SSL connection will be terminated immediatly.

   (e) **SYN Flooding** The book does not seem to cover this. It would seem from the internet that SSL does not protect against a SYN flooding attack as this attack is at the TCP layer not the SSL layer. The TCP handshake has to complete before there is a connection

between hosts for the SSL handshake to take place on. And since SYN flooding takes place on the TCP layer it has nothing to do with SSL.
Sources:

i. http://www.arbornetworks.com/asert/2012/04/ddos-attacks-on-ssl-something-old-something-new/
ii. http://en.wikipedia.org/wiki/Transmission_Control_Protocol
iii. http://www.iss.net/security_center/advice/Exploits/TCP/SYN_flood/default.htm

5. From the diagram it looks like when a valid packet is recieved its sequence number is used to determine where in the overall message it fits. The correct slot in the window is then marked showing that htat packet has arrived. If an attacker attempts to replay by sending another packet within that window it will already be marked as arrived. The faked one also cannot possibly come second because it is a replay of anotehr packet, meaning it has to have seen the other packet go by first. If the fake packet arrives too late (outside the window) it is simply rejected.

6. (a) **Advantages** With only two cpu modes the access model is very simple. One would only need a single bit to tell which mode the cpu was mode. Keeping track of who was allowed to do what is also much easier as there is only a very limited number of modes any particular program could enter. **Disadvantages** This is a very inflexible model. The program either has maximum access or very little access. A situation could very easily arrise where you want to allow kernal-like access but restrict certain memory locations form being overwritten, or constrain the program in some other way. With only two modes there is no way to do this.

   (b) More modes means finer granularity on what priviledges you can hand out or revoke.

7. (a) **no access for group and other**

   (b) **read/exec for group none for other**

   (c) **read/exec for group and other**