CS 554 Homework #1

Due: Thursday, Feb 25
Alexander Powell

1) Explain the following concepts:

- **Botnets**

  A botnet is a number of Internet-connected computers communicating with other similar machines in an effort to complete repetitive tasks and objectives. This can be as mundane as keeping control of an Internet Relay Chat channel, or it could be used to send spam email or participate in distributed denial-of-service attacks. The word botnet is a combination of the words robot and network. The term is usually used with a negative or malicious connotation.

- **Phishing**

  Phishing is the attempt to acquire sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a trustworthy entity in an electronic communication.

- **Rootkit**

  A rootkit is a collection of computer software, typically malicious, designed to enable access to a computer or areas of its software that would not otherwise be allowed (for example, to an unauthorized user) while at the same time masking its existence or the existence of other software. The term rootkit is a concatenation of “root” (the traditional name of the privileged account on Unix-like operating systems) and the word “kit” (which refers to the software components that implement the tool). The term “rootkit” has negative connotations through its association with malware.

- **Heartbleed Bug**

  Heartbleed is a security bug disclosed in April 2014 in the OpenSSL cryptography library, which is a widely used implementation of the Transport Layer Security protocol. Heartbleed may be exploited regardless of whether the party using a vulnerable OpenSSL instance for TLS is a server or a client. It results from improper input validation (due to a missing bounds check) in the implementation of the TLS heartbeat extension, thus the bug’s name derives from “heartbeat”. The vulnerability is classified as a buffer over-read, a situation where more data can be read than should be allowed.

- **Stuxnet**

  Stuxnet is a malicious computer worm believed to be a jointly built American-Israeli cyber weapon. Although neither state has confirmed this openly, anonymous US officials speaking to the Washington Post claimed the worm was developed during the Obama administration to sabotage Irans nuclear program with what would seem like a long series of unfortunate accidents.

References for #1:

https://en.wikipedia.org/wiki/Botnet
https://en.wikipedia.org/wiki/Phishing
https://en.wikipedia.org/wiki/Rootkit
https://en.wikipedia.org/wiki/Heartbleed
https://en.wikipedia.org/wiki/Stuxnet

2) Random J. Protocol-Designer has been told to design a scheme to prevent messages from being modified by an intruder. Random J. decides to append to each message a hash of that message. Why doesn't this solve the problem? (We know of a protocol that uses this technique in an attempt to gain security.)

   Solution:
   This fails to solve the problem because anyone can generate and append a hash to a message. Because of this, someone with malicious intent can change the original message if they wanted, and then regenerate the hash value. Futhermore, the person recieving the message won't know that the message has been tampered with.

3) Suppose Alice, Bob, and Carol want to use secret key technology to authenticate each other. If they all used the same secret key K, then Bob could impersonate Carol to Alice (actually any of the three can impersonate the other to the third). Suppose instead that each had their own secret key, so Alice uses KA, Bob uses KB, and Carol uses KC. This means that each one, to prove his or her identity, responds to a challenge with a function of his or her secret key and the challenge. Is this more secure than having them all use the same secret key K? (Hint: what does Alice need to know in order to verify Carol's answer to Alice's challenge?)

   Solution:
   No, it would not be any more secure than having them all use the same secret key, because Alice still needs to know the secret keys of Alice and Bob. In this case, Bob could impersonate Carol, thus making it no more secure than the first example.

4) What's wrong with the protocol in 2.4.4 Authentication? (Hint: assume Alice can open two connections to Bob.)

   Solution:
   The problem with protocol 2.4.4 is that Alice doesn't need to have information about K_{AB}. If Bob challenges using r_B, then Alice can open another connection to Bob and challenge him using r_B. Then, Alice would use Bob's response to respond to his first challenge, and she is able to abort her second connection.

5) How many DES keys, on the average, encrypt a particular plaintext block to a particular ciphertext block?

   Solution:
   Using DES, there are 2^{56} possible keys and 2^{64} ciphertext blocks. Therefore, we require

$$\frac{2^{56}}{2^{64}} = 2^{56-64} = 2^{-8} = \frac{1}{256}$$

   keys on average to encrypt a particular plaintext block to a particular ciphertext block.

6) Are all the 56 bits of the DES key used an equal number of times in the K_i? Specify, for each of the K_i, which bits are not used.

Solution:
The 56 bits of the DES key are not used an equal number of times. This is because 8 bits are left out for every round, and there are 16 rounds, meaning a total of 128 bits are missing. It's clear the $56 \nmid 128$, so the bits are not used equally. The following table shows for each K_i, which bits are not used.

K_i	Bits
1	6, 7, 11, 12, 43, 46, 50, 52
2	3, 4, 35, 38, 42, 44, 61, 62
3	19, 22, 26, 45, 46, 52, 55, 57
4	3, 6, 10, 29, 30, 36, 39, 41
5	13, 14, 23, 25, 49, 52, 53, 59
6	7, 9, 28, 33, 36, 37, 43, 61
7	12, 17, 21, 27, 45, 49, 54, 58
8	1, 5, 11, 29, 33, 38, 42, 63
9	3, 21, 25, 28, 30, 34, 55, 58
10	5, 9, 12, 14, 18, 39, 42, 52
11	2, 20, 23, 26, 36, 58, 61, 63
12	4, 7, 10, 42, 45, 47, 49, 51
13	26, 29, 31, 33, 35, 54, 55, 59
14	10, 13, 15, 17, 19, 38, 39, 43
15	1, 3, 22, 23, 27, 28, 59, 62
16	14, 15, 19, 20, 51, 54, 58, 60

7) Why is a DES weak key (see 3.3.6 Weak and Semi-Weak Keys) its own inverse? Hint: DES encryption and decryption are the same once the per-round keys are generated.

Solution:
For any given week DES key, we know each C_0 and D_0 is either all 1s or all 0s. The same can be said for each C_i since each C_i is a permutation of C_0, and the same goes for each D_i. This tells us that all K_i's are the same, which would mean

$$\{K_1, K_2, K_3, \ldots, K_{16}\} = \{K_{16}, K_{15}, K_{14}, \ldots, K_1\}$$

Therefore, we have shown that in this case the encryption and decryption functions are the same, except for the difference in the order of K_i. So, it is clear that a DES weak key is its own inverse.

8) Why is each DES semi-weak key the inverse of another semi-weak key?

Solution:
Decryption works by essentially running DES backwards. To decrypt a block, you'd first run it through the initial permutation to undo the final permutation (the initial and final permutations are inverses of each other). You'd do the same key generation, though you'd use the keys in the opposite order.

9) Verify the MixColumn result in Figure 3-25 by using the same method (in conjunction with Figure 3-28's table) to compute InvMixColumn of the MixColumn result and checking that you produce the MixColumn input.

Solution:

To verify that the InvMixColumn gives us the same result we start off with the output in the example as our input. That is, the byte sequence we are transforming is $\{42, 4c, b4, 36\}$. The results are shown in the table below.

XOR Operation	Result
b1 $\oplus$ 82 $\oplus$ 85 $\oplus$ 9d	2b
e5 $\oplus$ 64 $\oplus$ 10 $\oplus$ 45	d4
42 $\oplus$ 1a $\oplus$ 77 $\oplus$ f1	de
1f $\oplus$ 63 $\oplus$ 31 $\oplus$ e0	ad

Hence, we have verified that the MixColumn result in Figure 3-25 is correct since we got the original input using the InvMixColumn table.

10) What pseudo-random block stream is generated by 64-bit OFB with a weak DES key?

Solution:

The OFB sequence is

$$E_X(IV), E_X(E_X(IV)), E_X(E_X(E_X(IV))), \ldots$$

Since a weak key is it's own inverse, we can say that for any block $b : E_X(b) = D_X(b)$. Therefore, $E_X(E_X(b))$. This gives us the final OFB sequence:

$$E_X(IV), IV, E_X(IV), IV, \ldots$$

11) Let's assume that someone does triple encryption by using EEE with CBC on the inside. Suppose an attacker modifies bit x of ciphertext block n. How does this affect the decrypted plaintext?

Solution:

Triple encryption of EEE with CBC on the inside is just three CBC encryptions done one after the other. If block n of the ciphertext is modified then the blocks of plaintext from n to $n+3$ are affected down the line. These are the only plaintext blocks that are affected.

12) Consider the following alternative method of encrypting a message. To encrypt a message, use the algorithm for doing a CBC decrypt. To decrypt a message, use the algorithm for doing a CBC encrypt. Would this work? What are the security implications of this, if any, as contrasted with the "normal" CBC?

Solution:

It would work, however if the plaintext blocks are all the same, this will cause all the resulting ciphertexts to be the same except for the first one, since XOR is operated on it.

13) Alice is establishing an account with Bob, a discount on-line broker. She wants her trading to be private. Since both Alice and Bob have heard that the one-time pad is a very secure cryptosystem, they generate a 96-bit-long random pad K for Alice to use in the future for encrypting all her buy/sell orders to Bob. They agree on the following format for each order: First, Alice writes down a single character, either 'B' for 'Buy' or 'S' for 'Sell.' Then she puts a single space, followed by a five-digit decimal number for the number of shares she wants to buy or sell (if she doesn't need to use all five digits, she puts zeros in the

front). Finally, she puts another space followed by the four-letter ticker symbol of the stock she wants to buy or sell (if not all four letters are needed, she puts spaces in the front). Thus, for example, “B 00100 MSFT” means “Buy 100 shares of Microsoft” and “S 25000 AOL” means “Sell 25,000 shares of AOL Time Warner.” Whenever Alice wants to send an order to Bob, she puts her order in the above format, converts the resulting 12-character string to ASCII to get 12 bytes (=96 bits), and encrypts it using the Vernam cipher with the agreed-upon key K. Describe at least one attack against this scheme. You may have certain assumptions (e.g., you may get a copy of a plaintext message). In your submission, you need to clearly state your assumption(s), describe the procedure of your attack, and give an example illustrating your idea. (Hints: there are four types of cryptanalysis techniques: plaintext-only, known plaintext, chosen plaintext, and chosen ciphertext.)

Solution:

If Alice and Bob perform the encryption in a way such that Alice updates the key for each order from Bob, then we can say that Alice will encrypt P_{i+1} as C_{i+1} where $C_{i+1} = P_{i+1}$ XOR k_{i+1}. In this case k_{i+1} is the key for the current message. To decipher the message, Bob will decrypt C_{i+1} just like Alice encrypted it. If someone with malicious intent was able to gain access to the plaintext, then they could also figure out what secret key was used. In this case, the secret key would be found using the following method:

$$k_{i+1} = P_{i+1} \text{ XOR } C_{i+1}$$

Once someone has the secret key, they can decipher any future messages sent from Alice to Bob.

14) See attached java files.