SeungJi Lee
sl136

Homework 5

Problem 1

1. One major feature of grsecurity is that the patch flags data memory as non-executable, and program memory as non-writable. Also, it also provides ASLR, which randomize the important memory spaces. This feature prevents many attacks aimed at memory corruption such as buffer overflow attacks.
2. Grsecurity restricts chroot in many ways to prevent privilege escalation attacks. Some of the restrictions include disallowing kill outside of chroot, no setpgid, getpgid outside of chroot, no pivot_root, no double chroot, etc.
3. Grsecurity also restricts symlink/hardlink to prevent /tmp races.

Problem 2

1. The RSA scheme depends on the Euler's theorem. However, if r is not relatively prime, then, it may not be true that r ^ (ed) = r, since r ^ (phi(n)) may not equal to 1. This is not a major concern in practice since N = pq is chosen to be a multiple of two prime that is large (in general, p, q have 1024 bits each). Then, there are (p-1)(q-1) relatively prime numbers to N. So chances that r is relatively prime to N. But, certainly, it is worth checking for.
2. The scheme can be changed as flow to provide 1 – (1/k) probability that the coin signed is a valid one:
    a. The bank asks the customer to generate k coins and provide cryptographic hashes for each of them.
    b. Then, the bank randomly selects a coin and signs it.
3. Storing n commitment pairs for the identity of the costumer, the bank can learn about the identity of the customer when he uses the same coin twice with probability 1 – (1/2)^n. So, when the coin is used, the customer is required to reveal a(i), or b(i) of the commitment pair depending on randomly selected selector. The merchant stores that strings. Now, when the costumer uses the same coin again, the merchant also stores another set of strings that are parts of the commitment pairs. Now, if we have any full commitment pair, we can reveal the identity of the customer by XORing the pair. So, on the double deposit of the same coin, the bank can find a full pair and identify the customer. However, there is (1/2)^n chance that the customer cannot be identified; when there is no fully pair. So, on the second use, for each pair, there is ½ chances that the same part of the commitment pair is selected. Also, there are n pairs. Thus, there is (1/2)^n chance that the customer is not identified. Then, there are 1-(1/2)^n chance that he is identified.

Problem 3

1. In this mode, if a single bit in the block P(i) is changed, then only i^{th} cipher block = C'(i) will be different from actual C(i). The difference between C'(i) and C(i) may or may not be drastic since E is done with cryptographic hash function.
   If a single bit in the C(i) is changed, then, only the P(i) will be lost.
2. If a single bit in P(i) is changed, then, all cipher text block after (including) i will be different from the actual cipher text blocks.
   If a single bit in C(i) is changed, then, P(i) and P(i+1) will be lost.
3. If a single bit in P(i) is changed, then all cipher text block after (including) i will be different from the actual cipher text blocks.
   If a single bit in C(i) is changed, then, all plain text block after (including) i will be lost.

SeungJi Lee
sl136

4. If a single bit in P(i) is changed, then, all cipher text block after (including) i will be different from the actual cipher text blocks.
   If a single bit in C(i) is changed, then, P(i) and P(i + 1) will be lost.
5. If a single bit in P(i) is changed, then, C(i) will be different from actual cipher text block.
   If a single bit in C(i) is changed, then, P(i) will be lost.

Problem 4

1. With F(i+1), one can simply XOR P(i+1) with F(i+1) to get $D_k(C(i+1))$. Now, all we need is to encrypt this with E_k to get C(i+1). This can be done by setting this $D_k(C(i+1))$. Now, one can get F(0) by sending a cross-realm authentication of pname with just 0's. Then, the first encrypted block of the ticket will be 0 XOR F(0) = F(0). Then, since F(0) is just the encryption key, one can encrypt $D_k(C(i+1))$ to get C(i+1).
2. Since you know the plaintext and C(0), you can get F(1) by C(0) XOR P(0). Using the above attack, you can get C(1). Now, you can use this scheme to get C(i).
3. You can get the first block by requesting a cross-realm authentication with pname set to P(0). The first encrypted block will be the first block of the ticket.

Problem 5

1. We can identify the phone simply as a card in this system. Now, each user has ID, account information. The encrypted ID is once again encrypted by a key that only the user knows, and is stored in the phone. Also, the bank and the user share a shared key. Anytime the user uses the phone to make the purchase, the user sends a message to the bank with four components: { decrypted user ID, merchant ID, amount of payment, a unique seq number different from previous transactions }, encrypted with the shared key. Now, the bank processes the request and put the amount of payment to the merchant's account and sends the validation to the merchant. The validation message is as follows: { merchant ID, amount of payment, the seq number }, encrypted with the key shared by the bank and the merchant. Now, the user sends plaintext of the seq number to the merchant to verify the final payment.
2. Even if the attacker steals the phone, and record all the messages, the attacker cannot use the phone for payments. One, the attacker do not know the key used by the user to encrypt the ID stored in the phone. Two, the transactions hold a unique sequence numbers, and thus message replay cannot succeed.