Homework 3

Due Date: June 1, 2000 **Points**: 200

1. (*10 points*) In a book on UNIX system security, one author states that the reason there has not been a computer virus on UNIX systems other than in the laboratory is because viruses require binary compatibility across systems; that is, the machine languages of the two systems must be compatible. Is he right? Justify your answer.
2. (*20 points*) An *iteration attack* on the RSA cipher is one in which repeated encipherings of the ciphertext produce the plaintext. Consider the ciphertext $C = 3$, $n = 55$, and $e = 17$. Please show that this message can be broken with the iteration attack. Show how to verify the plaintext is correct.
3. (*40 points*)Assume objects are statically bound to security classes.

   a. For the following program, illustrate the compile-time certification checks:

```
program p1(k, m, f, g, h)
var k: file class K;
m: file class M;
f: file class F;
g: file class G;
h: file class H;
a: integer class A;
b: integer class B;
begin
     input a from f;
     input b from g;
     while b ≠ 0 do
     begin
          if a > 0 then output b to h;
          if b > 0 then output a – b + 1 to k;
          output b to m;
          input a from f;
          input b from g;
     end
end.
```

   b. The following partial orders define 2 lattices:

   L_1: $C \le S$

   L_2: $D_0 \subseteq D_1 \subseteq D_3$, $D_0 \subseteq D_2 \subseteq D_3$

   Assume a lattice of security classes constructed from the cross-product of L_1 and L_2 and determine whether the program in part a is secure when

   $A = (S, D_3)$ $B = (C, D_1)$ $F = (S, D_2)$

   $G = (C, D_1)$ $H = K = (S, D_3)$ $M = (C, D_3)$

4. (*30 points*) Suppose someone wrote a file system scanner that computed cryptographic checksums of files, and compared them to a master list, reporting differences. What considerations would the author need to take into account to make this security tool as useful as possible? Discuss attacks and countermeasures.
5. (*100 points*) This continues our penetration testing of *pacific-hts*. In the last exercise you hypothesized flaws in the system's networking implementation. Now it is time to test them!

   a. In each of your three vulnerability descriptions was a short item about how to test for the vulnerability (at least, there was *supposed* to be!) Expand each of these into a full description, as follows:

   your name;

   server with the vulnerability;

   how to verify the vulnerability *if you have source code*. What would you look for? You are free to describe some hypothetical code. For example, if a buffer overflow might occur on input, you would say something like "look for the input functions, and see if they (1) respect buffer boundaries or (2) if they are in a loop that does not check bounds." (The idea here is if you acquire source code, you'll have a starting point.) If you can get the source code and check it, so much the better!

how to verify the vulnerability *in the absense of source code* (if an “attack program” is required, you may use pseudocode to describe the attack program). Be very detailed here; what would “correct” behavior be, and what would erroneous behavior be? If you did this in the previous assignment, you may repeat it here, but please be sure that any competent programmer could reproduce what you plan to do.

effects of exploiting the vulnerability; would you gain access? would you simply deny service or affect the response speed?

disruptions caused by exploiting the vulnerability: would you interfere with normal use of the network? Could you accidentally (or intentionally) interrupt or disrupt others’ use of the network, or others’ systems?

b. If possible, check to see if the vulnerability exists. ***Act ethically – if disruptions could occur other than to the users of pacific-hts, don’t launch the attack!!!*** (If your attack could disrupt the network, please wait ... we will have a Windows 2000 system set up in the security lab next week, on a network you can use to launch attacks. If you need a gullible systems administrator, please let me know and I’ll “turn off” my cynicism for the test.)

For part a, please submit each description in a file labeled with a short name of the vulnerability, and place any exploit tools you need or would like to use into your homework directory. Include a *README* file identifying what you submit. For part b, please submit the results of running your tool or checking for the exploit, and say whether *pacific-hts* is vulnerable, and what the consequences would be if this were exploited.