Massachusetts Institute of Technology — Handout 12
6.857: Network and Computer Security — April 13, 2009
Professor Ronald L. Rivest — **Due:** April 29, 2009

Problem Set 5

This problem set is due *via email,* to `6857-hw@mit.edu` on *Wednesday, April 29* by the beginning of class.

You are to work on this problem set in groups of three or four people. For this problem set, you are free to choose your own groups. If you do not have a group, please email `6.857-tas@mit.edu`, and we will help you find a group. Be sure that all group members can explain the solutions. See Handout 1 (*Course Information*) for our policy on collaboration.

Homework must be submitted electronically! Each problem answer must appear on a separate page. Mark the top of each page with your group member names, the course number (6.857), the problem set number and question, and the date. We have provided templates for LaTeX and Microsoft Word on the course website (see the *Resources* page).

Grading and Late Policy: Each problem is worth 3 points. Late homework will not be accepted without prior approval.

With the authors' permission, we will distribute our favorite solution to each problem as the "official" solution—this is your chance to become famous! If you do not wish for your homework to be used as an official solution, or if you wish that it only be used anonymously, please note this on your homework.

Problem 5-1. Buffer Overflows

The TAs are tired of emailing grades to each group, so they have set up a grade server at `6857-sp09.xvm.mit.edu` where students can check their grades. Each group can log in to the server and check the `grades/my-grades.txt` file to get their current grades. Please send an email to 6.857-tas to get your group's login information.

Naturally, only the TAs have write permissions on the grade files. But when you log in, you also notice a setuid program in your home directory called `update_grades`, along with its C source code. When you inspect the code, you notice a potential buffer overflow that would let you edit your course grades.

You try to tell the TAs about the exploit, but the TAs ignore your complaint and suggest that you stop being so negative. You decide to show the TAs the error of their ways by changing your grades.

(a) These days, GCC has built-in protection against buffer overflows. One of its security measures is to reorder the variables on the stack to place buffers after other local variables. Explain why this measure would eliminate the vulnerability in `update_grades`.

(Since the TAs only write secure code, they turned off the built-in protection by compiling with the `-fno-stack-protector` flag.)

(b) Execute a buffer overflow attack to change the contents of your grade file. Include the string you used in your solution.

(c) Explain how to craft an input string that would execute a shell, therefore allowing you to run arbitrary commands as a TA.

(d) (optional) Execute the attack from part (c). Prove that you managed to get a shell by creating another file in the `grades` directory. Include any code used in the attack.

Problem 5-2. Digitally Signed Email

When you send an email, it is transmitted through many servers, each of which can potentially modify your message. Luckily we can use public key cryptography to sign our messages, and thereby allow others to verify their integrity. We can also use PKI (like OpenPGP) to safely distribute our public keys.

Figure out how to send a digitally signed message using your current mail client to your other project team members. Verify the digitally signed messages received from your project team members. If your current mail client doesn't support signatures, you can download and use Thunderbird with the Enigmail extension.

(a) Write up the steps you needed to do the above. (Include a description of your mail client, etc.) What certificates did you have to work with?

(b) Have each member of your team send a digitally signed message to 6.857-tas. (Note: the staff needs to be able to verify your signature for you to get credit for this problem! You may need to get a certificate to them somehow. We suggest posting your certificates to a PGP keyserver such as pgp.mit.edu.)

Problem 5-3. Evolving Viruses

In his nefarious quest for more harm, Dr. Evil has written a virus that introduces random changes into its code when it infects a new computer. Mutant variants so produced may or may not have the same functionality and/or may or may not be able to reproduce (spread).

Dr. Evil knows of your computer security expertise from taking 6.857, and makes an offer you cannot refuse to enlist your help in identifying variant viruses that have new (and thus possibly more destructive) functionality.

(a) How would you more formally define for Dr. Evil that two virus variants have equivalent functionality? (Dr. Evil instructs you to ignore differences relating merely to the time taken for the virus to do something, or to the memory required for the virus to do something. But dealing with the file system and network are relevant.)

(b) Can you argue to Dr. Evil that in general, the problem of determining whether two virus programs are equivalent is undecidable? How would you do so?

(c) Does your argument in the previous part depend upon an assumption that the virus has a potentially infinite amount of storage available? If so, how? Is it decidable to decide the equivalence of viruses that are restricted to utilizing at most one terabyte of storage?

(d) If deciding the equivalence of two arbitrary virus variants is undecidable, is it perhaps easier to decide if the offpring of a virus has evolved new functionality by comparison to Dr. Evil's particular original virus? Argue your case.