Massachusetts Institute of Technology
6.857: Network and Computer Security
Professor Ronald L. Rivest

Handout 18
November 6, 2003

Problem Set 6

This problem set is due *on paper,* in room 6-120 on *Thursday, November 13* at the beginning of class.

You are to work on this problem set in groups of three or four people. Problems turned in by individuals, pairs, pentuples, etc. will not be accepted. Be sure that all group members can explain the solutions. See Handout 1 (*Course Information*) for our policy on collaboration. If you do not have a group, seek partners by emailing `6.857-students-public@mit.edu`.

Homework must be typed and clear. Each problem answer must appear on separate sheets of paper. Mark the top of each sheet with your names (**alphabetically by last name**), 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.

Grading and Late Policy: Each problem is worth 10 points, except where noted. Late homework will not be accepted without prior approval. Homework should not be submitted by email except with prior approval. (*Somebody* from your group should be in class on the day that the homework is due.)

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 6-1. Crash (Into) Me

WARNING: This problem is quite involved. Start early!

Read Aleph One's article "Smashing The Stack For Fun And Profit" (1996) and Cowan et al's "StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks" (1998). Both are available on the class website.

A server has been set up for your exclusive use on port 666 of the computer `v.lcs.mit.edu`. The server is written in C source code that looks more-or-less like this:

```
#include <stdio.h>
#include <string.h>
int main(int argc,char **argv)
{
    char buf[80];
    char *cc;

    puts("\r\nWelcome to server1. What's your name?\r");
    gets(buf);
    cc = strchr(buf,'\r');
    if(cc) *cc = '\000';

    printf("\r\nHello %s, my name is server1.c\r\n",buf);
}
```

Here is an example of the server at work:[1]

```
[simsong@dhq16 bin] 328 % telnet v.lcs.mit.edu 666
Trying 18.26.1.237...
Connected to v.lcs.mit.edu.
```

[1] You'll get better results using `telnet` on a Unix system than on a Windows system.

```
Escape character is ’^]’.
Welcome to server1. What’s your name?
warning: this program uses gets(), which is unsafe.
Simson
Hello Simson, my name is server1.c
Connection closed by foreign host.
[simsong@dhq16 bin] 329 %
```

Notice that this program displays the message `warning: this program uses gets(), which is unsafe.` when it runs.

(a) Why is `gets()` unsafe?

(b) Rewrite this program so that it is safe.

(c) Exploiting the unsafe bug in the program, provide us with a list of the files on the computer `v.lcs.mit.edu` that are in the directory where this program is running. Create a file in that directory that has your email address as its file name. If you dare, put an interesting message inside that file.

NOTE: We’re sure that this works, but we haven’t gotten it to work yet. If we can’t get it to work, we’ll let you know.

Problem 6-2. Covert Channels

You have been hired as the senior security engineer at Church Commission Revisited, a company that makes the popular covert spyware program $2 + 2 = 5$. A previous version of the company’s product was found to be embedded in a number of popular commercial web browsers. The engineers that developed that product have since gone missing. Hence, your new job.

Your mission at CCR is to develop a new “call home” feature for $2+2 = 5$. That is, the product is constantly collecting information about the user and needs to have a way to leak this information back to CCR’s network of covert webservers (which also house legitimate content such as major news sites, e-commerce site, and pornography services) located around the world. You need to devise a new way for sending $2 + 2 = 5$’s information payloads back to the CCR servers without anyone noticing.

The previous version of $2 + 2 = 5$ employed a variety of stealth techniques — which were unfortunately revealed in a not-very-flattering slashdot article. Those techniques were:

- TCP/IP source port modulation. The bottom 8 bits of the TCP/IP source port number was chosen to transmit 8 bits of information per connection back to the server. This turned out to be rather easy to spot by a simple packet dump analysis.
- Using IP packet options. Each IP packet header has room for “options” to be listed. This was also easy to spot with a packet dump analysis.
- Time modulation. $2 + 2 = 5$ modified the TCP/IP stack so that packets were sent on even-numbered seconds to transmit a “1,” and odd-numbered seconds to transmit a “0.” This technique was very hard to discover and wasn’t generally known until the publication of the slashdot article.

In a well-researched and well-written essay, design **three** new techniques that $2+2 = 5$ could use to transmit information about the user back to the web server. For each of these techniques, estimate the effective data rate and discuss how the technique might be inadvertently revealed. Be sure that your technique works through Network Address Translation appliances such as the current generation of home firewalls.

You may find the “Introduction to the TCP/IP Protocol” article, posted on the course website, to be useful.

For one point of extra credit, write a paragraph about why CCR’s name is significant.

Problem 6-3. Shut Down the Internet

For this, the very last problem of 6.857, we'd like you to shut down the Internet (hey, it had a good run).[2]

(a) Define what it means, in your opinion, to "shut down the Internet." Be sure to state (i) what you mean by "the Internet," (ii) what you mean by "shut down," and (iii) how long the Internet needs to "stay down" to be "shut down."

(b) Estimate how fast the fastest worm could shut down the Internet, using your definition above. Explain your reasoning.

(c) Design a worm that could accomplish the task outlined above. Your worm should be controllable — that is, you should be able to have it shut down the "Internet" of a major university (e.g. Harvard), a medium-sized country (e.g. France), the US Internet, or the "entire" Internet.

You may find some useful ideas at `http://hotwired.wired.com/synapse/feature/97/33/garfinkel0a_text.html`.
Do not write or release this worm! Just design it on paper.

[2] Just kidding. Please don't do this.