**MCS 494 midterm 2**
2004.11.17
D. J. Bernstein

Your answers must be based solely on your own knowledge and the information on this sheet. You are not permitted to use books, notes, or computers. Do not ask your proctor for interpretations or clarifications.

Do not hand in this sheet. Anything that you want to have graded must appear in the answer booklet. Make sure that your name is on the front of the booklet.

**Problem 1.** A file `/etc/crontab.default` has the following contents:

```
0 * * * * root /usr/libexec/atrun
```

A file `/etc/crontab` has the following contents:

```
0 * * * * root /usr/libexec/atrun-local
```

Here is the C code for a program `/bin/crontab-reset`:

```
#include <fcntl.h>
#include <stdio.h>
int main(int argc,char **argv)
{
  char *result = "succeeded";
  int out;
  int in;
  char ch;
  int r;
  out = open("/etc/crontab",O_WRONLY | O_TRUNC | O_CREAT,0644);
  if (out == -1) goto failed;
  in = open("/etc/crontab.default",O_RDONLY);
  if (in == -1) goto failed;
  for (;;) {
    r = read(in,&ch,1);
    if (r == -1) goto failed;
    if (r == 0) goto done;
    if (write(out,&ch,1) < 1) goto failed;
  }
  failed:
  result = "failed";
  done:
  fprintf(stderr,"%s %s\n",argv[0],result);
  return 0;
}
```

The system administrator runs `/bin/crontab-reset`. (1) What number does the first `open()` return? You may assume that nothing unusual happens: in particular, the file will be opened successfully. (2) What number does the second `open()` return? (3) What are the final contents of `/etc/crontab.default`? (4) What are the final contents of `/etc/crontab`? (5) What does the program display on the screen?

**Problem 2.** Same as Problem 1, with one complication: the program receives a TERM signal immediately after the second `open()`. (1) What number does the first `open()` return? (2) What number does the second `open()` return? (3) What are the final contents of `/etc/crontab.default`? (4) What are the final contents of `/etc/crontab`? (5) What does the program display on the screen?

**Problem 3.** Same as Problem 1, with one complication: the program is started with a resource limit preventing it from having more than 4 files open. (1) What number does the first `open()` return? (2) What number does the second `open()` return? (3) What are the final contents of `/etc/crontab.default`? (4) What are the final contents of `/etc/crontab`? (5) What does the program display on the screen?

**Problem 4.** Same as Problem 1, with one complication: file descriptor 2 is closed when the program begins. (1) What number does the first `open()` return? (2) What number does the second `open()` return? (3) What are the final contents of `/etc/crontab.default`? (4) What are the final contents of `/etc/crontab`? (5) What does the program display on the screen?

**Problem 5.** In the situation of Problem 1, assume that `/bin/crontab-reset` is setuid root. A local user `joe` creates a file `evil.c` and runs

```
gcc -o evil evil.c
./evil
```

with the following unauthorized result: `/etc/crontab` contains the line

```
0 * * * * root /home/joe/atrun
```

among other lines. What were the contents of `evil.c`?