Assignment 7. Buffer overruns

Useful pointers

- Elias Levy a.k.a. Aleph One, Smashing the stack for Fun and Profit, *Phrack* **7**, 49 (1996-11-08), file 14.
- David Wheeler, Secure Programming for Linux and Unix HOWTO—Creating Secure Software (2007-08-06).
- Ken Thompson, Reflections on Trusting Trust, *Communications of the ACM* **27**, 8 (1984-08), 761-763.
- CERT Coordination Center

Laboratory: Exploiting a buffer overrun

As usual, keep a log in the file `lab7.txt` of what you do in the lab so that you can reproduce the results later. This should not merely be a transcript of what you typed: it should be more like a true lab notebook, in which you briefly note down what you did and what happened.

For this laboratory, you will find and exploit a simple buffer overrun in a web server.

Consider the following patch to thttpd. This patch applies to thttpd 2.25b.

```
--- thttpd.c~   2005-06-29 10:50:59.000000000 -0700
+++ thttpd.c    2005-11-19 22:27:32.000000000 -0800
@@ -1588,6 +1588,7 @@ handle_read( connecttab* c, struct timev
     int sz;
     ClientData client_data;
     httpd_conn* hc = c->hc;
+    char readbuf[1024];

     /* Is there room in our buffer to read more bytes? */
     if ( hc->read_idx >= hc->read_size )
@@ -1604,7 +1605,7 @@ handle_read( connecttab* c, struct timev

     /* Read some more bytes. */
     sz = read(
-       hc->conn_fd, &(hc->read_buf[hc->read_idx]),
+       hc->conn_fd, readbuf,
        hc->read_size - hc->read_idx );
     if ( sz == 0 )
        {
@@ -1626,6 +1627,7 @@ handle_read( connecttab* c, struct timev
        finish_connection( c, tvP );
        return;
        }
+    memcpy (&(hc->read_buf[hc->read_idx]), readbuf, sz);
     hc->read_idx += sz;
     c->active_at = tvP->tv_sec;
```

1. Build thttpd 2.25b with this patch applied, using GCC's `-fno-stack-protector` option, and run the modified thttpd on port 8080 on your host. You may find the thttpd man page useful.

2. Verify that your web server works in the normal case.
3. Make your web server crash by sending it a suitably-formatted request.
4. Run your web server under GDB, and get a traceback immediately after the crash.
5. Briefly describe how you'd go about building a remote exploit for the bug in the modified thttpd. Your exploit should allow you to run arbitrary code on the web server, with the same privileges as the web server itself.
6. Use GCC's `-S` option to generate the assembly language code for thttpd.c, both with and without the `-fno-stack-protector` option. Call the resulting files `thttpd-f.s` and `thttpd.s`. Use `diff` to compare the two assembly-language files. Which code looks less efficient, and why? Write a simple shell command that invokes `diff` and determines which functions are called (in the sense of the machine-language `call` instruction) by one version and not the other, and use the command to see what functions these are.

Homework: CERT review

Suppose you have built and deployed a networked application from standard software components and are now worried that the application might be vulnerable to outside attackers via the Internet.

Assume that each of the following CERT Vulnerability Notes describes a component of your system. Rank the seriousness of each vulnerability, so that the most urgent vulnerability is listed first. (By "urgent" we mean "urgent that you stay up all night if necessary and fix this right away in your deployed system".) Justify your rankings by evaluating the plausibility of attack scenarios.

- VU#943165 (2010-05-10) Apple Safari window object invalid pointer vulnerability
- VU#886582 (2010-04-19) Java Deployment Toolkit insufficient argument validation
- VU#507652 (2010-04-02) Oracle Sun Java fails to properly validate Java applet signatures
- VU#964549 (2010-03-23) Mozilla WOFF decoder integer overflow
- VU#945216 (2001-10-24) SSH CRC32 attack detection code contains remote integer overflow

Submit

Submit the following files.

- The file `lab7.txt` as described in the lab.
- A file `hw7.txt` containing your answer to the homework.

All files should be ASCII text files, with no carriage returns, and with no more than 200 columns per line. The shell command

```
expand lab7.txt hw7.txt |
  awk '/\r/ || 200 < length'
```

should output nothing.

$Id: assign7.html,v 1.15 2010/05/10 19:56:56 eggert Exp $