

From *Python for Everyone*, by Horstmann and Nicaise, page 283.



Computing & Society 6.1 Computer Viruses

In November 1988, Robert Morris, a student at Cornell University, launched a so-called virus program that infected about 6,000 computers connected to the Internet across the United States. Tens of thousands of computer users were unable to read their e-mail or otherwise use their computers. All major universities and many high-tech companies were affected. (The Internet was much smaller then than it is now.)

The particular kind of virus used in this attack is called a *worm*. The worm program crawled from one computer on the Internet to the next. The worm would attempt to connect to *finger*, a program in the UNIX operating system for finding information on a user who has an account on a particular computer on the network. Like many programs in UNIX, *finger* was written in the C language. In order to store the user name, the *finger* program allocated an array of 512 characters (an array is a sequence structure similar to a list), under the assumption that nobody would ever provide such a long input. Unfortunately, C does not check that an array index is less than the length of the array. If you write into an array using an index that is too large, you simply overwrite memory locations that belong to some other objects. In some versions of the *finger* program, the programmer had been lazy and had not checked whether the

array holding the input characters was large enough to hold the input. So the worm program purposefully filled the 512-character array with 536 bytes. The excess 24 bytes would overwrite a return address, which the attacker knew was stored just after the array. When that method was finished, it didn't return to its caller but to code supplied by the worm (see the figure, A "Buffer Overrun" Attack). That code ran under the same super-user privileges as *finger*, allowing the worm to gain entry into the remote system. Had the programmer who wrote *finger* been more conscientious, this particular attack would not be possible.

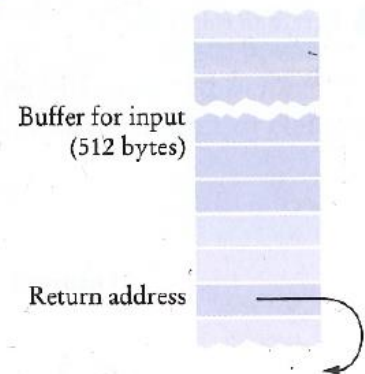
In Python, as in C, all programmers must be very careful not to overrun the boundaries of a sequence. However, in Python, this error causes a run-time exception and never corrupts memory outside the list.

One may well speculate what would possess the virus author to spend many weeks to plan the antisocial act of breaking into thousands of computers and disabling them. It appears that the break-in was fully intended by the author, but the disabling of the computers was a bug, caused by continuous reinfection. Morris was sentenced to 3 years probation, 400 hours of community service, and a \$10,000 fine.

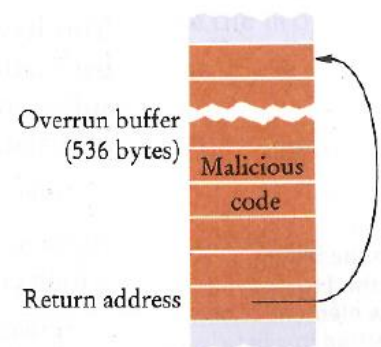
In recent years, computer attacks have intensified and the motives have become more sinister. Instead

of disabling computers, viruses often steal financial data or use the attacked computers for sending spam e-mail. Sadly, many of these attacks continue to be possible because of poorly written programs that are susceptible to buffer overrun errors.

1 Before the attack



2 After the attack



A "Buffer Overrun" Attack