2.2 Buffer Overflow Attacks

Another way of attacking a system is called a buffer overflow (or buffer overrun) attack. Some experts would argue that the buffer overflow occurs just as often DoS attacks, but this is not the case as it was several years ago. A buffer overflow attack is designed to put more data in a buffer than the buffer was designed to hold. This means that although the threat might be less than it once was, it is still a very real threat.

Any program that communicates with the Internet or a private network must receive some data. This data is stored, at least temporarily, in a space in memory called a buffer. If the programmer who wrote the application was more careful, the buffer would truncate or reject any information that exceeds the buffer limit.

Given the number of applications that might be running on a target system, and the number of buffers in each application, the chance of having at least one incorrectly written buffer is significantly high enough to cause any cautious system administrator concern. A person moderately skilled in programming can write a program that purposefully writes more data into the buffer than it can hold. For example, if the buffer can hold 1024 bytes of data and you attempt to fill it with 2048 bytes, the extra 1024 bytes will simply loaded into memory.

If the extra data is actually a malicious program, then it will be loaded into memory and will run on the target system. Or perhaps the perpetrator simply wants to flood the target machine’s memory, thus overwriting other items that are currently in memory and causing them to crash. Either way, the buffer overflow is a very serious attack.

Fortunately, buffer overflow attacks are a bit harder to execute than the DoS or a simple MS Outlook script virus. To create a buffer overflow attack, a hacker must have a good working knowledge of some programming language (C or C++ is often chosen). The hacker must also understand the target operating system/application well enough to know whether it has a buffer overflow weakness and how it might be able to exploit the weakness.