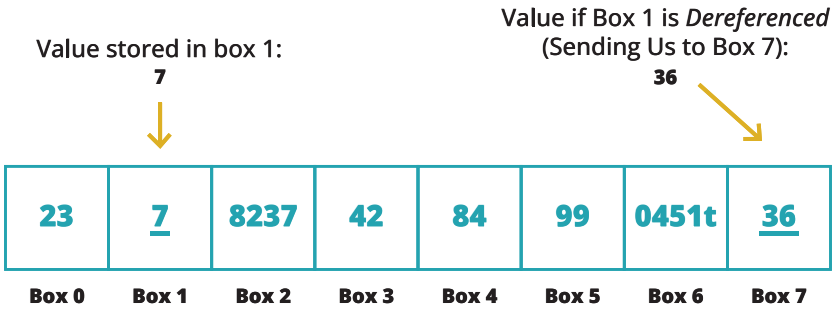
Pointers

A variable that stores the address of another variable is known as a pointer. The majority of programming languages do not require you to be concerned about pointers, but some do, such as C and C++. Pointers are still used "behind the hood" of the language even if you don't have to worry about them when you're programming.

It is possible to read the value at the address that a pointer points to, a process known as dereference. Consider, for illustration, a row of boxes with the numbers 0 through 15 on them. The number 36 is in box fourteen, which is indicated by a pointer in box one. When you read box one's contents, you get 14, but when you defer to box one, you get 36.



A pointer that points to nothing is known as a null pointer. This is different from a pointer referring to a variable with a value of 0, which is like receiving a piece of paper with the number 0 written on it. A null pointer is equivalent to receiving nothing. Dereference attempts normally result in crashes, but they occasionally allow for arbitrary code execution.

Messing With Memory

Memory Leaks

Because the memory in computers is limited, efficiency is key. Programs will deallocate memory that they no longer need if everything is running smoothly, but mistakes might happen. When memory that has been allocated is not released after it has finished being used, a memory leak occurs. Because of this, a software may gradually start sucking up more and more memory until there is none left, which leads to a crash.

The phenomenon of resource exhaustion, in which a computer runs out of finite resources like memory, disk space, network bandwidth, etc., is shown through memory leaks. Attackers may utilize resource exhaustion in denial-of-service attacks.

DLL Injection

A DLL is a file that other programs can utilize to get instructions. Because they may use DLLs that implement the features they require, programmers no longer must write each of their programs from the ground up.

Regrettably, not all DLLs are reliable. When an attacker can attach a harmful DLL to a trustworthy software, this is known as DLL Injection. The DLL seems to the application to be innocent, but it is acting as an agent for the attackers while disguising itself as a component of the genuine program.

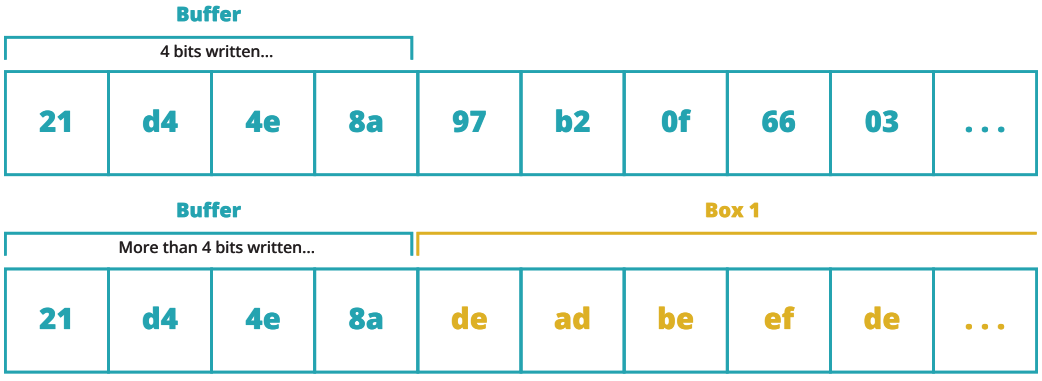
Attackers will try to avoid detection by restructuring their DLLs so that they don't match the antivirus's signatures for known malicious files because antivirus software is aware of malicious DLLs.

Shimming, a technique where a DLL is injected into a program to "translate" out-of-date function calls into ones supported by the current OS, is another way that DLLs are used to preserve compatibility with legacy applications. This is a common technique for introducing harmful DLLs into the software.

Overflow Attacks

It's crucial to be able to manage higher amounts of data because not all of our data neatly fits into a single byte. A buffer, a space of memory allotted by a software to hold data in, is one technique used for this. A software can allocate a 10-byte buffer to hold data if it anticipates receiving 10 bytes of data.

What happens when a buffer is overloaded with data? In the worst-case scenario, a buffer overflow occurs, causing data to keep writing outside of its proper location. Remember that the instructions carried out by the processor are stored in computer memory as well as data. A software may mistakenly overwrite its own code if it writes outside of its buffer, which could result in unexpected behavior.



This can be used for a **Buffer Overflow Attack,** in which the attacker deliberately transmits data that overflows a buffer. This enables the attacker to replace the harmful instructions of the program with their own malicious instructions.

An integer overflow attack is a different kind of overflow attack. Integers, unlike buffers, have a fixed size, but this does not preclude overflowing. Depending on how many bytes they use and how they are processed, integers have upper and lower restrictions on the numbers they can contain. If you attempt to store a number that is larger than the maximum, the number will "wrap around" and start over at the lowest value. This can be used by a cunning attacker to make programs behave in an unexpected way.

Conclusion

By making use of flaws that allow for fraudulently created input to alter the program's code, attackers can directly target binary applications. Malicious DLLs can also be used by attackers to inject malicious code into trustworthy apps.