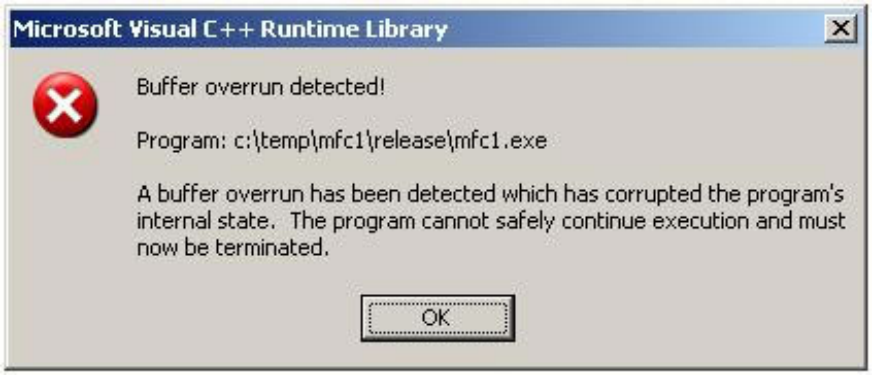
The Internet is experiencing a huge problem with C / C ++, causing developers to 'sweat'

An error affects the iPhone, another error affecting Windows and the third bug affects servers running Linux - the total attack of a certain 'dark army'?

An error affects the iPhone, another error affecting Windows and the third bug affects servers running Linux - the total attack of a certain 'dark army'?At first glance, these errors may seem irrelevant, but in fact all three can derive from the cause of a software being exploited written in a programming language that can cause a kind of error called "memory." unsafety ".Because these types of errors cannot be prevented, programming languages ?? such as C and C ++ unintentionally create conditions for a serious stream of computer security vulnerabilities to spread and grow almost without end in many. year.

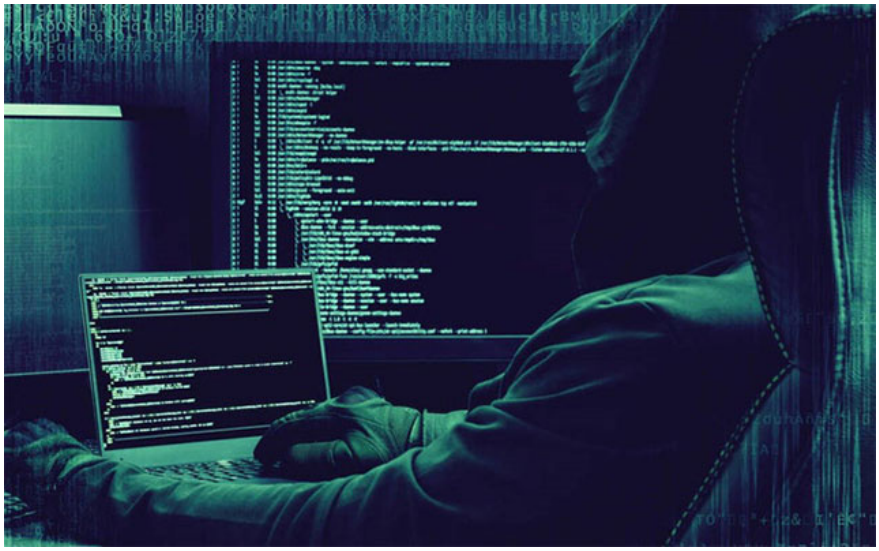
Imagine you have a program with a list of 10 numbers.What if you ask for a list for its 11th element?Most of us say that an error will occur and in this safe programming language for memory (eg Python or Java) this is completely correct.In a memory-insecure programming language, it looks at any memory corner to find the 11th element (if this element exists) and try to access this element. .Sometimes this will lead to a problem, even if that part of the memory has nothing to do with our list.This type of security vulnerability is known as buffer overflow, and this is one of the most common types of memory vulnerabilities.HeartBleed is also a buffer-overflow vulnerability and has affected 17% of secure web servers on the internet.Specifically, HeartBleed allows reading 60 kilobytes of lists, including passwords and other user data.



1. Hacker purged two-factor security just by automated phishing attacks

In addition, there are other types of memory security vulnerabilities related to C / C ++.Examples include type confusion (mixed values ??that exist at memory location), use after free (use part of memory after you have completed the necessary tasks with operating system) and use of uninitialized memory (use part of the memory space before you store anything in it).Together, they form some of the most common vulnerabilities in widely used software such as Firefox, Chrome, Windows, Android and iOS.Security experts have followed and

provided security advice for these platforms for years, and they found that in most releases for these platforms, more than half of the vulnerabilities were Is related to memory insecurity.More worrisome, serious vulnerabilities among them (generally, vulnerabilities can lead to remote code execution, allow an attacker to run any code they want on your computer. This is often the most serious type of vulnerability) that almost always poses a direct threat to the safety of memory.In 2018, Alex Gaynor - a software security engineer at Mozilla, conducted security studies for widely used open source image processing libraries such as ImageMagick and GraphicsMagic, and as a result. He found more than 400 memory-related security vulnerabilities.



If these vulnerabilities become widespread, they can cause widespread damage.Why are these programming languages ??still in widespread use despite containing such serious vulnerabilities?The first reason is that although there is no shortage of programming languages ??that can prevent memory vulnerabilities, but C and C ++ are programming languages ??that are up to decades. century, used very popular and almost became a 'religion', while languages ??can ensure memory safety can be used for low-level programming like web browser and operating system , for example, Rust and Swift . just started to get a little more known.

1. Warning: New extortion code GandCrab is attacking Vietnamese Internet users

A bigger problem is when developers sit down and discuss what programming language they will choose for the new project, they often make decisions based on the language that their team knows best, as well as the effect The productivity and ecosystem of libraries can be utilized.Security is almost never considered as a core element.This means that programming languages ??that emphasize security often do not meet the remaining elements, and that is a disadvantage that makes them unchecked.

Moreover, many important software projects related to internet security are not new, they were started to build more than a decade ago, for example Linux, OpenSSL and Apache web server have all been is more than twenty years old.For large projects like this, rewriting everything in a new language is not a simple thing or can be implemented overnight.This means that projects will need to be written in two languages, instead of one as traditional, and therefore, complexity will also increase significantly.It also means that a large team will need to be retrained, leading to more time and money.

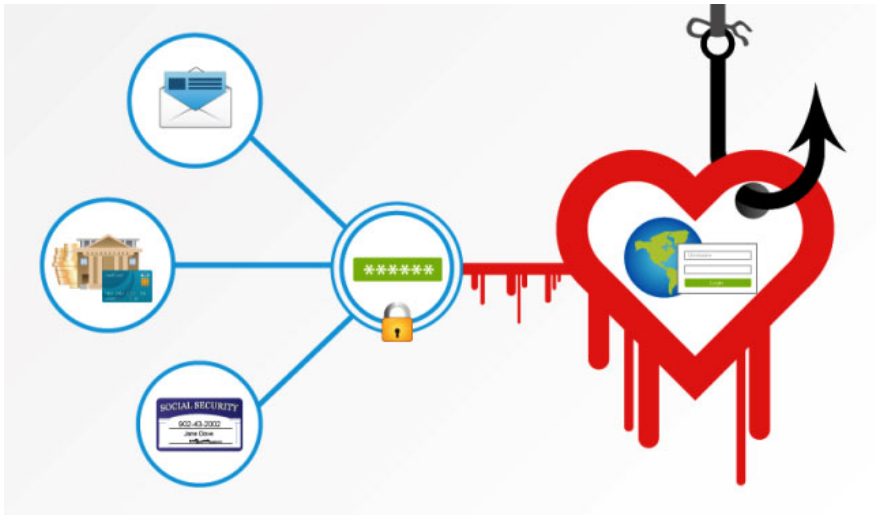
Finally, the biggest problem is that many developers do not believe that security in programming languages ??is a big problem.Many software engineers believe that the problem is not that languages ??like C / C ++ facilitate these vulnerabilities to be exploited, but because other developers have written error codes.According to this

theory, it is not a matter of trying to get the 11th item in the list of 10 items that could lead to a serious vulnerability, but that someone wrote code trying to get the 11th item at the top position. First, and they are an engineer or are not qualified or not enough discipline discipline.In other words, some people think that the problem is not in the programming language itself, but because some people do not know how to use programming languages ??well.

One of the criteria when choosing that programming language is: "How will this choice affect security?"

These vulnerabilities are everywhere and affect companies with the largest security budgets and the most talented developers!And how can we make memory-safe programming languages ??more popular, easier to learn?After hundreds, thousands of vulnerabilities should be able to be prevented by using a better programming language, the evidence clearly shows that "trying harder to avoid errors" is not a possible strategy. exam.

However, there are still some good news.Not everyone denies this issue.Rust is a relatively new programming language, intended to be used for all problems that C and C ++ encounter, while ensuring memory safety and thus avoiding these types of security pitfalls. .Rust is increasingly widely accepted.Currently, it is used by Mozilla, Google, Dropbox and Facebook and this proves that many businesses and organizations are also beginning to look for systematic solutions to unsafe issues. Memory.In addition, Apple Swift is also a memory-safe programming language, while its predecessor is Objective-C, absolutely not.



There are some things we can do to speed up the process of finding comprehensive solutions to memory security security disasters that are becoming more and more serious.First, we should be better aware of the amount of damage that memory insecurity causes.The CVE project, a industry-wide database of known vulnerabilities, can track every vulnerability whether it is a problem related to memory insecurity or not, and whether the language is Does the memory safety program prevent vulnerabilities?This will help us answer questions like, "Which projects will benefit when using memory-safe programming languages?".

1. There is a new zero-day vulnerability in Windows

Secondly, we should invest more in studying how to move existing large software projects to the most secure language memory.Currently, the idea of ??moving something like a Linux kernel to another programming language is an almost impossible task.Therefore, specialized studies on what kind of tools can facilitate this transition or how the programming language can be designed to be simpler will greatly reduce the cost. , effort and time in improving old and large-scale projects.

Finally, we can change people's minds around security in software technology.When I first studied C ++ at university, sometimes the program you wrote might be 'collapsed' for some reason, but someone told you that many of them are holes. hidden security yet?The lack of awareness of the link between errors and problems that developers encounter, as well as issues of concern for security risks early in the career of developers is the symbol for security is still not a secondary concern in software technology and how this subject is taught in the lecture hall.When embarking on a new project, you must accept that one of the most important criteria when choosing a programming language must be "how will this choice affect security?"

Memory insecurity is currently a disaster for our software industry.But it is not the main reason why dozens of Windows or Firefox updates were released to fix the vulnerability that could be avoided earlier.We need to change ourselves in regard to each memory insecurity vulnerability as a single incident, but instead, consider them to be systematic issues.And then, we need to invest in technical research to see how we can build better tools to solve this problem.If successful implementation of that change and investment, we can completely improve security for users and make HeartBleed, WannaCry security vulnerabilities and million dollar issues on

iPhone less popular and less serious.

See more:

1. The corner of getting rich: A company hung a $ 1 million prize for anyone who hacked WhatsApp and iMessage
2. The provisions of the Criminal Code relate to the field of information technology and telecommunications networks
3. Vulnerabilities in Android allow malware to read device information even without permission
4. Warning: A virus that causes a sudden restart of your computer or a blue screen error is exploding in Vietnam