CS 462

Module 10 Assignment

Professor Cartledge

Garrett Moison
11-4-2022

Memory corruption errors are usually a result of problems with your driver or RAM. However, since there are other common causes, it is recommended that individuals attempt other solutions before proceeding with a more comprehensive solution. First, one should scan their computer for viruses, uninstall any recently updated or downloaded third-party software, and utilize their operating system's clean boot or troubleshooting tool to identify the cause of corruption. Suppose the previous steps do not solve the memory corruption. In that case, one can use their operating system's driver verifier tool to detect which driver is creating the vulnerability and search for a solution through the manufacturer. Finally, if all else fails, one can use websites like "memtest86.com" to test their computer's memory sticks and identify any errors. This method should be the last resort since it involves scanning one memory stick at a time and removing the others from the computer.

Buffer overflows are common in systems that use code written in C and C++, which most common operating systems use. However, other languages typically have built-in safety mechanisms that reduce the possibility of a buffer overflow vulnerability. With that in mind, switching programming languages, while problematic, is the most effective method to prevent buffer overflows. On the other hand, suppose one wants to continue using code written in C and C++. In that case, they can utilize security measures in their code or runtime protection tools built into modern operating systems, like ASLR and SEHOP. Finally, when security measures and runtime protection fail, one can investigate third-party security solutions that protect from buffer overflows and other vulnerabilities.

Sarwar Sayeed, Hector Marco-Gisbert, Ismael Ripoll, and Miriam Birch identified several solutions to control flow integrity vulnerabilities in their article "Control-Flow Integrity: Attacks and Protection." Their first solution is to initiate a control flow graph before executing a

program, which monitors runtime behavior and terminates the application if it detects any irregularities. Another solution, titled CCFIR, gathers the legitimate target of indirect instructions and randomly places them in a designated section while the indirect instructions are restricted. Lastly, control flow guard, a popular security mechanism developed by Microsoft, “protects against memory error exploitations by enhancing security where programs execute their codes.”

Insecure data handling occurs due to poor storage encryption, giving adversaries unauthorized access to the data on a victim’s device and resulting in data theft or malware installation. Focusing on Apple’s newer M2 chip, to identify and prevent insecure data storage, one must threat model the mobile application, operating system, and frameworks in use to understand the assets they process and how the APIs handle them. One can do this by testing how they handle features like URL caching, application backgrounding, logging, etc. This method will help users to identify what is causing the vulnerability, so they can research a solution or, if possible, remove the application entirely.

Incorrect usage of APIs, also known as API misuse, can lead to bugs, system crashes, or security vulnerabilities. To identify API misuse, Sarah Nadi and her team developed a benchmark comprised of existing API misuses, which led them to design an API misuse detector called MuDetect. MuDetect mines API usage rules to find misuses in projects. Those API misuses are then displayed and categorized by different aspects using an API Usage Graph (AUG). Once detected, one can correct the individual misuses and patch the vulnerability.

Works Cited

Buffer Overflow Attack. Imperva. (2019, December 29). https://www.imperva.com/learn/application-security/buffer-overflow/

M2: Insecure Data Storage. OWASP Foundation. (n.d.). https://owasp.org/www-project-mobile-top-10/2016-risks/m2-insecure-data-storage

Nadi, S. (n.d.). *API Misuse Detection*. Sarah Nadi. https://sarahnadi.org/smr/api-misuse/

Receiving A Corrupted Memory Error? How to Fix memory_corruption on Your PC. One Computer Guy. (2022, September 9). https://www.onecomputerguy.com/memory-corruption/

Sayeed, S., Marco-Gisbert, H., Ripoll, I., & Birch, M. (2019, October 10). *Control-flow Integrity: Attacks and Protections*. MDPI. https://www.mdpi.com/2076-3417/9/20/4229#cite