CSEP564 : Computer Security : Reading 2

Karuna Sagar Krishna

October 9, 2024

Paper Title

SoK: Eternal War in Memory

Paper Authors

Lazlo Szekeres, Mathias Payer, Tao Wei, Dawn Song

Problem

The goal of the paper as indicated by the title is Systematization of Knowledge (SoK) on memory corruption. The authors provide a holistic view of various memory attacks more abstractly and generally, how these individual attack steps can be combined to formulate various concrete attacks, what defenses exist for such attack steps and why these defenses are adopted or not adopted by wider community. The authors are hoping that by organizing the knowledge it helps future researchers build a better understanding and also helps build defenses that are practical for wider deployment.

Approach

As indicated in the paper, this is a systematization of knowledge paper and they extend and build on top of previous literature. The authors have approached the above goal/problem by first presenting a general model for memory corruption and then listing the main evaluation criteria to rate various defense policies. The general model is laid out as a flow chart with various steps or building blocks which when put together by walking certain flow chart path can lead to various categories of attacks. These steps are further classified to indicate what defense policy category can be applied to mitigate it. The flow chart provides a great overview by establishing a mental model of how memory corruptions are exploited and the authors further expand on various attack categories and discuss few defense policies. The authors have established evaluation criteria to be used when designing future defense policies; these properties indicate why existing defense policies are either adopted or not adopted. Given the flow chart framework and evaluation criteria, the authors walk through various policies where they highlight what security does these policies offer, what is left open, how does it affect performance and if these policies offer various compatibility that enables it to be used in practice. The paper lists various defense policy types as identified by the above framework; probabilistic methods, memory safety, data and control attack defenses. The authors point out various challenges under each of these policy types, evaluate and compare different policy proposal under each type. Certain challenges such as increased use of JIT compilation and user scripting invalidate cheaper and effective defense policies.

Conclusions

The authors aim that researchers will look at the bigger picture to provide stronger policies instead of focusing on specific attack vectors. They also aim to equip these researchers with criteria to evaluate their proposed defense policies so that these policies can be widely deployed instead of being limited to research/academic interest. A suggestion provided by authors is to use open source platforms and compilers to experiment with wider community to test the robustness and performance of the policy.

New Ideas

The paper proposes a general model for memory corruption which abstracts way the details of exact attack vector and instead focuses on the problem patterns. I think this is a great idea to deal with complex problems such as memory corruption since it provides a good framework to rationalize and design defenses that can generally be applied in various scenarios.

Another new and interesting idea is that the authors list evaluation criteria which are properties that defense policies should exhibit to not only effectively mitigate the security risk but also be practical for wider adoption. As engineer, this enables us to evaluate and deploy defense policies accordingly. Further, as indicated in the paper, the authors target researchers to use these properties to design better defense policies.

Improvements

The paper considers memory corruption in low level languages such as C and C++. It is not clear if high level languages with automatic memory management also face similar memory corruption issues, to what extent and what are the differences.

I think, the authors could have the paper more concise by putting up a comprehensive table to describe various defense policies with one or two sentences about how they work, along with references and rate them according to the evaluation criteria. In other words, they could have eliminated sections 5, 6, 7 and 8 and instead summarized via this table. It seems table 2 attempts to accomplish this summarization but it is not comprehensive.

New Directions

The authors could have abstracted out concrete defense policies such as W $\oplus$ R, BBC and SoftBound to explain what are some of the common building blocks/patterns used in building existing policies. This would have helped the reader and potentially future researchers to have a toolbox of various blocks that could be put together to achieve certain defense properties

The paper was published in 2013 and it would be great to revisit this systematization of knowledge paper to see what has changed in the last decade. Recently, Rust has been gaining support from government as well as many tech companies, it would be interesting to see if Rust has similar memory corruption issues and how it help or not help.

Another direction to explore is if we can have AI/ML design general defense policies, or specific defenses for a particular application either by linting the source code, instrumenting the source or binary code.