BACK TO THE BUILDING BLOCKS:

A Path Toward Secure and Measurable Software

FEBRUARY 2024

THE WHITE HOUSE
WASHINGTON

Table of Contents

ABSTRACT

President

PART I: INTRODUCTION

Users of software and hardware products are consistently placed in the untenable position of reacting to cyber emergencies. Responding on a crisis-by-crisis basis often leaves them on their heels, and those securing systems on the front lines should not be expected to bear the full weight of this burden. The intense reactive posture demanded by the current status quo reduces

PART II: SECURING THE BUILDING BLOCKS OF CYBERSPACE

To reduce the burden currently placed on end users to protect themselves from cybersecurity threats, efforts must be made to proactively eliminate entire categories of software vulnerabilities. To better understand the prevalence of these categories, software manufactures should consider publishing timely, complete, and consistent Common Vulnerability and Exposures (CVEs) data, including the Common Weakness Enumeration (CWE). Past analysis of CVE data identified memory safety bugs as one of the most pervasive classes of vulnerabilities that has plagued cyber defenders for decades.

Memory safety vulnerabilities are a class of vulnerability affecting how memory can be accessed, written, allocated, or deallocated in unintended ways.[iii] Experts have identified a few programming languages that both lack traits associated with memory safety and also have high proliferation across critical systems, such as C and C++.[iv] Choosing to use memory safe programming languages at the outset, as recommended by the Cybersecurity and Infrastructure Security

PART III: ADDRESSING THE SOFTWARE MEASURABILITY PROBLEM

To make progress toward securing the digital ecosystem, it is necessary to realign incentives to favor long-term investments. For this realignment to generate ecosystem-wide behavior change, it is critical to develop empirical metrics that measure the *cybersecurity quality* of software. This will help inform both producer and consumer decision-making, as well as public policymaking efforts. Ongoing work to improve how software quality and security are understood, including coordinated vulnerability disclosure, response programs, and timely CVE records, informs essential decision making throughout the ecosystem. Nevertheless, more progress is required to develop empirical metrics to effectively measure code.

Software measurability is one of the hardest open research problems to address; in fact, cybersecurity experts have grappled with this problem for decades. The problem requires not only refining existing metrics or tools, but also the pioneering of a new frontier in software engineering and cybersecurity research. By advancing capabilities to measure and evaluate software security, more vulnerabilities can be anticipated and mitigated before software is released. The metrics developed from these measurements will also inform the decision-making of a broad range of stakeholders, further improving the security of the digital ecosystem and incentivizing long-term investments in secure software development.

Challenges with Software Measurability

In 2016, the software quality group at the United States National Institute of Standards and Technology (NIST) convened a workshop to explore enhancements in software measures and metrics, aiming to dramatically reduce software vulnerabilities.[xxviii] The conclusions from this workshop underscored the multifaceted nature of the software measurability problem and identified three core challenges: software metrology, software behavior, and software analysis.

First, the inherent challenge of software metrology

PART IV: CONCLUSION

The challenge of eliminating entire classes of software vulnerabilities is an urgent and complex problem. Looking forward, new approaches must be taken to mitigate this risk. Doing so will allow the United States to continue its progress toward President

PART VI: ENDNOTES

[i] *The National Cybersecurity Strategy Implementation Plan*, The White House, July 2023 *available at* NCSIP.WH.GOV.

[ii] *The National Cyber Workforce and Education Strategy*, The White House, July 2023, *available at* NCSWES.WH.GOV.

[iii] *The Case for Memory Safety Roadmaps*, Department of Homeland Security, Cybersecurity Infrastructure and Security Agency, December 2023, *available here* [hereinafter Roadmaps].

[iv] *See id*. *See also* Introduction to Memory Unsafety for VPs of Engineering · Alex Gaynor, August 2019, *available here*.

[v] *See CISA Open Source Software Security Roadmap*, Department of Homeland Security, Cybersecurity Infrastructure and Security Agency September 2023, *available here*.

[vi] *See Report to the CISA Director*, Department of Homeland Security, Cybersecurity Infrastructure and Security Agency, December 2023, at page 2, *available here*.

[vii] *See generally* Morris worm - Wikipedia. *See also* National Vulnerability Database, Department of Commerce, National Institute of Standards and Technology, *available here*. *See also BLASTPASS: NSO Group iPhone Zero-Click, Zero-Day Exploit Captured in the Wild* , University of Toronto, Munk School of Global Affairs and Public Policy, September 2023, *available here*.

[viii] *See Microsoft MSRC Blog We Need a Safer Systems Programming Language*, July 2019, *available here*.

[ix] *See Memory Safe Languages in Android 13*, Google Security Blog, December 2022, *available here*.

[x] *See generally About Prossimo*, *available here*. A risk criteria framework to prioritize code rewrites can be defined as software that is

[xxx] *Id*.
[xxxi] NIST Special Publication *supra* note xxvii at 24.
[xxxii] *Id*. at 59.
[xxxiii] *See Federal Cybersecurity Research and Development Strategic Plan*, National Science and Technology Council, The White House, December 2023 *available here*.
[xxxiv] *See*

THE WHITE HOUSE
WASHINGTON