On the Idea of Risk And Input Complexity: Embracing Insecurity Using Weird Machines Theory to Measure Attack Surfaces

Froylan Maldonado
Naval Information Warfare Center Pacific
U.S. Department of Defense
San Diego, U.S.A.
froylan.g.maldonado.civ@us.navy.mil

Dr. Matthew L. Levy
Naval Information Warfare Center Pacific
U.S. Department of Defense
San Diego, U.S.A.
matthew.l.levy.civ@us.navy.mil

***Index Terms*—Attack Surface, Measurement, Weird Machines.**

I. Extended Abstract

We consider the idea of an attack surface and the ability to measure an attack surface as instrumental to understanding the magnitude of vulnerabilities in applications and systems that are critical to cybersecurity risk assessments in organizations. However, attack surfaces are an ill-defined concept in academia and practice [1]–[3]. Thus, we submit to this workshop that the ideas inherent in LangSec, non-exploitability, and insecurity are promising in providing actuarial level precision in providing a topological understanding of a few key concepts: (1) Attack surfaces. (2) Conceptualizing theoretical applications that have attack surfaces of zero. (3) Attack surfaces of non-zero orders of magnitude.

In practice, attack surfaces appear well-defined. In research, there is a paucity of explicit definitions and measurement concepts. For example, practitioner literature defines attack surfaces as (1) the sum of vulnerabilities, pathways, or methods—sometimes called attack vectors—that hackers can use to gain unauthorized access to the network or sensitive data or to carry out a cyberattack, or (2) The set of points on the boundary of a system, a system component, or an environment where an attacker can try to enter, cause an effect on, or extract data from that system, component, or environment [2]–[4]. In the above definitions, (1) is measuring all attack vectors while (2) is measuring points on the boundary of a system. Clearly, this illustrates the difficulty in understanding attack surfaces to measure them accurately. This might also explain why attack surface measurement research has been reduced to entry and exit points [4]. Given the myriad ways an attacker can enter, affect, escalate privileges, and extract data from a system or environment, we contend an attack surface is much more than described above.

Conversely, the theories embodied in LangSec [5]–[7] are promising in offering a rejoinder to the discourse on attack surface measurement. Under these auspices, the attack surface is the property of the code and property of the data format, and if the code is complex, it can involve many states with many inputs and thus becomes responsible for increasing the attack surface. The tuple we allude to (E, Q) consist of an exploit string and a set of weird states that are accessed by the exploit string. Thus, an attack vector can be described as a sequence of inputs, E, that map to a subgraph Q of the program that is running on the CPU. This then implies that all the tuples of the form (E,Q) define an attack surface of a program. In theory, if we can map exploit strings to actual states in a program, it is possible to envisage a topological space that can be manipulated using known mathematical techniques. In sum, discussing attack surfaces from a LangSec perspective – as an actual "topological surface" can potentially lead to interesting findings.

References

[1] Christopher Theisen, Nuthan Munaiah, Mahran Al-Zyoud, Jeffrey C Carver, Andrew Meneely, and Laurie Williams. Attack surface definitions: A systematic literature review. *Information and Software Technology*, 104:94–103, 2018.

[2] National Institute of Standards and Technology. Protecting controlled unclassified information in nonfederal systems and organizations. Technical Report NIST Special Publication 800-171, Revision 2, U.S. Department of Commerce, Washington, D.C., 2020.

[3] Inc. IBM Systems. What is an attack surface? https://www.ibm.com/topics/attack-surface, 2014. Accessed: 2024-01-10.

[4] Pratyusa K Manadhata and Jeannette M Wing. A formal model for a system's attack surface. In *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, pages 1–28. Springer, New York, NY, 2011.

[5] Thomas Dullien. Weird machines, exploitability, and provable unexploitability. *IEEE Transactions on Emerging Topics in Computing*, 8(2):391–403, 2017.

[6] Sergey Bratus, ME Locasto, and ML Patterson. Exploit programming: From buffer overflows to "weird machines" and theory of computation. *SECURITY ;login*, 36:1–9, 2011.

[7] Sergey Bratus and Anna Shubina. Exploitation as code reuse: On the need of formalization. *it-Information Technology*, 59(2):93–100, 2017.