GUARNARY: Mitigating Buffer Overflow Using Hardware Assisted Virtualization Features

Stella Bitchebe[+,o], Yves KONE[+,*], Alain Tchana[+], [+]Labo LIP - Ecole Normale Supérieure de Lyon, [o]Labo I3S - Université Côte d'Azur, [*]Labo IRIT - INP Toulouse

Compas 2022

1. Context and Motivation

- Buffer overflow is the top one vulnerability in 2021 according to the CWE (Common Weakness Enumeration) [1]
- Secure Allocators (Slimguard [2], Guarder [3], etc.) use *guardians* to prevent and detect overflows
- Existing types of guardians:
  - **Canary**: Low memory overhead + Asynchronous detection
  - **Guard Page**: High memory overhead + Synchronous detection

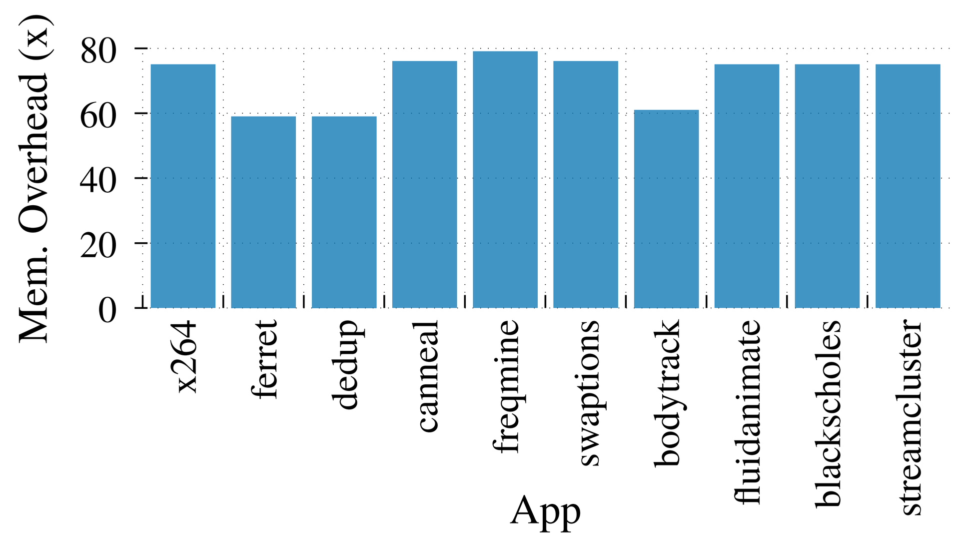


Figure 1: Memory waste of PARSEC when all application's buffers are allocated at the boundary of a guard page using Slimguard as the memory allocator.

2. Dilemma: Synchronuous Dectection vs. Memory Overhead

- Security distance: for a vulnerable buffer `b`, it is the number of bytes that separate it from a guardian

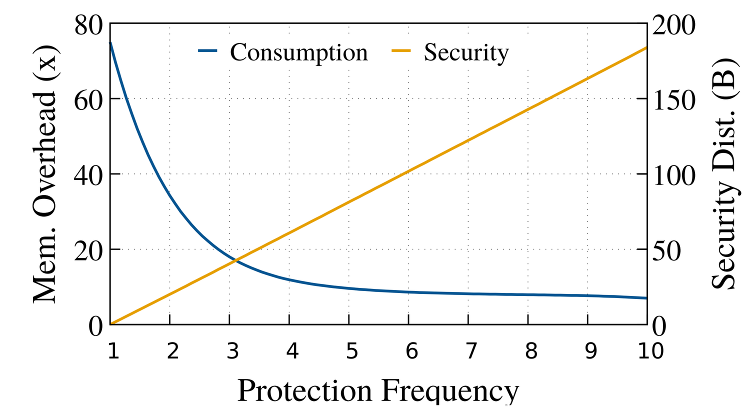


Figure 2: Memory waste and average security distances of PARSEC-`blackscholes` when varying the protection frequency.

- Protection frequency: `F` is called the protection frequency if a guard page is placed for every `F` allocated buffers

Frequency	1	2	3	4	5	6	7	8	9	10
Buffers protected(%)	100	50	33.33	25	20	16.67	14.28	12.5	11.11	10

Table 1: Proportion of PARSEC-`blackscholes`'s buffers placed at the boundary of a guard page for different values of the protection frequency. The allocator is Slimgard.

- User configures `F` and the allocator combines guard pages with canaries to minimize the security distance while optimizing the memory consumption

3. Intel SPP: Sub-Page Write Permission

SPP [4] is a recent Intel hardware virtualization feature that allows the hypervisor to write-protect guest's memory at a sub-page (128B) granularity instead of 4KB.

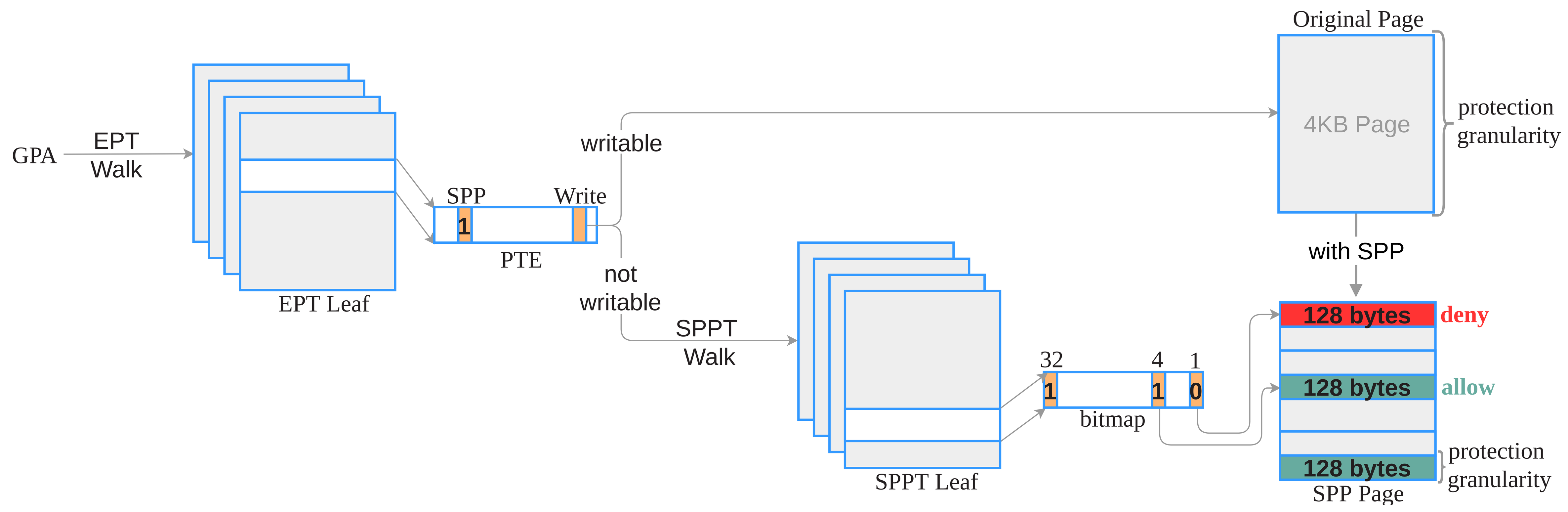
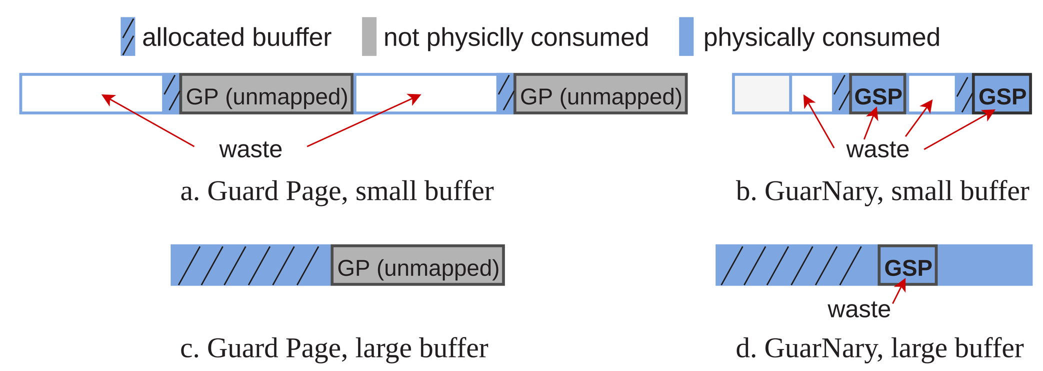


Figure 5: Overview of SPP functioning.

4. Guarnary Challenges

Guarnary is a novel type of guardian that uses Intel sub-pages as barriers. It is midway between canaries and guard pages, which gives it the advantages of both guardians: low memory consumption and synchronous detection. Its use raises the following challenges:

(C_1) One size does not fit all:



Guarnary must satify the following equation:

$$\sum GN + \sum RGN < \sum RGP$$

Figure 3: Challenge C_1 illustration. GSP stands for guarg sub-page, $\sum GN$ memory consumed by guarnary, $\sum RGN$ and $\sum RGP$ internal fragmentation waste resp. of guarnary and guard page.

(C_2) Costly hypercalls: SPP is configurable solely by the hypervisor.

(C_3) Physical page heterogeneity: see Figure 4.

(C_4) Protection pattern heterogeneity: the protection pattern is the bitmap of sub-pages that are write-protected within an SPP page.

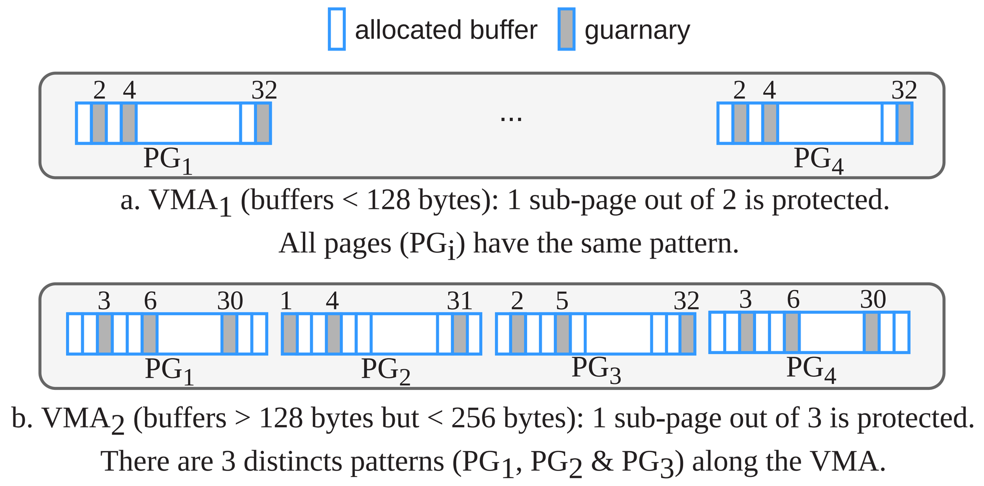


Figure 4: Protection pattern and frequency illustration.

References

[1] Cwe/sans top 25 most dangerous software errors. https://www.sans.org/top25-software-errors, 2022.

[2] Beichen Liu et al. Slimguard: A secure and memory-efficient heap allocator. *Middleware*, 2019.

[3] Sam Silvestro, Hongyu Liu, Tianyi Liu, Zhiqiang Lin, and Tongping Liu. Guarder: A tunable secure allocator. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 117–133, Baltimore, MD, August 2018. USENIX Association.

[4] Intel ept-based sub-page write protection support. https://lwn.net/Articles/736322/, Oct 2017.

Contact

yves.kone@ens-lyon.fr

bitchebe@i3s.unice.fr

alain.tchana@ens-lyon.fr