# Patch Me If You Can—Securing the Linux Kernel

Gunnar Kudrjavets*
*Amazon Web Services*
Seattle, WA 98109, USA
gunnarku@amazon.com

***Abstract*—In February 2024, the Linux kernel became a CNA (CVE numbering authority). The number of CVEs issued for the kernel increased by an order of magnitude. This increase places additional patching demands on kernel vendors and software companies maintaining custom Linux kernels. The industry needs the software analytics research community's help to understand the patch velocity, develop the prediction models, and estimate the effort required to patch the kernel.**

## I. Background and motivation

When the Linux kernel became a CNA, it changed the semantics of a CVE (Common Vulnerabilities and Exposures). Nearly every defect in the kernel is now a CVE. The kernel team made this decision because "almost any bug might be exploitable to compromise the security of the kernel" [1], [2]. Between 2006 and 2018, approximately 84 CVEs per year were classified to impact the kernel [3]. From February till May of 2024, the kernel CNA issued an average of 19 CVEs *per day* [3].

Various kernel vendors, such as Red Hat and any company using custom kernels, need to handle patching at least an order of magnitude more CVEs than in the past. Another factor that complicates this situation is that organizations that analyze the severity of CVEs, such as NVD (National Vulnerability Database), must promptly handle this additional incoming flow of CVEs. The analysis is necessary to decide in which order to patch the CVEs. However, even fixing a subset of CVEs in a prioritized order is an unsupported practice. The Linux kernel CVE team states that "[C]herry-picking individual commits is not recommended or supported by the Linux kernel community at all" [4]. The kernel CNA issues a CVE only if a fix is available in the stable kernel branch [1]. Anyone who uses a kernel that is not based on the stable branch must fight "The Forever (Patching) War" against the CVEs [5].

Researchers have investigated various aspects of the kernel patching process in the past [6], [7], [8], [9]. Given the recency of CVE-related changes, limited research has been conducted and published on this topic. Most of the findings come from grey literature and recent white papers. Since the kernel became a CNA, the industry's requirements for patching velocity have dramatically changed [10].

* Conducting research is not related to Gunnar Kudrjavets' role at Amazon.com, Inc. All opinions and statements communicated in this paper are the author's own.

## II. Availability of data

Fortunately for the software analytics research community, all the necessary data for mining and analysis is publicly available. The Linux release model uses several branches, such as mainline, stable, and LTS (Long-Term Support) [11]. Researchers can analyze and track various metadata associated with the patches and the speed with which engineers (back)port them between different branches. A cut-off date of February 13, 2024 (when the Linux kernel became a CNA) establishes the clear separation between the two "CVE patching eras" [12].

## III. Industry's needs

The kernel's development process and branching model are well-documented [13]. The kernel team maintains several LTS branches (e.g., 4.19.x, 5.10.x, 6.6.x). Each CVE can apply to zero or more LTS branches. We define the *patch ratio* as a percentage of all applicable CVEs patched at a given time for a specific branch. We define the *time-to-patch* as the time from issuing a CVE until the patch for that CVE is committed to a specific LTS branch.

The industry needs data and help to find answers to the following questions:

1) **Trends and velocity**. Are there statistically significant differences in the patch ratio between the LTS branches? Do some LTS branches (newer, older) get patched faster than others? Do patch ratio and time-to-patch increase or decrease over time? Is the patching done mainly by volunteers or commercial organizations? Does time-to-patch depend on CVE's severity?
2) **Predicting the incoming CVE rate**. A prediction model will help determine the resource allocation needed for timely patching. Are some architectures more impacted than others? Do some subsystems have more patches than others? Are more CVEs issued on weekdays or weekends? What is the eventual post-analysis distribution of CVEs of a different severity type [14]?
3) **Maintenance cost**. How many patches are ported to LTS branches? How many patches apply cleanly? How many need backporting? What is the estimated cost of patching a CVE? Can the patching process be automated?

Based on the initial observations from 2024, we do not expect a sudden decrease in the number of CVEs issued by the kernel CNA. The kernel community's official stance is that "the CVE assignment team is overly cautious and assign CVE numbers to any bugfix that they identify" [1].

## References


[1] The kernel development community. (2024, Dec.) CVEs. [Online]. Available: https://docs.kernel.org/process/cve.html

[2] G. Kroah-Hartman. (2024, Oct.) Linux Kernel CVEs: What Has Caused So Many to Suddenly Show Up? [Online]. Available: https://www.youtube.com/watch?v=KumwRn1BA6s

[3] TuxCare. (2024, Jul.) A look at 3 months of Linux kernel CVEs. [Online]. Available: https://tuxcare.com/wp-content/uploads/2024/07/TuxCare-Report-3-Months-of-Kernel-CVEs.pdf

[4] The Linux kernel CVE team. (2024, Sep.) General disclaimer about patching. [Online]. Available: https://lore.kernel.org/linux-cve-announce/2024091833-CVE-2024-46717-2f30@gregkh/T/

[5] J. Haldeman, *The Forever War*. New York, NY, USA: St Martin's Press, Feb. 2009.

[6] X. Li, Z. Zhang, Z. Qian, T. Jaeger, and C. Song, "An investigation of patch porting practices of the Linux kernel ecosystem," in *Proceedings of the 21st International Conference on Mining Software Repositories*, ser. MSR '24. New York, NY, USA: Association for Computing Machinery, Jul. 2024, pp. 63–74. [Online]. Available: https://doi.org/10.1145/3643991.3644902

[7] Y. Xu and M. Zhou, "A multi-level dataset of Linux kernel patchwork," in *Proceedings of the 15th International Conference on Mining Software Repositories*, ser. MSR '18. New York, NY, USA: Association for Computing Machinery, May 2018, pp. 54–57. [Online]. Available: https://doi.org/10.1145/3196398.3196475

[8] R. Liu, H. Shi, Y. Zhang, R. Wang, Y. Shen, Y. Chen, J. Luo, X. Shi, C. Hu, and Y. Jiang, "PatchBert: Continuous stable patch identification for Linux kernel via pre-trained model fine-tuning," in *2024 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. Los Alamitos, CA, USA: IEEE Computer Society, Mar. 2024, pp. 349–358. [Online]. Available: https://doi.org/10.1109/SANER60148.2024.00042

[9] Y. Jiang, B. Adams, and D. M. German, "Will my patch make it? And how fast? Case study on the Linux kernel," in *2013 10th Working Conference on Mining Software Repositories (MSR)*, ser. MSR '13, San Francisco, CA, USA, May 2013, pp. 101–110. [Online]. Available: https://doi.org/10.1109/MSR.2013.6624016

[10] D. Melotti. (2024, Sep.) Linux CVEs Open Discussion. [Online]. Available: https://www.youtube.com/watch?v=RaQB1qX2KaA

[11] Linux Kernel Organization, Inc. (2024, Dec.) Active kernel releases. [Online]. Available: https://www.kernel.org/category/releases.html

[12] G. Kroah-Hartman. (2024, Feb.) Linux is a CNA. [Online]. Available: http://www.kroah.com/log/blog/2024/02/13/linux-is-a-cna/

[13] The kernel development community. (2024, Dec.) How the development process works. [Online]. Available: https://docs.kernel.org/process/2.Process.html

[14] Red Hat, Inc. (2024, Dec.) Understanding Red Hat security ratings. [Online]. Available: https://access.redhat.com/security/updates/classification