Reducing Useless Agent Actions in RL-based Cache Structure Vulnerability Exploration

Kanato Nakanishi, Soramichi Akiyama
Ritsumeikan University, Osaka, Japan

ACM Reference Format:
Kanato Nakanishi, Soramichi Akiyama. 2024. Reducing Useless Agent Actions in RL-based Cache Structure Vulnerability Exploration. In *Proceedings of Asia-Pacific Workshop on Systems (APSys'24).* ACM, New York, NY, USA, 1 page. https://doi.org/XXXXXXX.XXXXXXX

1 Background and Problem

Cache timing attacks exploit cache memory timing information to obtain confidential data [1] and it is a serious threat. AutoCAT [2] is a reinforcement learning-based (RL) approach to automatically find if a given cache structure is vulnerable to cache timing attacks. The use of RL is promising because it does not require human experts and can potentially find previously unknown attack patterns.

Although promising, the problem of AutoCAT is that the learning process is time-consuming. The first reason is that each training trial (one conversion of the model) requires a significant amount of time. In our experiments, one training trial for a particular cache configuration took more than 4 hours. The second reason is that multiple trials of training are necessary to comprehensively reveal vulnerabilities for each cache structure. We found that AutoCAT discovered different attack sequences in different training trials, even for the same learning parameters. In the hardware product development life cycle, Engineering Validation Test and Design Validation Test involve many tasks beyond vulnerability assessment[1]. Spending excessive time on vulnerability assessment directly leads to delays in the entire product development process.

2 Proposal and Early Results

We propose a method to reduce the learning time of AutoCAT by reducing useless agent actions in the exploration of attack sequences. An attack sequence refers to a series of actions such as accessing the cache, flushing a cache line, and letting the victim move. The main idea is that there is no need to try actions that do not alter the cache state because they do not contribute to the attack. For example, in a flush+reload attack, performing a flush action on a cache line immediately followed by another flush on the same cache line is useless. Useless actions are detected by calculating the hash of the cache state. If the hash values before and after an action are the same, the action is considered useless and a negative reward of -0.01 is given to the agent.

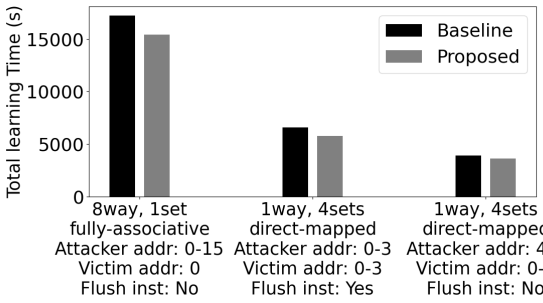

Figure 1: Comparison of Learning Time between Baseline (vanilla AutoCAT) and Proposed Method.

Figure 1 shows the learning time in seconds required for one learning trial in three different configurations. The x axis shows each configurations including the cache structure (the number of ways and sets) and the addresses that the attacker and victim can access. The y axis is averaged over 10 trials. The left bars (black) represent the learning time for vanilla AutoCAT, and the right bars (grey) represent the learning time for AutoCAT with our proposed method enabled. The learning time was reduced by 12.4 % in the best (middle) case while in the worst (right-most) case it was reduced by 7.4 %.

3 Future Work

Future work includes evaluating the proposed method for cache structures on real CPUs and efficiently finding the best negative reward value. We found that a too small negative reward value (e.g, -1.0) slows down a learning trial for some cache structures, thus we need a systematic way to find the best negative reward value.

References

[1] F. Liu, Y. Yarom, Q. Ge, G. Heiser, and R. B. Lee. 2015. Last-level cache side-channel attacks are practical. In *IEEE S&P*. 605–622.
[2] Mulong Luo, Wenjie Xiong, Geunbae Lee, Yueying Li, Xiaomeng Yang, Amy Zhang, Yuandong Tian, Hsien-Hsin S. Lee, and G. Edward Suh. 2023. AutoCAT: Reinforcement Learning for Automated Exploration of Cache-Timing Attacks. In *HPCA*. 317–332.

[1] https://www.encata.net/blog/overview-of-the-hardware-product-development-stages-explained-poc-evt-dvt-pvt

APSys'24, September 4 - 5, 2024, Kyoto, Japan
2024. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX