The Discrepancy of the Megaflow Cache in OVS: Final Episode

Levente Csikor[1], Vipul Ujawane[2], Dinil Mon Divakaran[3]

[1]National University of Singapore, email: levente.csikor@gmail.com
[2]Indian Institute of Technology, Kharagpur, email: vipul999ujawane@iitkgp.ac.in
[3]Truswave, email: dinil.divakaran@trustwave.com

Proposal for a Lightning talk

In the previous talks, we demonstrated that the Tuple Space Search (TSS) scheme, used for packet classification algorithm in the MegaFlow Cache (MFC) of OVS, has an algorithmic deficiency that can be abused by an attacker in different ways by pushing this generally high performing packet classifier to its corner case of degraded performance. We called this attack as Tuple Space Explosion (TSE). In TSE, a legitimately looking low-rate attack traffic (with no particular pattern) inflates the tuple space making the cardinal linear search process in TSS to spend an unaffordable time for classifying each packet; this eventually leads to a complete denial-of-service (DoS) for the users [2].

In the first part [4], we focused on a limited attack scenario. In particular, we demonstrated that for each set of flow rules, e.g., Access Control Lists (ACL), there exists a well-engineered traffic trace from which almost every packet creates a new tuple. We showed that the basic `Whitelist+DefaultDeny` ACLs tenants are typically given as default in cloud systems are particularly vulnerable. However, in order to carry out this attack, the adversary has to have access to or knowledge of the installed ACLs.

In Part II [3], we analyzed that when the attacker is not aware of the ACL, to what extent a randomized traffic trace can inflate the tuple space. Particularly, we showed that with less than 7 Mbps attack rate, significant degradation of 88% could be achieved.

Both works above, however, had one crucial aspect in common. We focused on one type of datapath, exclusively, namely the kernel datapath installed by the underlying system's own packet manager. In many real-world (production) environments, administrators simply rely on the built-in software tools to install applications to reduce or even completely avoid all the crux around manual installations and compilations from source code, e.g., via `apt-get install openvswitch-common` in Debian-based Linux distributions. Even though in most of the cases, we eventually end up having the same application with negligible (performance) difference, when applications also have modules supplied by the underlying kernel (e.g., in the case of Open vSwitch since the Linux 3.3 kernel debut in 2012 [5]), there can be significant deviations among the implementations. In particular, as it turned out after the discussions (with some of the OVS developers) during our previous talks, *(i)* the kernel networking stack developers do not prefer exact flow caching; therefore, the kernel datapath of OVS lacks the first-level Exact Match Cache (EMC). This means that the whole fast-path only comprises the MFC, thereby making TSE more efficient. On the other hand, *(ii)* while the userspace datapath provided by Intel's DPDK significantly improves the packet processing performance (by avoiding context-switching, interrupt-based packet handling, and the side-effects of OS schedulers), it essentially shares the same code base, and most parts of the algorithms are implemented according to the same original design.

Therefore, to round out our whole study around the discrepancy of the MFC, in this **lightning talk**, we investigate to what extent other datapaths are exposed to the TSE attack. First, *(i)* we scrutinize the kernel datapath of OVS compiled and installed from its up-to-date, "out-of-kernel-tree" source code, developed by the core OVS developers. We show that the additional caching layer of EMC can significantly increase the performance of OVS under the TSE attack, thereby

requiring the adversary to increase her attack rate to become successful. Subsequently, *(ii)* we also analyze the Intel DPDK-based userspace datapath (also known as `OVS-DPDK`), which is less vulnerable to TSE due to an efficient ranking algorithm in the tuple space introduced by patch in 2016 [1]. In essence, this ranking algorithm sorts the tuples according to the overall number of hits their entries have. Thus, whenever a packet of a frequent flow has to be classified, its corresponding tuple will be ranked higher, making the linear search process faster to find it. This renders the low-rate TSE attack much less efficient; in particular, after the attack commences, the victim flows slowly "climb back" to the front of the tuple space and their throughput resurge to higher values.

To counter this ranking algorithm, we propose TSE 2.0, which by letting some tuples expire and re-spawning them by carefully switching the original TSE attack on and off, keeps the ranking algorithm busy. Thus, eventually, TSE 2.0 causes a complete denial-of-service (DoS) for the users of the same software switch. Furthermore, we propose TSE 2.1 against `OVS-DPDK` running on multiple cores, wherein we slightly increase the attack rate of TSE 2.0, but, at the same time, we carefully adjust the packet sending sequence to achieve the same results as with TSE 2.0. We experimentally show that TSE 2.1 can still mount a low-rate DoS attack as long as `OVS-DPDK` is running on less than five cores.

References

[1] B. Bodireddy and A. Fischetti. OVS-DPDK Datapath Classifier. Intel Blogpost, `https://intel.ly/3kCbIi8`, October 2016 [Accessed: Oct 2020].

[2] L. Csikor, D. M. Divakaran, M. S. Kang, A. Korosi, B. Sonkoly, D. Haja, D. P. Pezaros, S. Schmid, and G. Rétvári. Tuple Space Explosion: A Denial-of-Service Attack Against a Software Packet Classifier. In *ACM CoNEXT 2019*, Dec 2019.

[3] L. Csikor, M. S. Kang, and D. M. Divakaran. The Discrepancy of the Megaflow Cache in OVS, Part II. Full talk at OVS+OVN Conference, `https://bit.ly/2SsfGh7`, Dec. 2019.

[4] L. Csikor and G. Rétvári. The Discrepancy of the Megaflow Cache in OVS. Full talk at OVS Fall Conference, `https://bit.ly/30A5qb9`, Dec. 2018.

[5] S. M. Kerner. Open vSwitch (OVS) Becomes a Linux Foundation Collaborative Project, Aug 2016 [Accessed: Jun 2020].