Dear OVSCON committee,

We are writing you to propose a full talk at the 6th Open vSwitch Conference 2019 (OVS+OVN '19 Conference) held on Dec 10 and 11 at Red Hat, Westford, MA, United States.

Please find the required details below:

- Title: **The Discrepancy of the Megaflow Cache in OVS, Part II**
- Names, email address, and affiliation for each speaker:
  Levente Csikor (levente.csikor@gmail.com)
  Senior Research Fellow
  NUS-Singtel Cyber Security Research and Development Laboratory, National University of Singapore, Singapore
- Type of talk: **Full talk**
- Further authors: Dinil Mon Divakaran (dinil.divakaran@trustwave.com), Min Suk Kang (kangms@nus.edu.sg), NUS-Singtel Cyber Security Research and Development Laboratory, National University of Singapore, Singapore

Extended Abstract

Open vSwitch (OVS) has stood the test of time in the field of OpenFlow (OF) software switching and its brand has been spreading virally in an unprecedented way; it is present in almost all (open source) networking environment starting from simple (Linux-based) operating systems through heavily virtualized Cloud Management Systems (e.g., OpenStack) to serverless environments (e.g., Kubernetes).

Due to Network Function Virtualization (NFV) and the ever increasing trend of offloading (business-critical) workloads to the public cloud, significant parts of the networking ecosystem (e.g., packet classification) have inherently become offloaded to virtualized packet processors (i.e., Open vSwitch). High traffic demands and latency-critical applications, on the other hand, require the packet classifier to be highly efficient and dependable. To support this, several years (and version numbers) ago, OVS has introduced a layered cache architecture in its fast past to achieve a reliable and blazing packet processing [1].

In our previous talk [2, 3], we have demonstrated that the packet classification algorithm in this caching architecture, namely the Tuple Space Search (TSS) scheme in the second level MegaFlow Cache (MFC), has an algorithmic deficiency that can be abused by an attacker to push this generally high performing packet classifier to its corner case. Particularly, we have shown that for each simple ACL (e.g., allow destination port 80 and drop everything else[1]) there is a specially crafted packet sequence that when subjected to this ACL in lower than 1 Mbps traffic rate, can virtually bring down OVS causing a complete denial-of-service for each workload accessible through that OVS instance, i.e., in a cloud environment, all services that happened to be scheduled to the same hypervisor become inaccessible. Note that such a service outage can cause millions of dollars for enterprises [4].

Our previous study, however, has some limitations: *(i)* an attacker exploiting this discrepancy has to be aware of the target ACL installed in the flow table and *(ii)* beside some immediate yet impractical remedies (e.g., switching MFC off, relying on a different hypervisor switch implementation instead), it is lacking of any countermeasure. Moreover, to easily support *(i)*, the threat model was limited to cloud environments, where an attacker has to lease her own resources, define her own (malicious) ACLs and send a specially crafted packet sequence towards her own service resulting in a denial-of-service attack against all co-located workloads only.

In this work, we have carried on with the main idea of [2] and we study whether an attacker can get rid of the above mentioned limitations (i.e., attacking arbitrary victims), and how its impact (e.g.,

[1] According to the cloud security best practice's `Whitelist + default deny` policy.

level of degradation, required packet rate) would change (if it is possible at all, see [5] for more details). Furthermore, we have developed a promising mitigation technique and we are reaching out to the OVS community to share and discuss the obstacles still present on the way to completely resolve this issue.

During our talk, first, we briefly cover the main properties of the caching architecture and the main outcomes of our previous study (670 kbps of attack traffic from a single traffic source can easily degrade a single OVS instance from its full capacity of 10 Gbps to 2 Mbps) that we term here as *co-located* case.

Then, we show that when an adversary has no such access to her target (e.g., neither leased resources in the cloud nor knowledge of the installed ACLs), she can still achieve a significant degradation of 88% from the maximum capacity with low attack traffic volume ($6.72 Mbps$). One interesting aspect of this latter approach (hereafter, termed as *general* case) is that it does *not* demonstrate any specific patterns of its attack traffic: it only requires *completely random* packet header fields and arbitrary message contents, along with arbitrary packet arrival times making the overall identification hard as it is not straightforward to define a specific signature of the attack traffic.

As a mitigation technique, we present a cache management scheme, which we call *MFC Guard* (`MFCg`), that dynamically monitors the number of entries in the MFC and *removes* less important ones to reduce the performance overhead of the TSS algorithm. It is worth noting that we have observed negligible impact on the overall packet processing performance during the monitoring process itself (i.e., executing `ovs-dpctl dump-flows` in, say, each second).

We show that `MFCg` can limit and even completely avoid the performance degradation for the packets that are eventually allowed to the system. However, nothing comes without a sacrifice: since removing a cache entry from the MFC results in the corresponding malicious flow's packets to be processed (again) by the slow path (i.e., `ovs-vswitchd`), this guaranteed performance of the allowed packets imposes some extra overhead. Contrary to the fact those packets should be cached again in the MFC, we have observed an unexpected behavior: such packets will always be processed by the slow path henceforth. Even if the overhead becomes constant with this conduct, as long as the attack rate is less than $1,000$ pps (< 1 Mbps) the slow path only consumes 15% of the CPU; recall, this packet rate is enough to bring down OVS in case of co-location [2]). However, when the packet rate is $10,000$ pps (< 7 Mbps), the CPU load jumps up to $\approx 80\%$ (this rate would be enough to degrade the full capacity to 10% in the general case). We can conclude that our current `MFCg` implementation can efficiently mitigate both attacks as long as the attacking rate is low, however further optimization can be carried out regarding the behavior of OVS. On the other hand, if the attack rate is much above $10,000$ pps, such an attack can be considered as a volumetric attack, for which there are multiple solutions to efficiently detect and mitigate (e.g., excess amount of packets and over-provisioning, scrubbing techniques).

References

[1] J. Pettit, "Accelerating Open vSwitch to "Ludicrous Speed"," Blog post: Network Heresy - Talses of the network reformation, https://networkheresy.com/2014/11/13/accelerating-open-vswitch-to-ludicrous-speed/, 2014.

[2] Levente Csikor and Gábor Rétvári, "The Discrepancy of the Megaflow Cache in OVS," Full talk at OVS Fall 2018 Conference, Dec. 2018.

[3] L. Csikor, C. Rothenberg, D. P. Pezaros, S. Schmid, L. Toka, and G. Rétvári, "Policy injection: A cloud dataplane dos attack," in *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos*, ser. SIGCOMM '18, 2018, pp. 147–149. [Online]. Available: http://doi.acm.org/10.1145/3234200.3234250

[4] Dan Kobialka, "Kaspersky Lab Study: Average Cost of Enterprise DDoS Attack Totals $2M," Blog post, https://www.msspalert.com/cybersecurity-research/kaspersky-lab-study-average-cost-of-enterprise-ddos-attack-totals-2m/, 2018.

[5] Levente Csikor and Dinil Mon Divakaran and Min Suk Kang and Attila Korosi and Balázs Sonkoly and David Haja and Dimitrios P. Pezaros and Stefan Schmid and Gábor Rétvári, "Tuple Space Explosion: A Denial-of-Service Attack Against a Software Packet Classifier," in *to appear at ACM CoNEXT 2019 Conference*, Dec 2019.