1 Review

The paper propose a fast automated approach for quickly detecting previously unknown worms and viruses based on two key behavioral characteristics - common sequence in the packets, and unique sources generating infections and destinations being targetted. The idea is simple and is to get the count of unique source and destination addresss for each substring, and report substrings with count greater than some threshold as potential worm signatures. While the idea is simple, the challenge is to execute it at high speed. The paper proposes interesting ideas to achieve this.

First, only those packets are considered whose content substrings appeared at least some x times. Multi-stage filters were used for this purpose and found to reduce memory footprint dramatically. The paper also propose usage of value sampling to consider fewer substrings to reduce CPU footprint as well. Next, address dispersion is quanitified for such candidate content sub-strings to reduce the false positives. Again, it could be done by using list or hashtable for each such content, but that would have high memory footprint and hence won't be efficient. The paper propose usage of scaled bitmap which does approximate counting and requires very less memory. The implementation could operate at 200 Mbps and the paper claims that the hardware implementations can scale up to 40 Gbps. The memory footprint was less than 4MB and can allow potential on-chip implementations.

The experiences with early bird, live or trace bases seem to be quite promising. However, the paper also states the concern that it could be evaded by attackers in many ways. First, different version of content with same semantic meaning can be used. Second, the invariant content may appear across multiple packets. While the paper says that multiple packets from the same flow can be coalesced, and accordingly substrings can be used, but that would require keeping per flow state and hence would be inefficient. Further, the attacker could spoof the source address of the packet to reflect coming from same address. Also, the attacker can impact other packets by using the commonly occurring content in them and hence making other packets appearing potential worm signatures. One way this may cause to drop good packets and impact the corresponding applications, and the other way it would increase the number of false positives. Despite of all these limitations, the experience with Earlybird had been good, and it would significantly raise the bar of worm authors.

In terms of performance studies, there seems to be some issues. The memory footprint due to content prevalence table is just 2 MB. Considering, that every packet is being checked for content hashing, this footprint seems to be small. Number of false positives just due to hash collisions can be more at higher speed, since number of packets will become very large. This may question the total memory footprint to be with in limit of SRAM size.