Sophisticated network hacks have existed since the 70s, but the uptake of internet connectivity in the last decade brought a barrage of automated remote attacks. Protection against these worms and viruses is largely based on port-filtering, selective virus-scans and other endhost-based defences. Such methods miss many potential attacks and need knowledgeable users for proper operation.

More effective filtering can be achieved in the network using datastream analysis, as performed by most network Intrusion Detection Systems (nIDS). The current generation is, however, considered too crude and slow for in-line placement as an Intrusion Prevention System (IPS). High-speed processing is limited to per-packet string-matching, which is useless against polymorphic attacks, and even then datarates rarely grow out of the 10s of Mbits/s.

Therefore we are developing an IPS that can perform more detailed data inspection, using information from all protocol levels, at much higher rates. The core of our system is Streamline, an OS networking subsystem that reduces avoidable overhead such as copying and context-switching and integrates all available hardware (checksum offload, SIMD cores, NPU, FPGAs). Streamline optimises the network dataplane per-application by overlaying a graph of functions over the graph of execution environments (application, kernel, NIC, ...). It then interconnects functions through methods that minimise costly data-transfer and asynchronous communication. To act as an IPS we have developed 3 novel functions in Streamline.

Even moderately sophisticated intrusion attempts will employ IP and TCP fragmentation to circumvent per-packet filters. Many IDSs cannot reassemble flows because copy overhead makes the task too costly to perform. To decrease this processing cost we have developed a method of TCP reassembly based solely on pointer movement, instead of copying. Stream reassembly acts as a first-layer guard against attacks by sanitising the input stream that applications receive.

To replace string-matching we have developed Ruler, a filter that performs more effective regular expression (regex) matching. Ruler is geared for high-speed traffic and merges all patterns in a single deterministic finite automaton and matches all of them at the same time, a prerequisite as pattern-sets run in the 10s of thousands. To process at multi-Gigabit rates we compile DFAs to FPGAs and/or network processors. Streamline takes care of compiling, shipping and setting up code on these devices.

Ruler improves low-level detection, but it cannot use host-based information to find malicious traffic. A third function merges network- and host-based methods by applying protocol-specific filters in the network. Essentially, a home-grown signature generating honeypot known as Argos detects which protocol field is responsible for a buffer overflow and determines what the maximum length is for the field beyond which it is certain to overflow a critical value. It does so without knowledge of the application or its source. Using this information, Streamline is able to apply a highly efficient protocol-specific filter capable of stopping polymorphic zero-day overflow attacks at link rate.

Security is further tightened by combining these homegrown methods with externally developed functions that do flow-based behavioural detection, such as counting the number of outgoing IP addresses, or monitoring traffic entropy.

The proposed IPS works at all layers. It performs per-packet header filtering to stop the simplest attacks, per-stream payload inspection for finding known exploits, behavioural detection to find otherwise stealthy intrusions and application-specific input sanitisation to block hard-to-detect polymorphic worms. Streamline’s graph-based structure ensures that new algorithms and hardware devices are easily incorporated so that the system can be continuously improved both in terms of effectiveness and performance.