

CLAIMS

What is claimed is:

1. A computer-implemented method comprising:
obtaining a packet filter definition that specifies characteristics of packets;
compiling the packet filter definition to instructions of low-level code;
generating a hash template based on a subset of the instructions, wherein the hash template includes pairs of byte offsets and byte counts that define locations within the packets at which the characteristics are disposed;
based on application of the hash template to a plurality of stored packets, creating hash table entries in a hash table, wherein the hash table entries are respectively associated with subsets of the stored packets, wherein the subsets of the stored packets have respectively unique patterns of values within their bytes at the locations defined by the pairs of byte offsets and byte counts;
updating the hash table entries to refer to metadata relating to their respective subset of the stored packets;
storing the hash table in association with the metadata;
receiving a new packet filter definition; and
in response to receiving the new packet filter definition, retrieving, from the hash table, particular metadata referred to by a particular hash table entry that is associated with a particular subset of captured packets, wherein the new packet filter definition matches: the locations defined by the pairs of byte offsets and byte counts, and the unique patterns of values of the particular subset of captured packets.
2. The computer-implemented method of claim 1, wherein the packet filter definition is a textual representation of one or more values that can be present in packet headers or payloads.
3. The computer-implemented method of claim 1, wherein the instructions of low-level code are assembly-language instructions executable by a virtual machine.

4. The computer-implemented method of claim 3, wherein the subset of the instructions are load instructions that read from specific byte offsets within packet headers or payloads.

5. The computer-implemented method of claim 4, wherein generating the hash template comprises:

 parsing the instructions to identify the load instructions; and
 generating one of the pairs of byte offsets and byte counts from each of the load instructions.

6. The computer-implemented method of claim 1, wherein the metadata referred to by the hash table entries include packet counts or byte counts for the associated subsets of stored packets.

7. The computer-implemented method of claim 1, wherein the metadata referred to by the hash table entries includes storage addresses or per-packet timestamps for the associated subsets of stored packets.

8. The computer-implemented method of claim 7, wherein the storage addresses are multi-dimensional.

9. The computer-implemented method of claim 1, further comprising:
 storing at least part of the hash template, in association with the hash table, as a representation of the packet filter definition.

10. The computer-implemented method of claim 1, wherein the packet filter definition is stored as a packet filter representation in association with the hash table, the computer-implemented method further comprising:

 determining, from the packet filter representation, that the new packet filter definition matches the locations defined by the pairs of byte offsets and byte counts;

 based on the packet filter representation, identifying the hash table;

generating, for the hash table entries, skeleton packets having the respectively unique patterns of values within bytes at the locations defined by the packet filter representation; and

determining, from the skeleton packets, that the new packet filter definition matches the unique patterns of values of the particular subset of captured packets.

11. A non-transitory computer-readable medium storing program instructions that, when executed by one or more processors of a computing system, cause the computing system to perform operations comprising:

- obtaining a packet filter definition that specifies characteristics of packets;
- compiling the packet filter definition to instructions of low-level code;
- generating a hash template based on a subset of the instructions, wherein the hash template includes pairs of byte offsets and byte counts that define locations within the packets at which the characteristics are disposed;
- based on application of the hash template to a plurality of stored packets, creating hash table entries in a hash table, wherein the hash table entries are respectively associated with subsets of the stored packets, wherein the subsets of the stored packets have respectively unique patterns of values within their bytes at the locations defined by the pairs of byte offsets and byte counts;
- updating the hash table entries to refer to metadata relating to their respective subset of the stored packets;
- storing the hash table in association with the metadata;
- receiving a new packet filter definition; and
- in response to receiving the new packet filter definition, retrieving, from the hash table, particular metadata referred to by a particular hash table entry that is associated with a particular subset of captured packets, wherein the new packet filter definition matches: the locations defined by the pairs of byte offsets and byte counts, and the unique patterns of values of the particular subset of captured packets.

12. The non-transitory computer-readable medium of claim 11, wherein the packet filter definition is a textual representation of one or more values that can be present in packet headers or payloads.

13. The non-transitory computer-readable medium of claim 11, wherein the instructions of low-level code are assembly-language instructions executable by a virtual machine.

14. The non-transitory computer-readable medium of claim 13, wherein the subset of the instructions are load instructions that read from specific byte offsets within packet headers or payloads.

15. The non-transitory computer-readable medium of claim 14, wherein generating the hash template comprises:

parsing the instructions to identify the load instructions; and
generating one of the pairs of byte offsets and byte counts from each of the load instructions.

16. The non-transitory computer-readable medium of claim 11, wherein the metadata referred to by the hash table entries include packet counts or byte counts for the associated subsets of stored packets.

17. The non-transitory computer-readable medium of claim 11, wherein the metadata referred to by the hash table entries includes storage addresses or perpacket timestamps for the associated subsets of stored packets.

18. The non-transitory computer-readable medium of claim 17, wherein the storage addresses are multi-dimensional.

19. The non-transitory computer-readable medium of claim 11, further comprising:
storing at least part of the hash template, in association with the hash table, as a representation of the packet filter definition.

20. The non-transitory computer-readable medium of claim 11, wherein the packet filter definition is stored as a packet filter representation in association with the hash table, the operations further comprising:

determining, from the packet filter representation, that the new packet filter definition matches the locations defined by the pairs of byte offsets and byte counts;

based on the packet filter representation, identifying the hash table;

generating, for the hash table entries, skeleton packets having the respectively unique patterns of values within bytes at the locations defined by the packet filter representation; and

determining, from the skeleton packets, that the new packet filter definition matches the unique patterns of values of the particular subset of captured packets.