From Packet to Flow: Network Security Algorithms to Break Bottleneck

Jun Li
Research Institute of Information Technology
School of Information Science and Technology
Tsinghua University, Beijing, China

Many contributions from my colleagues and students, especially Yaxuan Qi, Bo Xu, and Xin Zhou
Outline

Why from Packet to Flow?

Features and Bottlenecks
- Packet Classification
- Stateful Inspection
- Deep Inspection

Algorithms and Performance
- Fast Packet Classification: AggreCuts
- Efficient State Management: SigHash
- High Performance Content Inspection: MRSI

Summary
Why from Packet to Flow?

- Increasing sophistication of applications
  - Stateful inspection firewalls
  - Deep inspection in IDS/IPS
- Continual growth of network bandwidth
  - OC192 or higher link speed
  - Millions of concurrent connections
- Requirement for holistic defense
  - Against complex and blended network threats
  - Integrated security features in unified security architecture
  - Unified Threat Management (UTM)
Features and Bottlenecks

- Packet Classification
  - High-speed with modest memory
- Stateful Inspection
  - Large number of connections
  - Order-preserving
- Deep Inspection
  - Enormous signatures
  - Various signature characteristics
Novel Algorithms (1)

- Packet classification algorithm (AggreCuts)
  - Aggregation Cuttings
  - Multi-dim range match
  - Worst-case bounded and adjustable
    - Limited decision tree depth
    - No linear search
  - Efficient memory storage
    - Space aggregation with bitmap
    - Support different memory hierarchies
Packet Classification Algorithms

Field-independent Search Algorithms

Trie-Based Algorithms
- BV
- ABV
- AFBV

Table-Based Algorithms
- Prefix Match
- Equivalent Match
- CP
- Index Search
- Binary Search

Trie-Based Algorithms
- H-Trie
- SP-Trie
- HSM
- GoT

Decision-Tree Algorithms
- Bit-Test
- Range-Test
- Modular
- Single-Field
- Multi-Field

AggreCuts

Network Security Lab, RIIT
Tsinghua University
Space Aggregation

- Space Aggregation

Cuttings

Rule 1

Rule 0

Rule 2

Original Pointer Array

HABS

Compressed Pointer Array

Network Security Lab, RIIT
Tsinghua University
Decision-tree

Data-structure

<table>
<thead>
<tr>
<th>Bits</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:30</td>
<td>dimension to Cat (d2c)</td>
<td></td>
</tr>
<tr>
<td>29:28</td>
<td>bit position to Cat (b2c)</td>
<td></td>
</tr>
<tr>
<td>27:20</td>
<td>8-bit HABS</td>
<td></td>
</tr>
<tr>
<td>19:0</td>
<td>20-bit Next-Node CPA Base address</td>
<td></td>
</tr>
</tbody>
</table>

- d2c=0: src IP; d2c=1: dst IP;
- d2c=10: src port; d2c=11: dst port.
- b2c=00: 31–16; b2c=01: 23–16; b2c=10: 15–8; b2c=11: 7–0
- if w=8, each bit represent 32 cuttings; if w=4, each bit represent 2 cuttings.

The minimum memory block is 278*4 Byte. So if w=8, 20-bit base address support 128MB memory address space; if w=4, it supports 3MB memory address space.

Network Security Lab, RIIT
Tsinghua University
AggreCuts vs. HiCuts

Performance Evaluation

- Memory Usage:
  - an order of magnitude less

- Memory Access:
  - 3~8 times less

- Throughput on IXP2850:
  - 3~5 times faster
Novel Algorithms (2)

- Stateful inspection algorithm (SigHash)
  - Signature based hashing
    - Support large concurrent connections
    - Efficient memory usage
    - High speed TCP handshakes
  - Per-flow packet order preserving
    - External Packet order preserving
    - Internal Packet order preserving
Signature-based Hashing

- $m$ signatures for $m$ different states with same hash value
- Resolving collision in SRAM (fast, word-oriented)
- Storing states in DRAM (large, burst-oriented)

5-tuple Header

<table>
<thead>
<tr>
<th>SIP 32bits</th>
<th>DIP 32bits</th>
<th>SP 16bits</th>
<th>DP 16bits</th>
<th>Prot 8bits</th>
</tr>
</thead>
</table>

Onchip CRC Unit

Hash Value

Signature

Hash Index

Signature Table in SRAM

Flow State Entry Table in DRAM

Flow State 1 Flow State 2 Flow State 3 Flow State 4

Sig1 Sig2 Sig3 Sig4
SigHash Performance

- Throughput
  - 10Gbps
  - (SRAM+DRAM)
  - 8Gbps
  - (DRAM only)
- Connections
  - 10M on IXP2850
- Collision
  - Less than 1%
  - Depends on different load factors
Handshake-separated Hash (IntelliHash)

- Process handshake packets in SRAM, data packets in DRAM, sharing the same hash value
- Speedup session creation
- Enhance anti-DoS capability
IntelliHash Procedure

- **Handshake packets processing**
  - Process SYN/SYN_ACK packets in SRAM
  - Process ACK packets in DRAM; if (LEN==zero && session!=exist), process in SRAM Zone
**IntelliHash**

**Performance Evaluation**
- Handshake packets processing speed
  - 8.5G (IntelliHash)
  - 6.5G (DirectHash)
- Session Creation Rate
  - Up to 2M connections per second (IntelliHash)
Per-flow Packet Ordering

- Packet Order-preserving
  - Typically, only required between packets on the same flow.

- External Packet Order-preserving (EPO)
  - Sufficient for processing packets at network layer.
  - Fine-grained workload distribution (packet-level)
  - Need locking

- Internal Packet Order-preserving (IPO)
  - Required by applications that process packets at semantic levels.
  - Coarse-grained workload distribution (flow-level)
  - No need for locking
Per-flow Packet Ordering

External Packet Order-preserving (EPO)

- Ordered-thread Execution
  - Ordered critical section to read the packet handles off the scratch ring
  - The threads then process the packets, which may get out of order during packet processing
  - Another ordered critical section to write the packet handles to the next stage

- Mutual Exclusion by Atomic Operation
  - Packets belong to the same flow may be allocated to different threads to process
  - Mutual exclusion can be implemented by locking
  - SRAM atomic instructions
Per-flow Packet Ordering

- Internal Packet Order-preserving (IPO)
  - SRAM Q-Array
  - Workload Allocation by CRC Hashing on Headers
Per-flow Packet Ordering

- Performance Evaluation
  - Throughput
    - EPO is faster, 10Gbps
    - IPO has linear speed up, 7Gbps
  - Workload Allocation
    - Hashing via On-chip CRC
    - Nearly balanced workload

Network Security Lab, RIIT
Tsinghua University
Novel Algorithms (3)

- **RSI (Recursive Shift Indexing)**
  - Reduce the number of useless matching
  - **Pro**: trade-off space with time
    - Directly using four-character block to create the BLT will use memory up to $256^4 \rightarrow 4$ GB
Bitmaps are used for preprocessing and deleted after that.
RSI Temporal Performance

![Graph showing RSI Temporal Performance](image-url)
RSI Spatial Performance

![Graph showing memory occupation vs pattern number for different patterns: AC, AC-BM, WM, SBMH, RSI. The graph illustrates a linear relationship between the pattern number and memory occupation.](image)
Break the Real Bottleneck

- Current version of Clam-AV
  - The basic signatures are handled by BMEXT that uses the last 3 characters of a signature to generate shifts

- Large dataset characteristics
  - ClamAV: 78k basic rules

- Our proposal: hybrid algorithms
  - DFA for short signatures: DFA-based algorithm implemented on fast on-chip memory
    - Space efficient
    - High performance (5.5G vs 1.2G on Octeon)
  - HASH for long signatures: Hash-based algorithm with larger shifts than BMEXT
    - Search with shifts/skips: i.e. MRSI
DFA Performance Limit

DFA size = 100MB, Len=512Byte
1.2Gbps on Octeon 3860

DFA size = 100KB, Len=512Byte
5.5Gbps on Octeon 3860
## Statistics of ClamAV Signatures

<table>
<thead>
<tr>
<th>Idx</th>
<th>Total Number</th>
<th>Average Length</th>
<th>Min Length</th>
<th>Len&lt;9 Num</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>29611</td>
<td>67.5</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>46954</td>
<td>123.7</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>164</td>
<td>106.8</td>
<td>28</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1402</td>
<td>110.7</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>355</td>
<td>46.6</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>n/a</td>
<td>n/a</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>105.1</td>
<td>17</td>
<td>0</td>
</tr>
</tbody>
</table>

- Large scale signature set
- Longer average length
- Very few short signatures
MRSI

- Use three BLTs
  - Increase the probability of getting leap
- Omit Phase 2 in original RSI data structure
  - Solve memory occupation expansion
  - Improve preprocessing speed
MRSI Performance

MRSI vs. BMEXT: Scanning Speed

MRSI vs. BMEXT: Memory Usage
MRSI Performance

MRSI vs. BMEXT: Scalability

MRSI vs. BMEXT: Performance under Attacks
MRSI Performance in AV

Real System Performance on Clam-AV
Summary

- Analyze the real problem
  - Packet classification
  - Stateful Inspection
  - Deep Inspection
- Propose new algorithms
  - Hardware aware
  - Time-space tradeoff
- Break the real bottleneck
Reference


Thanks
http://security.riit.tsinghua.edu.cn