On the Safety and Efficiency of Virtual Firewall Elasticity Control

Juan Deng†, Hongda Li†, Hongxin Hu†, Kuang-Ching Wang†, Gail-Joon Ahn‡, Ziming Zhao‡ and Wonkyu Han‡

NDSS 2017
Outline

- Introduction
- Overview of VFW Controller
- Our Approach
  - Dependency Analysis and Semantic Consistency
  - Flow Update Analysis
  - Buffer Cost Analysis
  - Optimal Scaling
- Implementation and Evaluation
Traditional Hardware-based Firewall

- Fixed location
- Constant capacity
Virtualized Environments

**Blur & Fluid Perimeters**

**Virtualized Network Zones**

Datacenter

Infrastructure

Service Migration
Traffic Volume Variation

Expensive option: capacity ≥ peak traffic load

Source: https://blog.cloudflare.com/a-winter-of-400gbps-weekend-ddos-attacks/
New Trends

- **Network Function Virtualization (NFV)**
  - Create and destroy software instances dynamically
- **Software-Define Networking (SDN)**
  - Dynamic traffic steering

NFV + SDN → Virtual Firewall
Firewall as a Service

- Virtual firewall in **commercial virtualized environments**
  - Amazon AWS
  - VMware vCloud
  - VCE Vblock
  - Microsoft Azure
  - Google Cloud Platform

- Virtual firewall used to protect **traditional enterprise networks**
  - Middlebox Outsourcing [SIGCOMM’ 12]
Elastic Virtual Firewall Scaling

- Overload → elastic scaling out
- Underload → elastic scaling in

Safe, Efficient and Optimal
Policy Migration in VFW Scaling

**Copy**

<table>
<thead>
<tr>
<th>Seq</th>
<th>Src_ip</th>
<th>Dst_ip</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>A</td>
<td>B</td>
<td>Deny</td>
</tr>
<tr>
<td>v2</td>
<td>C</td>
<td>D</td>
<td>Allow</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- VFW_1
- VFW_2

**Split**

<table>
<thead>
<tr>
<th>Seq</th>
<th>Src_ip</th>
<th>Dst_ip</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>A</td>
<td>B</td>
<td>Deny</td>
</tr>
<tr>
<td>v2</td>
<td>C</td>
<td>D</td>
<td>Allow</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- VFW_1
- VFW_2

Virtual Firewall Performance VS. Rule Size

Graph showing processing capacity (Mbps) versus number of firewall rules.
Challenges - Semantic Consistency

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Src_ip</th>
<th>Dst_ip</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>10.10.1.5</td>
<td>192.1.1.*</td>
<td>Deny</td>
</tr>
<tr>
<td>v2</td>
<td>10.10.1.*</td>
<td>192.1.1.9</td>
<td>Allow</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Safety Problem:** Allow illegal traffic

10.10.1.5 → 192.1.1.9

**Dependent**
Challenges - Correct Flow Update

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Src_ip</th>
<th>Dst_ip</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>10.10.*.5</td>
<td>192.1.2.*</td>
<td>Allow</td>
</tr>
<tr>
<td>v2</td>
<td>10.10.*.6</td>
<td>192.1.1.*</td>
<td>Deny</td>
</tr>
<tr>
<td>v3</td>
<td>10.10.*.7</td>
<td>192.1.1.*</td>
<td>Allow</td>
</tr>
<tr>
<td>v4</td>
<td>10.10.*.8</td>
<td>192.1.1.*</td>
<td>Deny</td>
</tr>
</tbody>
</table>

Flow Table

<table>
<thead>
<tr>
<th>Seq</th>
<th>Src_ip</th>
<th>Dst_ip</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1</td>
<td>10.10.1.*</td>
<td>192.1.1.*</td>
<td>To VFW₂</td>
</tr>
<tr>
<td>f2</td>
<td>10.10.1.*</td>
<td>192.1.2.*</td>
<td>To VFW₁</td>
</tr>
</tbody>
</table>

Incorrect delivery

Not exactly matched

10.10.1.5 → 192.1.2.9
10.10.1.6 → 192.1.1.9
10.10.1.7 → 192.1.1.9
10.10.1.8 → 192.1.1.9
Challenges - Buffer Overflow Avoidance

VFW\textsubscript{1} \quad \text{Not enough buffer (Packet loss)} \quad \text{VFW}\textsubscript{2}

\textbf{How much buffer is required?}

\textbf{In-flight Traffic}

SDN Switch

R/S

Challenges - Optimal Scaling

- **Goal:** minimum resource consumption
  - Scaling-out: *least* new instances
  - Scaling-in: *most* killed instances

- **Constraints**

- Satisfy SLAs
- Minimize Update
- Avoid Buffer Overflow
Overview of VFW Controller

VFW Controller

- Condition Detection
- Dependency Analysis
- Flow Update Analysis
- Buffer Cost Analysis
- Optimal Scaling Calculation
- Migration and Update Control
- Provision Control

Resource

- Virtual Firewall
- Virtualization Layer
- Servers
- SDN Switches
Dependency Analysis

Packet Space:

Direct dependency

Indirect dependency

<src_ip, dst_ip, src_port, dst_port, protocol>

r1:<10.10.1.*, 192.1.1.9, any, any, TCP>

Direct dependency

r2:<10.10.2.*, 192.1.1.*, any, any, TCP>

Direct dependency

r3:<10.10.2.5, 192.1.2.*, any, any, TCP>

PS(r1)

PS(r3)

Intra-dependency
Dependency Analysis

Relation between Firewall and Flow rules

Firewall Rule Group (V)

Flow Rule Group (F)

Inter-dependency

Congruence

\[ PS(V) = PS(F) \]

Superspace

\[ PS(V) \supset PS(F) \]

Subspace

\[ PS(V) \subset PS(F) \]

Intersection

\[ PS(V) \cap PS(F) \subset PS(V) \]
\[ PS(V) \cap PS(F) \subset PS(F) \]
Semantic Consistency

Causes

1. Group is broken
2. Order is not preserved

Group-based Migration

Group 1

- VFW 1

Group 2

- VFW 2

Group 1

- VFW 1

Group 2

- VFW 2
Flow Update Analysis

**V**: firewall rule group to be migrated

**F**: flow rule group inter-dependent with **V**

**Congruence**

\[ PS(V) = PS(F) \]

**Superspace**

\[ PS(V) \supset PS(F) \]

”CHANGE” all \( f_i \in F \)

**Subspace**

\[ PS(V) \subset PS(F) \]

**Intersection**

\[ PS(V) \cap PS(F) \subset PS(V) \]

\[ PS(V) \cap PS(F) \subset PS(F) \]

”CHANGE” or ”INSERT”
Flow Update Analysis

For each \( v_i \in V, f_j \in F \)

\[ PS(v_i) \cap PS(f_j) = \emptyset \]

No Update

or

\[ PS(v_i) = PS(f_j) \]

“CHANGE” \( f_j \)

or

\[ PS(v_i) \supseteq PS(f_j) \]

“INSERT” \( f'_j \) where \( PS(f'_j) = PS(v_i) \cap PS(f_j) \)
Buffer Cost Analysis

\[ \beta = (\sum \lambda) \times \{d_1 + d_3 - d_2 + b_1 + b_2\} \]

\(b_1, b_2\) : bandwidth
\(d_1, d_2, d_3\) : transmission delay
\(\lambda\) : sending rate of the affected flows
Optimal Scaling Calculation

\[ X = \{x_{11}, \ldots, x_{mn}\}, \quad x_{ij} \in \{0, 1\} \text{ are indicators} \]

■ Goals

**Scaling-out**

firewall rule group \( V_i \) is moved to instance \( j \) \( \rightarrow x_{ij} = 1 \)

\[ \min \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \gamma_i \]

Minimize extra instances

■ Constraints

**Solved by**

**Integer Linear Programming**

\[ \max \sum_{j=1}^{n} \sum_{i=1}^{m} x_{ij} \]

Maximize merged instances

■ Satisfy SLAs

Minimize Update

Avoid Buffer Overflow
Implementation

- Implementation
  - Xen-4.4.1, ClickOS [NSDI’14]
  - Floodlight, Open vSwitch
  - Simple stateful firewall: 7 new Click elements, ~3000 lines of C++.
  - VFW Controller: python interface, based on Hassel Library [NSDI’12]

- Testbed
  - CloudLab (https://www.cloudlab.us/)
  - Experiment profile is available:
    https://www.cloudlab.us/p/SeNFV/Firewall-VLANs
Intra-dependency in real-world firewall policies

<table>
<thead>
<tr>
<th>Policy</th>
<th>Rule(#)</th>
<th>Group(#)</th>
<th>Largest Group Member (#)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>12</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>18</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>D</td>
<td>52</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>E</td>
<td>83</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>F</td>
<td>132</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>G</td>
<td>354</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>H</td>
<td>926</td>
<td>13</td>
<td>18</td>
</tr>
</tbody>
</table>
Evaluation

- Capability to quickly scale

Split with UDP flow overload

Split with TCP flow overload

< 1 second
Evaluation

- Migration impact on throughput

Throughput Degradation

Impact on UDP throughput

Throughput Degradation

Impact on TCP throughput
Evaluation

Performance of optimal scaling calculation

6 addition virtual firewall instances, 1000 firewall rule groups to split

Scaling-out calculation

100 underloaded virtual firewall instances

Scaling-in calculation
Conclusion

● NFV+SDN push forward a new breed of firewalls, *virtual firewalls*

● VFW Controller enables *safe, efficient* and *optimal* virtual firewall scaling

● Implementing and evaluating VFW Controller
This work was partially supported by grants from National Science Foundation (NSF-ACI-1642143 and NSF-ACI-1642031)
State Of The Art

- Safe Migration
  - Split/Merge [NSDI’13]
  - OpenNF [SIGCOMM’14]

- NFV and SDN for security
  - Bohatei [USENIX Security’15]

- SDN Firewall
  - FlowGuard [HotSDN’14]

- Firewall policy deployment [S&P’07]

- Distributed firewall [CCS’00]
## Study of Real-world Firewall Policies

<table>
<thead>
<tr>
<th>Policy</th>
<th>Rule(#)</th>
<th>Group(#)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>52</td>
<td>7</td>
</tr>
<tr>
<td>E</td>
<td>83</td>
<td>9</td>
</tr>
<tr>
<td>F</td>
<td>132</td>
<td>10</td>
</tr>
<tr>
<td>G</td>
<td>354</td>
<td>10</td>
</tr>
<tr>
<td>H</td>
<td>926</td>
<td>13</td>
</tr>
</tbody>
</table>

Rule Dependencies in Real-world Firewall Polices
Rule size impact on performance
### Challenges (Semantic Consistency)

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Src_ip</th>
<th>Dst_ip</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>10.10.1.5</td>
<td>192.1.1.*</td>
<td>Deny</td>
</tr>
<tr>
<td>v2</td>
<td>10.10.1.*</td>
<td>192.1.1.9</td>
<td>Deny</td>
</tr>
<tr>
<td>v3</td>
<td>10.10.1.*</td>
<td>192.1.1.*</td>
<td>Allow</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Src_ip</th>
<th>Dst_ip</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>v2</td>
<td>10.10.1.*</td>
<td>192.1.1.9</td>
<td>Deny</td>
</tr>
<tr>
<td>v3</td>
<td>10.10.1.*</td>
<td>192.1.1.*</td>
<td>Allow</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Original VFW**

10.10.1.5 → 192.1.1.1

**Merged VFW**

10.10.1.5 → 192.1.1.1

Semantic Consistency