Dynamic Security Orchestration for CDN Edge-Servers

Elaeheh Jalalpour, Milad Ghaznavi, Daniel Migault, Stere Preda, Makan Pourzandi, and Raouf Boutaba
David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada
Ericsson Research, Montreal, QC, Canada

Introduction and Motivation

- Content Delivery Networks (CDNs) provide high QoE in delivering digital content
- CDNs cache content in edge-servers in the vicinity of end-users
- Attacks against CDN edge-servers deteriorate QoE
- Limitations of current defense mechanisms
  - Hardware security functions
  - Expensive, not elastic, not flexible
  - Scrubbing centers
  - Restriction latency, proprietary mechanisms
- Current software solutions
  - Massive changes in infrastructure, not-automated deployment, exclusive to DDOS

Dynamic Security Orchestration

- Automatic and dynamic deployment and flexible modification of security services
- Instantiating and modifying security chains in reaction to events
- Inspecting only suspicious traffic by security services
- Classifying traffic at the beginning of the chains

Demo 1: Network Layer Rate Limiting

<table>
<thead>
<tr>
<th>Context</th>
<th>Security Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack</td>
<td>TCP flooding attacks</td>
</tr>
<tr>
<td>Impacts</td>
<td>Exhasting bandwidth resources</td>
</tr>
<tr>
<td>Defense</td>
<td>Per IP rate-limiting</td>
</tr>
</tbody>
</table>

Demo Setup

- A cluster of machines
  - 16-GB RAM
  - 8-cores 3.30-GHz Xeon CPU
  - 10-Gbps NIC.

Attack Emulation

<table>
<thead>
<tr>
<th>Stage</th>
<th>Flooding traffic share</th>
<th>Active traffic generators</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0%</td>
<td>Traffic Gen. 1</td>
</tr>
<tr>
<td>2</td>
<td>50%</td>
<td>Traffic Gen. 1 and 2</td>
</tr>
<tr>
<td>3</td>
<td>66.6%</td>
<td>Traffic Gen. 1, 2, and 3</td>
</tr>
<tr>
<td>4</td>
<td>75%</td>
<td>Traffic Gen. 1, 2, 3, and 4</td>
</tr>
</tbody>
</table>

Mitigation Chain

- Rate-Limit
- Virtual Edge Server
- Traffic Sink
- Security Chains
- Virtual Infrastructure Manager
- Orchestrator

Conclusion

- We demonstrated a configurable security system that protects CDN edge-servers
- This system behavior is governed by high-level policies
- The deployment of security function chains is dynamic and automatic
- We illustrated how our system can be flexibly programmed to mitigate real-world threats
- In first demonstration, our system mitigates a network layer flooding attack
  - Deploying a chain of a rate-limiting function recovering legitimate traffic
- In second demonstration, an application layer abusive behavior is rate-limited
  - Deploying a chain of a TLS termination and a WAF to rate-limit abusive requests

Architecture Components

- Orchestrator
  - Programed by policies scripted in an adapted version of F# language
  - Reacts to environment states and threats
- Virtual Infrastructure Manager (VIM)
  - Manages the resources of a virtual edge-server
  - Provides a north-bound API for Orchestrator and SMAS
- Security Monitoring Analytics System (SMAS)
  - Monitors and analyzes the collected data
  - Feeds the orchestrator with alerts that may trigger security actions