Leveraging SDN for Efficient Anomaly Detection and Mitigation on Legacy Networks

Kostas Giotis

Network Management & Optimal Design Laboratory (NETMODE)
School of Electrical & Computer Engineering
National Technical University of Athens

European Workshop on Software Defined Networks
September 3rd, 2014
Budapest, Hungary
Malicious traffic growth

- Companies, governments and institutions are increasingly targeted by surges of malicious traffic
  - **Target:** Service discovery & disruption

- DDoS attacks become more sophisticated
  - Exploitation of common protocols (DNS, NTP)
  - High traffic volumes, through Amplification Techniques
DDoS attacks

- Access Control Lists
  - Waste network resources
  - High-end equipment
  - High administrative costs

- Remote Triggered Black Hole (RTBH)
  - Propagate a null route to all iBGP peers
  - Requires less human intervention
  - Victim becomes unreachable

The shift towards amplification attacks may point to a new trend.

Amplification techniques produce attacks of higher volume

The RTBH Approach

Mitigate the attack and avoid collateral damage from malicious traffic

- Malicious traffic going to the Victim is dropped
- Prevent the waste of valuable resources, such as uplink bandwidth
- Benign traffic is dropped as well
- “Drop” is the only option
- Matching is based on Destination IP
CAN WE EXPLOIT SDN PROGRAMMABILITY TO ACHIEVE BETTER MITIGATION RESULTS?
Extending RTBH with OpenFlow

- Victim’s traffic is redirected to an OF switch
- Controller identifies malicious flows
- The Controller instructs the switch to drop malicious flows
- Benign Traffic is forwarded back, through the inport of the OF switch

✓ Matching traffic on a per-flow basis
✓ Benign Traffic is forwarded towards the Victim
✓ Preserve normal operation and reachability of the Victim
✓ Malicious traffic going to the Victim is dropped
✓ Prevent the waste of valuable resources, such as uplink bandwidth
Key properties of the proposed approach

- Match offending traffic on a per-flow level
  - Benign packets are still delivered to the victim

- Modular architecture design
  - Decoupling of the required functions such as:
    - Data gathering
    - Anomaly Detection & Victim Identification
    - Attack Mitigation

- Automated Triggering of the RTBH device
  - Reduce administrative costs

- Packet sampling capabilities
  - Native OF statistics collection does not scale
  - High-end equipment is avoided
A modular architecture approach

- Monitoring Service
  - sFlow (random packet sampling)
- POX Controller
- Trigger Router (ExaBGP)

Anomaly Detection/Identification
- Statistics Collection
- Anomaly Detection
- Victim Identification

Anomaly Mitigation
- Identify Malicious Flows
- Forward benign flows
- Drop malicious flows

RTBH Component
- Propagate Next-Hop route to redirect the traffic of the victim to an OF switch

Host under attack

No

Yes

Identify Malicious Flows
Experimental Setup

Victim

Exploiting software router

DDoS detection

Victim identification

Trigger router

ExaBGP

Forwarding logic of victim’s traffic - Attack mitigation

sFlow collector

Malicious traffic segregation

POX controller

Legacy Ethernet switch

OVS

Quagga software router

sFlow protocol

Victim’s traffic

RTBH device

sFlow collector

Network Management & Optimal Design Laboratory
DDoS Detection & Victim Identification

- Fine-grained DDoS Detection
  - Bidirectional count sketch algorithm
    - Efficiently store a summary of large data
    - Count sFlow samples based on Dst IP

- Victim Identification
  - Locate top Dst IP addresses with highly asymmetric communication pattern
  - Eliminate IP addresses that do not surpass a specific threshold

- Evaluation based on real traffic data
  - Captured DDoS attack
    - CAIDA DDoS Attack 2007 Dataset
  - Benign traffic captured from NTUA
Malicious Traffic Segregation

- Packet symmetry metric employed as proof of concept
  - you can choose your own algorithm

- For TCP connections: $1 \leq \frac{recv}{sent} \leq 4.5$

- For UDP traffic: similar approach, but site-dependent
We know the victim, we know the bad flows.. Now what?

- **OpenFlow offers Network Programmability**
  - Control flows that were redirected to the OF switch
  - Malicious flows are dropped
  - Benign flows are forwarded back through the inport (OFPP_IN_PORT OpenFlow action)

<table>
<thead>
<tr>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
<th>ACTION</th>
<th>PRIORITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>In port</td>
<td>ETHER</td>
<td>IP</td>
<td>Port</td>
<td>Out Port</td>
<td></td>
</tr>
<tr>
<td>src</td>
<td>dst</td>
<td>type</td>
<td>src</td>
<td>dst</td>
<td>Prot</td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>M</td>
<td>X</td>
<td>*</td>
<td>*</td>
<td>0x0800</td>
<td>S</td>
</tr>
</tbody>
</table>
Does it scale? Still more to do..

- DDoS attacks involve thousands of malicious flows
  - Experiment with Hardware OF switches that use expensive TCAMs
  - Investigate a Longest Common Prefix approach to aggregate bad flows

- Deploy a multilevel Anomaly Detection method
  - We do not need detailed flow inspection all the time
Thank you!

Kostas Giotis
coyiotis@netmode.ntua.gr