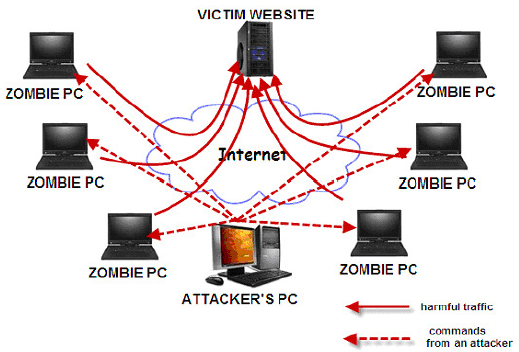
**Network Traffic Analyzer**

Richard Antiabong, Cody Eilrich, Rohit Malaga

**Background & Motivation**

Network traffic information generated from tools such as tcpdump or mirrored switch ports can be analyzed for malicious activity. However the processing demands to analyze raw traffic data may be intensive- especially when one considers analyzing all of the data coming into an enterprise with geographically redundant data centers. Therefore we propose using the real-time, distributed processing capabilities of Apache Storm to handle such a task.

DoS (Denial of Service) and related DDoS (Distributed Denial of Service) attacks (see Figure 1) are common in today’s networks. They are used to exhaust a network’s finite computing resources so that legitimate users are prevented from accessing the network resources. We developed a distributed solution to detect these kinds of attacks using the real-time processing capabilities of Apache Storm. Our solution receives raw network traffic, performs analysis and presents findings to the network managers using a graphical dashboard. The network managers may use the dashboard to send back network management commands back to network elements such as routers and firewalls in order to block offending parties’ access to network resources.

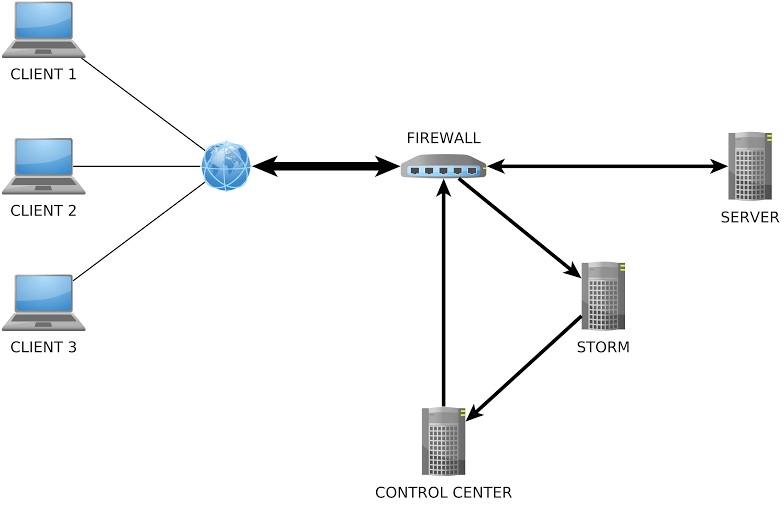


**Figure 1: DDoS Attack Structure**

We specifically implemented a solution for two types of DoS & DDoS attacks: ICMP and SYN flooding. However the functionality may of course be extended to address more threats.

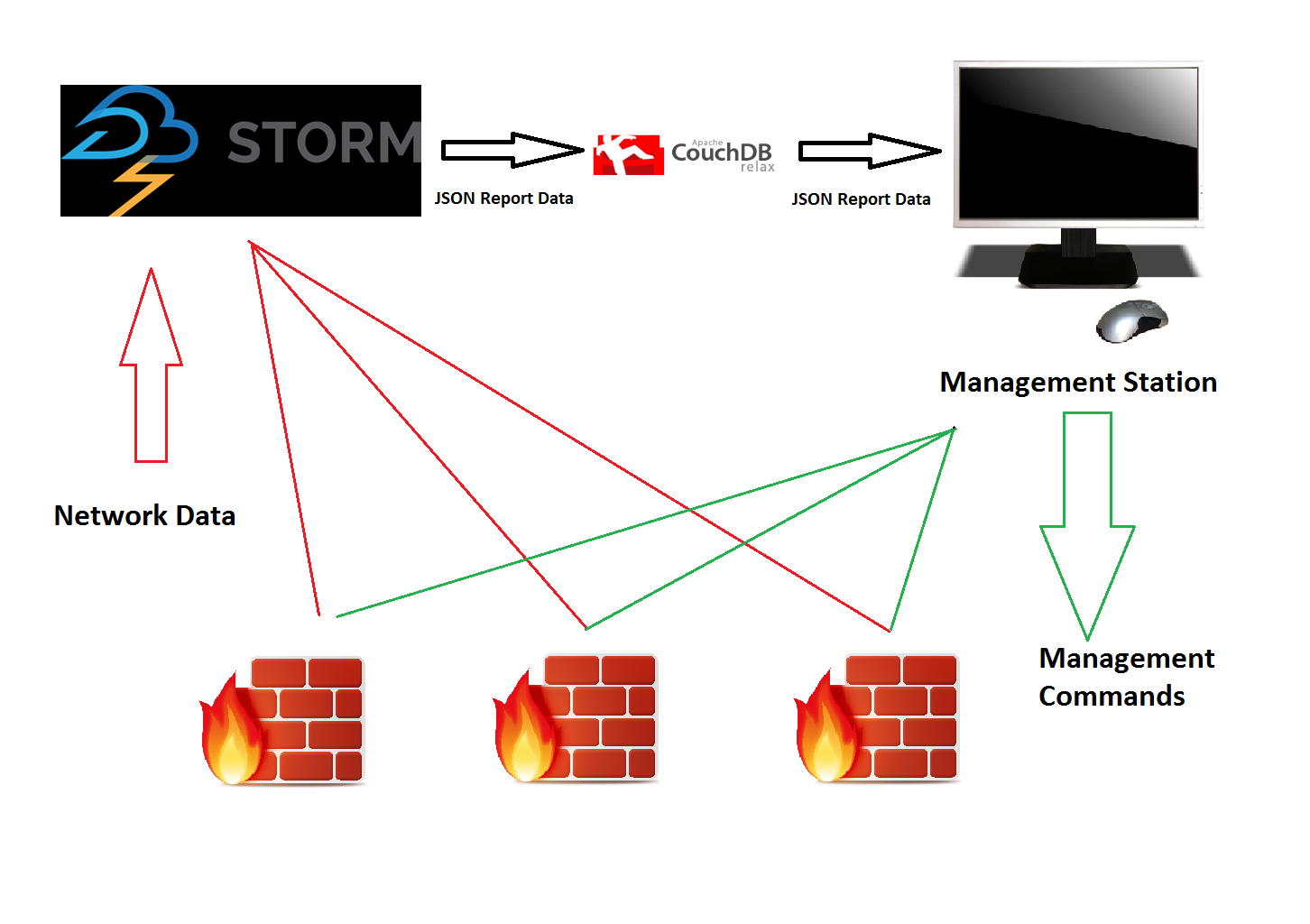
**Architecture**

We created a system to send raw tcpdump data from a virtual router to Apache Storm through a named pipe. Spouts and bolts are defined to process each IP packet. Overall and per source-IP statistics are recorded each second. Apache Storm provides traffic analysis and returns a decision deeming each source address to be legitimate or suspect. The per-second statistics and heuristic results are formatted into a platform independent JSON report and sent to CouchDB for storage.



**Figure 2: Physical Architecture**

The manager application retrieves the JSON statistics data from CouchDB and builds graphs of the SYN and ICMP attacks. Overall stats are displayed along with the largest offenders with an option to block them. The manager contains a library that logs in via ssh to the network elements and sends a command to modify the iptables on the system (either block/unblock an offending IP). The manager is accessible from any device over the Internet.



**Figure 3: High Level Software Design**

**Challenges**

Initially we had hoped to set up the manager to use SNMP to block offending source IPs. Unfortunately though, the devices we had access to did not include MIBs that would have supported us in this activity. We addressed this challenge by issuing iptables commands to block IPs over SSH.

Getting started with Apache Storm was no simple task. The various processes (Supervisor, Nimbus, Workers, et. al.) and architecture of Apache Storm was not intuitive for us. We also had issues getting Storm to run our traffic analyzer topology in production mode so that it could be monitored in the Storm UI. We were finally able to correct these issues after resolving some nasty Apache HttpClient dependency conflicts and disabling IPv6 in the Worker processes.

Finally the graphing tools also posed great challenges for us. Retrieving the JSON data from CouchDB and formatting it into a presentable format was also difficult. We overcame these challenges by displaying the traffic information not in graphs but in textual traffic reports supported by Javascript.