Subject

Distributed Denial of Service (DDoS) are a threat to Internet services ever since the widely published attacks on e-bay.com and amazon.com in 2000. ETH itself was the target of such an attack 6 months before these commercial sites where hit. ETH suffered repeated complete loss of Internet connectivity ranging from minutes to hours in duration. Massively distributed DDoS attacks have the potential to cause major disruption of Internet functionality up to and including severely decreasing backbone availability.

Attack Model

Most DDoS attacks share a common pattern: An infection phase where the initiator acquires the attack resources by compromising a large number of weakly protected hosts, ideally causing little or no visible change in host behavior, in order to make the compromise hard to notice. An infection phase can last from less than 10 minutes to several months. Attacks within the order of 100.000 and more compromised hosts have already been observed in practice (Code Red, Sapphire).

In a second phase, the attack phase, the attacker uses the compromised hosts to initiate actual attacks on a target computer or network. These attacks can be done autonomously or under direct or indirect control of the attacker. Although attack control increases the risk of identification for an attacker, there are possibilities to keep this risk small.

Diploma Thesis Task

A generic realtime UDP Netflow data processing framework will be developed and extensively tested to assure proper operation under high loads and bursty Netflow data traffic as observed in the Switch backbone network infrastructure.

This task is split into the following subtasks:
Understand network traffic raw data

As raw data of the network traffic, we use NetFlow\(^1\) traffic logs that are produced by border gateway routers of the SWITCH\(^2\) academic network. In a first step, the format of this data and its exact meaning must be understood.

Analyse the Samplicator program

Currently, SWITCH uses a program called “Samplicator” to queue the UDP Netflow data, process it and then duplicate it to two other machines. The Netflow traffic stream is currently very bursty in its nature, which is even aggravated by duplicating the data streams. In the processing step, the “Samplicator” converts the Netflow format and overwrites some data fields in the Netflow records. The source code of this program is available.

Design and Implementation of a realtime UDP Netflow data processing framework

The functionality of the “Samplicator” should be greatly improved. The result will be a realtime UDP Netflow data processing framework, which consists of modular components that very likely will be run on distributed machines.

In this thesis at least the following components of the framework will be developed:

- The *queueing component* de-bursts the Netflow data by using an efficient leaky-bucket algorithm for emptying the queue.

- The *statistics component* calculates the most essential traffic statistics.

- The *duplicating component* duplicates the Netflow data stream and sends it to 1-N hosts. The number and properties (IP address, port etc.) of the destination hosts can be configured without stopping data acquisition.

- The *logging component* writes the raw Netflow data into files.

- The *monitoring and alerting component* reads the Netflow data and shows a selection of important statistics and activates alerts upon detection of critical traffic patterns (based on given traffic signatures).

Further optional components are:

- Graphical plots of aggregated statistics over time extracted from Netflow data

  etc.

\(^1\)A proprietary data standard by CISCO  
\(^2\)www.switch.ch
Testing

Test scenarios with artificial bursts and high loads are to be conceived and executed in order to assure the realtime capabilities of this framework.

Documentation and Presentation

A documentation that states the steps conducted, lessons learnt, major results and an outlook on future work and unsolved problems has to be written. The code should be documented well enough such that it can be extended by another programmer within reasonable time. At the end of the semester, a presentation will have to be given at TIK that states the core tasks and results of this semester thesis. If important new research results are found, a paper might be written as an extract of the thesis and submitted to a computer network and security conference.

Dates

The diploma thesis will start on May 5th, 2003 and will be finished by September 5th, 2003.

Supervisors

Arno Wagner, wagner@tik.ee.ethz.ch, +1 632 7004, ETZ G64.1
Thomas Dübendorfer, duebendorfer@tik.ee.ethz.ch, +1 632 7196, ETZ G64.1