DDoS Security Testing

MIKE BERKELAAR & AZAD KAMALI (UVA)
SUPERVISOR: PIETER WESTEIN (DELOITTE)
SUMMER 2014

A (D)DoS Attack

- Is an attempt to make a service/resource unable to operate as intended
- Called “Distributed”, when more than one attackers are involved
- Comes from no where!
  - Distributed
  - Spoofed sources
- Hard to differentiate from legitimate usage

Types of DoS

- Disrupting Services
  - Configuration Information (DNS Poisoning)
  - State Information (disassociation in Wi-Fi)
  - Cutting Communication Path
- (Over)Consuming Valuable Resources
  - Bandwidth
  - Processing Time
- We will be focusing on the 2nd category

Defensive Measures

- Have more resources than attacker(s) (easy to say!)
- Make use of some in-line filtering devices
- Be prepared
  - Monitor behaviors
  - Dump logs and USE them
  - Test your infrastructure
    - What would it do under pressure?

Research Question

- How can various DoS attacks be simulated in a controlled way?
  - Which DoS attacks can be simulated in a potentially controlled way?
  - Which parameters should be used in order to have a controlled attack?
  - Which metrics should be monitored to measure the effects of a DoS

- Use-case
  - Test effects of potential DDoS attacks
  - Identify bottlenecks

Attack Layers

- Network Layer
  - Targeting Bandwidth of target and all nodes in the path to it
    - Ping of death
    - Amplification attacks

- Application Layer
  - Targeting Application specific aspects and/or TCP stack of OS
    - Massive (fake) HTTP requests
    - Heavy queries against Database servers
    - SYN Attack

When is the attack successful?

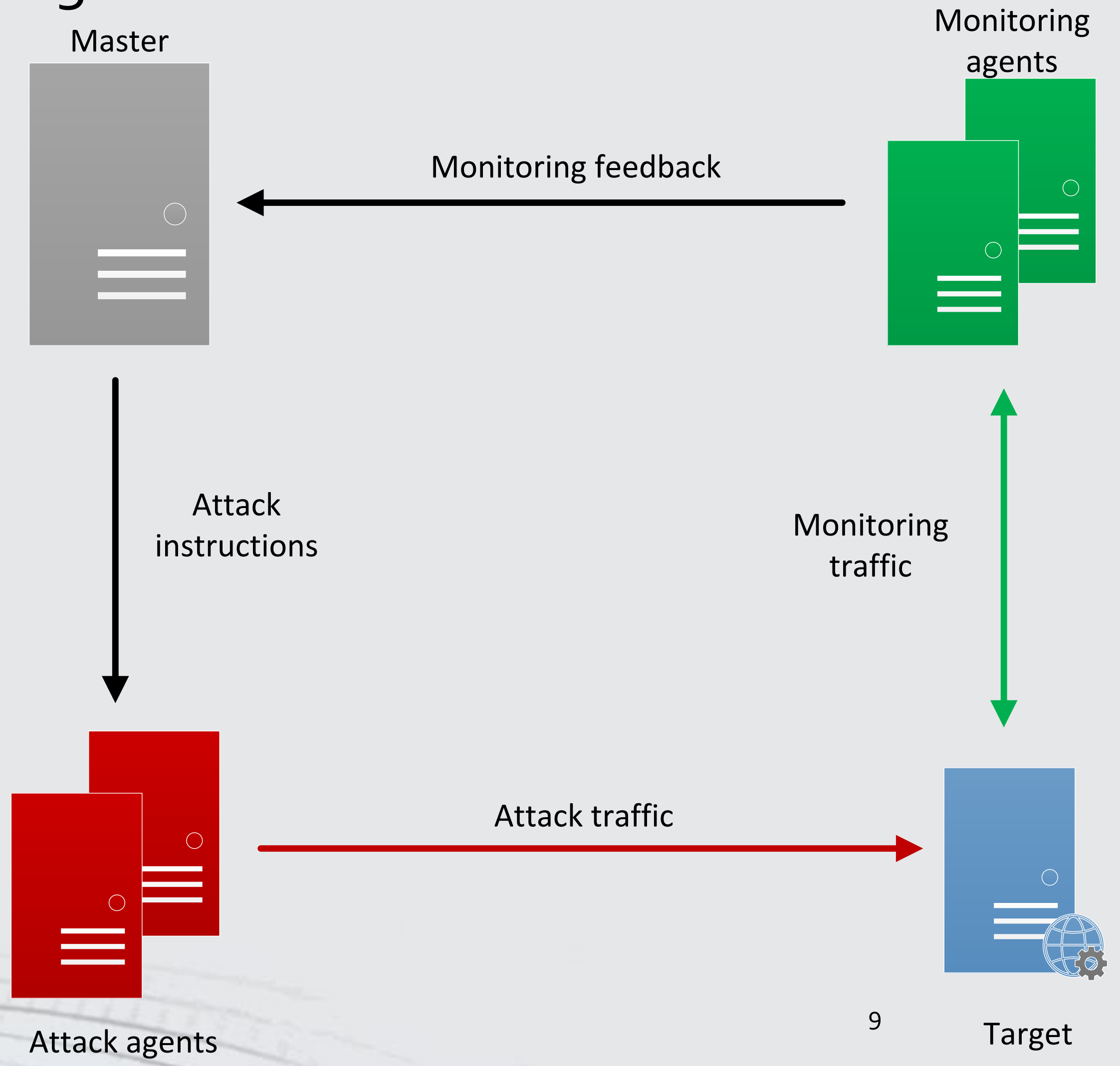
- When target is slowed down?
- When it is out for a while?
- When it is completely unavailable?

Basic Idea

- Based on feedback loops
  - Start a potential attack
  - Monitor the affects on target (get feedback)
  - Stop the attack at a certain point

Architecture

- Separation of monitoring and attacking
- Distributed execution
  - Performance
  - Monitoring consensus
- Extendable with various DoS attacks



Monitoring parts

- Resources
  - Remaining TCP queue space
  - System resource utilization

- Data Gathering
  - Resource status gathering via
    - SNMP
    - WMI
    - Other local daemons
  - RTT (ICMP, HTTP)
  - Timeouts (ICMP, HTTP)

Attack monitoring

- Monitoring (un)availability is a concern

- Monitoring accuracy may be off

Attack monitoring

- Reactive
  - Monitor if a defined threshold is reached
  - ‘Damage’ may have been done already
- Proactive
  - Watching trends could allow for predictions
  - Obvious choice if applicable
- Deal with noise and variance

Threshold Selection

- Different expectations
  - Performance Degradation
  - Partial unavailability
  - Complete unavailability

- Thresholds used in our tests:
  - 1 % random packet loss
  - 10 x response time regression

Proof of concept

- Python implementation of framework
- DDoS simulations
  - Traffic flood
  - Application level DoS
  - SYN flood

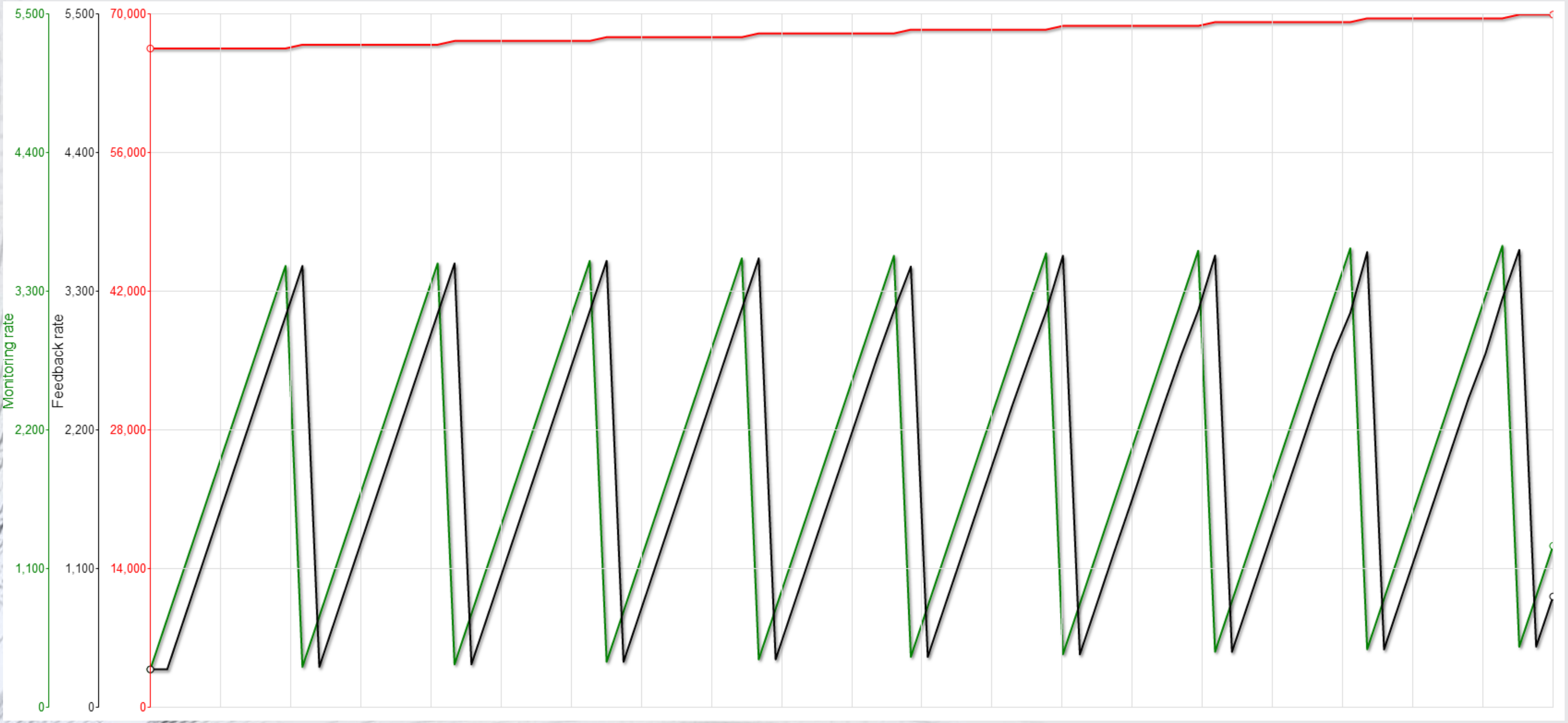
Traffic flood

- Exhaust network capacity

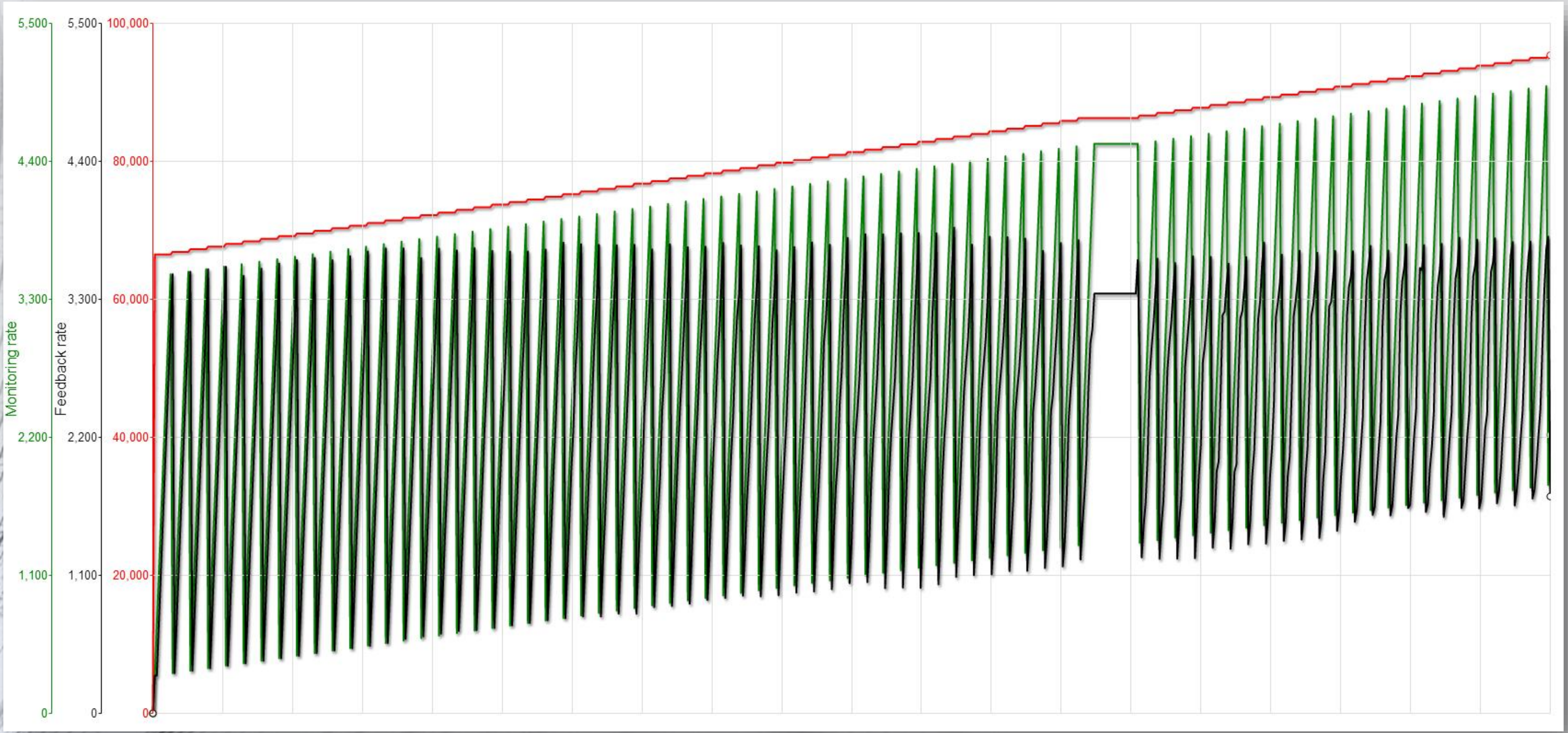
- Monitoring acts as a part of the attack
  - Probes for link capacity with ICMP packets
  - Hands off confirmed 'capacity' to attack-agents
  - Sliding rate as a percentage of the total attack rate

- Approximation of packet loss based on monitoring results

Traffic flood handoff



Traffic flood



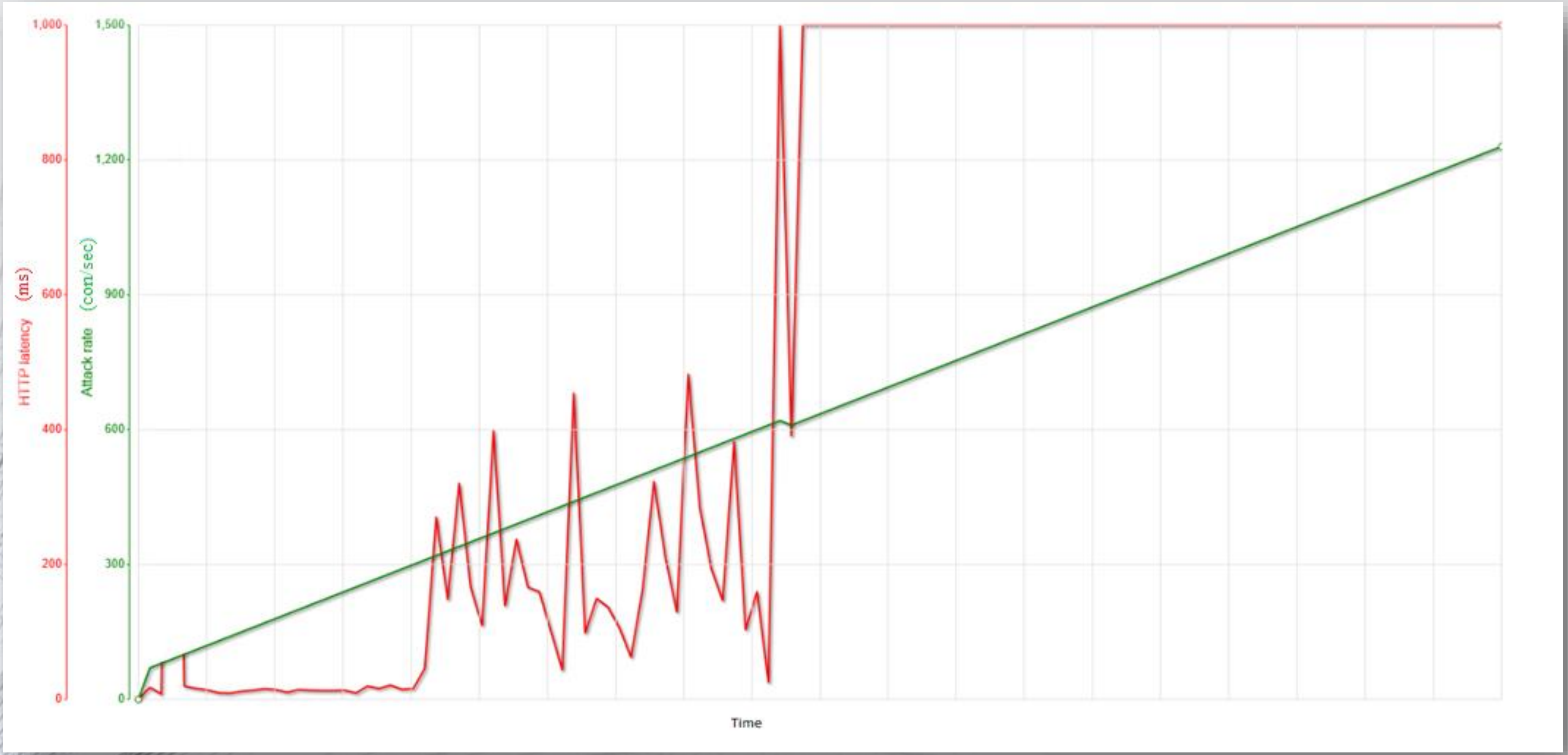
Application layer DoS

- Resource intensive script requested over HTTP

- Monitor HTTP response time
  - Values increase with attack rate

- Prediction of attack headroom based on response time slope

Application layer DoS



Conclusion

- DDoS attacks are controllable, depending on:
  - The definition of when a DDoS causes ‘damage’
  - The monitoring capabilities an attack class allows

Demo

- Controlled traffic flood demo