Flooding Denial of Service attacks detection on the source client hosts

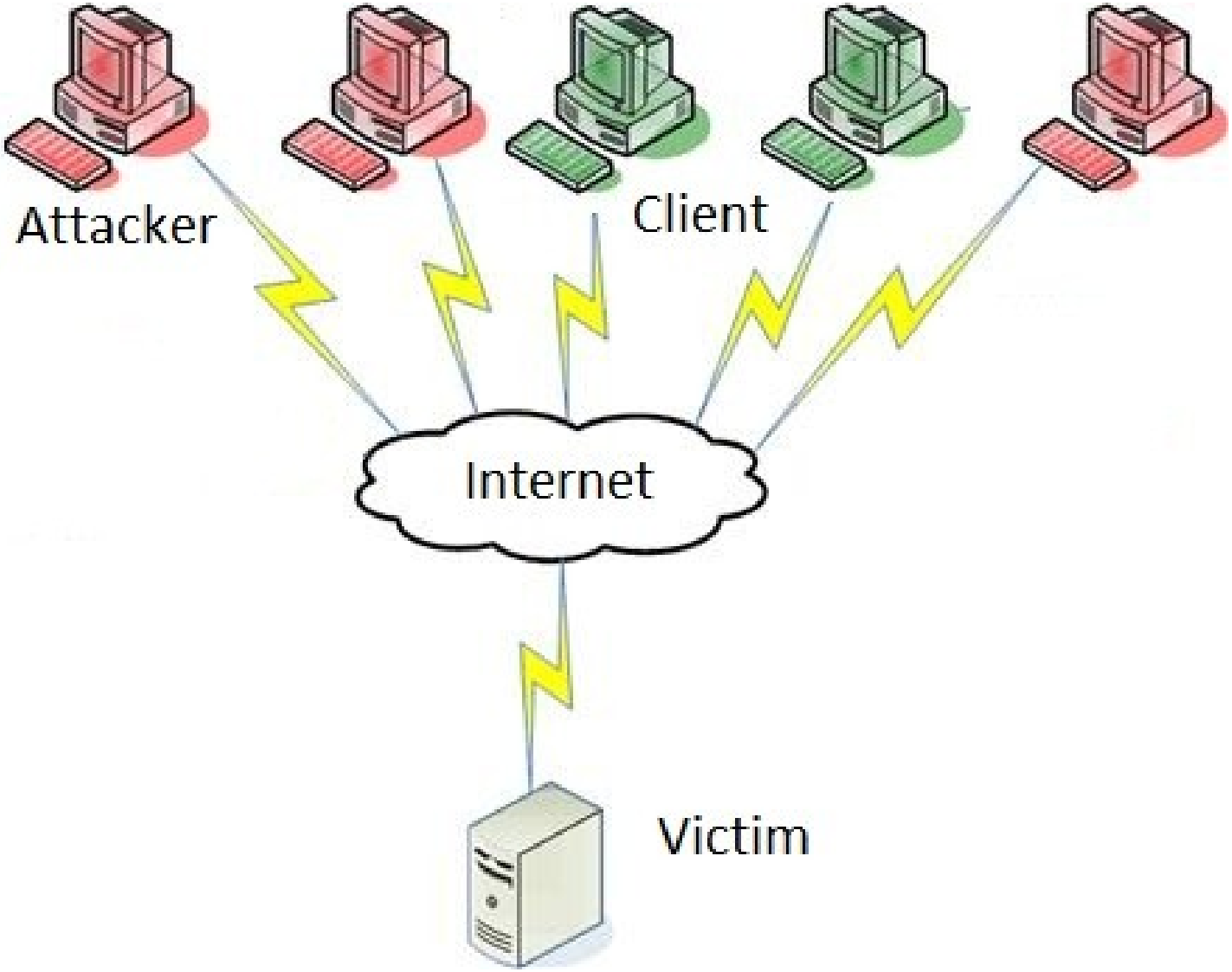
Vít Bukač

Definitions

- Denial of service attack (DoS)
  - An attempt to make a computer resource unavailable to its intended users
- Flooding DoS
  - Flooding attacks are performed by initiating a vast amount of seemingly legitimate transactions.

Aim of the thesis

- Design a new flooding denial-of-service attacks detection method for client hosts
- Measure its effectiveness


Attacker
Client
Internet
Victim

State of the Art – source network

- D-WARD: A Source-End Defense against Flooding Denial-of-Service Attacks
  - Sending rate/response rate packet ratio
- Filtering spoofed traffic at source end for defending against DoS / DDoS attacks
  - Simultaneous connections
  - Learning period + attack period

State of the Art – source host

- DDoSniffer: Detecting DDOS Attack at the Source Agents
  - Low packet connections, number of new connections, IP spoofing
  - Limited class of attacks, fixed thresholds
- A P2P-Based Distributed Detection Scheme against DDoS Attack
  - Packet count, traffic volume
  - Collaborative

Why source client hosts?

Advantages

- Congestion avoidance
- Small collateral damage
- Easy traceback
- Sophisticated algorithms
- Forensics data
- Access encrypted payload

Disadvantages

- Effectiveness
- Deployment incentive
- Hidden traffic
- Agent shutdown

Detection method

- Input: Host's network traffic features
- What: standalone detection engine, 2 stages
  1. Exclude demonstrably benign traffic
     Anomalies forward to second stage
  2. Verify anomalies with complex algorithms
- Output: DoS attack Y/N, process ID

Research outcome

- Papers
  - Study of usability of existing methods for source host detection
  - Description of the method incl. evaluation
  - Extension for low-rate and pulsing DoS attacks
- A prototype tool implementing the method

Conclusion

- Research a new Denial of service attacks detection method for source client hosts
  - 2-stage architecture with multiple processing algorithms
- Compare with existing solutions

- So far unpopular, rarely explored