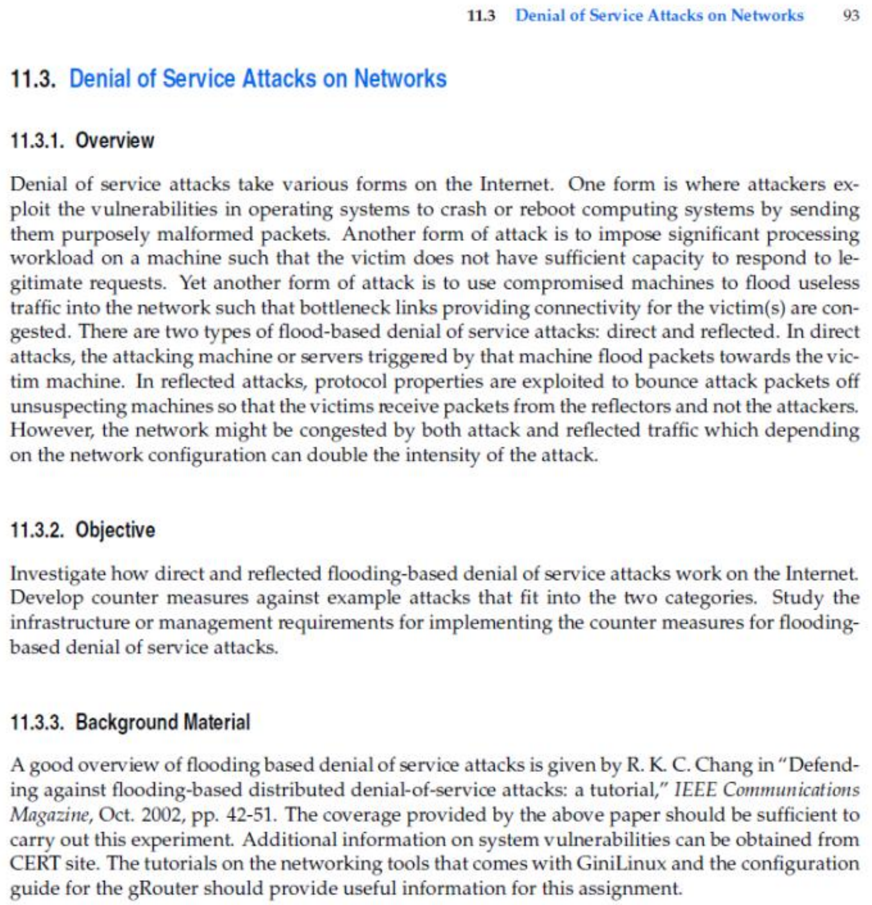
DoS Assignment

Monday, November 02, 2009
11:34 AM



11.3. Denial of Service Attacks on Networks

11.3.1. Overview

Denial of service attacks take various forms on the Internet. One form is where attackers exploit the vulnerabilities in operating systems to crash or reboot computing systems by sending them purposely malformed packets. Another form of attack is to impose significant processing workload on a machine such that the victim does not have sufficient capacity to respond to legitimate requests. Yet another form of attack is to use compromised machines to flood useless traffic into the network such that bottleneck links providing connectivity for the victim(s) are congested. There are two types of flood-based denial of service attacks: direct and reflected. In direct attacks, the attacking machine or servers triggered by that machine flood packets towards the victim machine. In reflected attacks, protocol properties are exploited to bounce attack packets off unsuspecting machines so that the victims receive packets from the reflectors and not the attackers. However, the network might be congested by both attack and reflected traffic which depending on the network configuration can double the intensity of the attack.

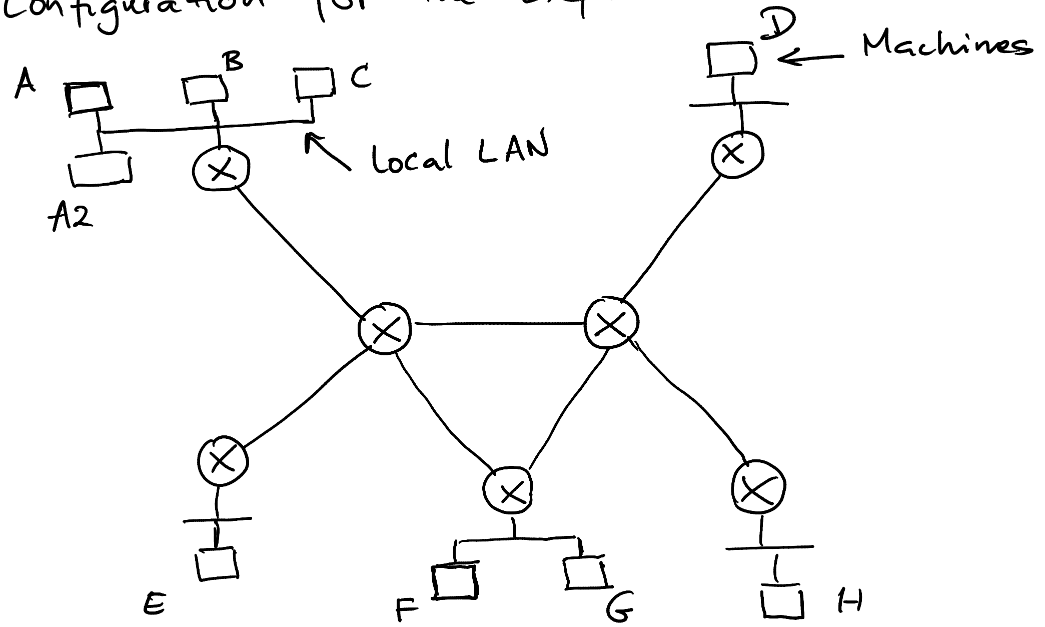
11.3.2. Objective

Investigate how direct and reflected flooding-based denial of service attacks work on the Internet. Develop counter measures against example attacks that fit into the two categories. Study the infrastructure or management requirements for implementing the counter measures for flooding-based denial of service attacks.

11.3.3. Background Material

A good overview of flooding based denial of service attacks is given by R. K. C. Chang in "Defending against flooding-based distributed denial-of-service attacks: a tutorial," *IEEE Communications Magazine*, Oct. 2002, pp. 42-51. The coverage provided by the above paper should be sufficient to carry out this experiment. Additional information on system vulnerabilities can be obtained from CERT site. The tutorials on the networking tools that comes with GiniLinux and the configuration guide for the gRouter should provide useful information for this assignment.

Configuration for the Experiment:



Measure the bandwidth between stations

E&D.

Run iperf -i 1 -s -p 5000 @ E in server mode at port 5000. This should give reports every 1 second.

Run iperf -t 600 -c IP_addr-of-E @ D to measure the band width between E&D.

The reports on available band width should appear on E every 1 second.

Attack 1: Launch a "smurf" attack. This attack exploits a feature where a machine responds to a reply request (e.g., ping) to a multicast/broadcast location. By default most machines (including the UMLs) do not reply to multicast reply requests. For the purposes of this experiment, activate this feature on the UMLs using the sysctl command.

⇒ [sysctl -a | grep broad ⇒ find the icmp option to change use sysctl -s]

Launch the smurf attack and measure its impact on B_{DE} (available BW between D&E)

Attack2: TCP SYN attack from one machine on another. Use the hping2 tool to launch the attack. Using the -i option you can set the repeat interval for the packet stream. Set this to a value close to -i u10000. If this value is close to -i u10000. If this value is set small -i u50, then too many packets will flood the netwrok and packet transmission will halt due to saturated queues.

Launch the TCP SYN attack and measure the impact.

Attack 3: Use ping or hping2 to launch a direct and reflected attack on the machines. Launch a coordinated attack from multiple machines to multiple targets (not more than 3 attack streams).

Measure the impact of the attack on the available BW between D & E.

In the second part, you need to develop counter measures for the attacks. There are two different strategies

① Filters

② Class-based queueing

Using Filters, you can remove attack traffic from the netwrok. What types of filters are applicable for the above atack situations?

[illegible] the possible filters to limit

Implement the possible filters to limit the attacks.

The class based queues (unlike filters) do not explicitly remove traffic. The CBQ method aggregates traffic into specific queues.

By defining queue sizes and the traffic characterizations accordingly, the impact of the attacks can be reduced.

Design the most effective counter measure using

(a) only filters

(b) only using class based queues.

Effectiveness of the countermeasure is measured by the bandwidth made available between D & E.