metropolitan
SH.P.A.L "UNIVERSITETI METROPOLITAN TIRANA"

Paul Neumann
pneumann@umt.edu.al

Low-intensity DoS attacks on BGP infrastructure

One need not fear superior numbers
if the opposing force has been properly scouted and appraised.

George Armstrong Custer

DoS attacks

Aim: Whole networks and/or systems, as well as individual hosts.

Goals: To consume resources in order of shutting down or substantial deteriorating services to the legitimate users.

Resources: Bandwidth, servers/routers computing time, protocol implementations.

Stack overflow, DNS flood, ping flood, packet drop, etc.

DoS attack detection

Anomalies in the traffic pattern: Events or conditions with significant statistical deviation from the usual pattern based on the data previously collected in standard conditions.

Traditional means of defence (firewalls, IDS, etc.) are inefficient.

SIEM: Any deviation over the threshold mean triggers incident alert.

Inefficient for the low-intensity DoS attacks.

Low-intensity DoS attacks

New trend in the cyber warfare: Low-intensity DoS attacks indistinguishable from regular traffic.

Communication channels not overloaded but have significant droppage of the request/acknowledgement packets.

Low-intensity DoS attacks may be adapted against HTTP, SMTP, and/or DNS traffic.

Apache- and *Microsoft* IIS-based systems most vulnerable.

Low-intensity DoS attacks

Require a number of participating or compromised hosts for rogue flooding of the target with useless packets.

Rogue implementation of the DoS methods will fail if a massive amount of anomalous traffic is detected by the firewalls.

Low-intensity DoS attack implement periodic increase (splashes) of the rogue traffic.

Low-intensity DoS attacks

For better efficiency splashes are made close to the time-out of the open session to keep the session alive.

Server/router buffers become gradually overloaded, leading to the denial of service condition.

Low-intensity DoS attacks do not require significantly big bandwidth or computing power.

TCP stack vulnerability

Additive-Increase/Multiplicative-Decrease (AIMD) algorithm combines linear growth of the congestion window with an exponential reduction when a congestion takes place.

When congestion is detected, transmitter decreases transmission rate by a multiplicative factor.

Multiplicative decrease is triggered when a timeout or acknowledgement message indicates a packet was lost.

It is possible to enforce zero-bandwidth through injecting DoS traffic into the regular traffic.

Network bandwidth DoS

DoS consists of short peaks of rogue impulses with carefully synchronized period.

If combined traffic during the peaks is big enough to cause packet droppage, transmission will fail.

Retransmission will be attempted after Retransmission Time-Out (RTO).

If the DoS period coincides with RTO, regular traffic will constantly encounter time-out.

Packet losses will close to 100%, and bandwidth to 0.

Experimental topology

Virtual machines based on *VirtualBox* platform.

Emulated *Intel Core i5-5200* CPU @ 2.20 GHZ.

Operating system: *Ubuntu Linux 14.04*.

HTTP servers: *Apache2* and *nginx*.

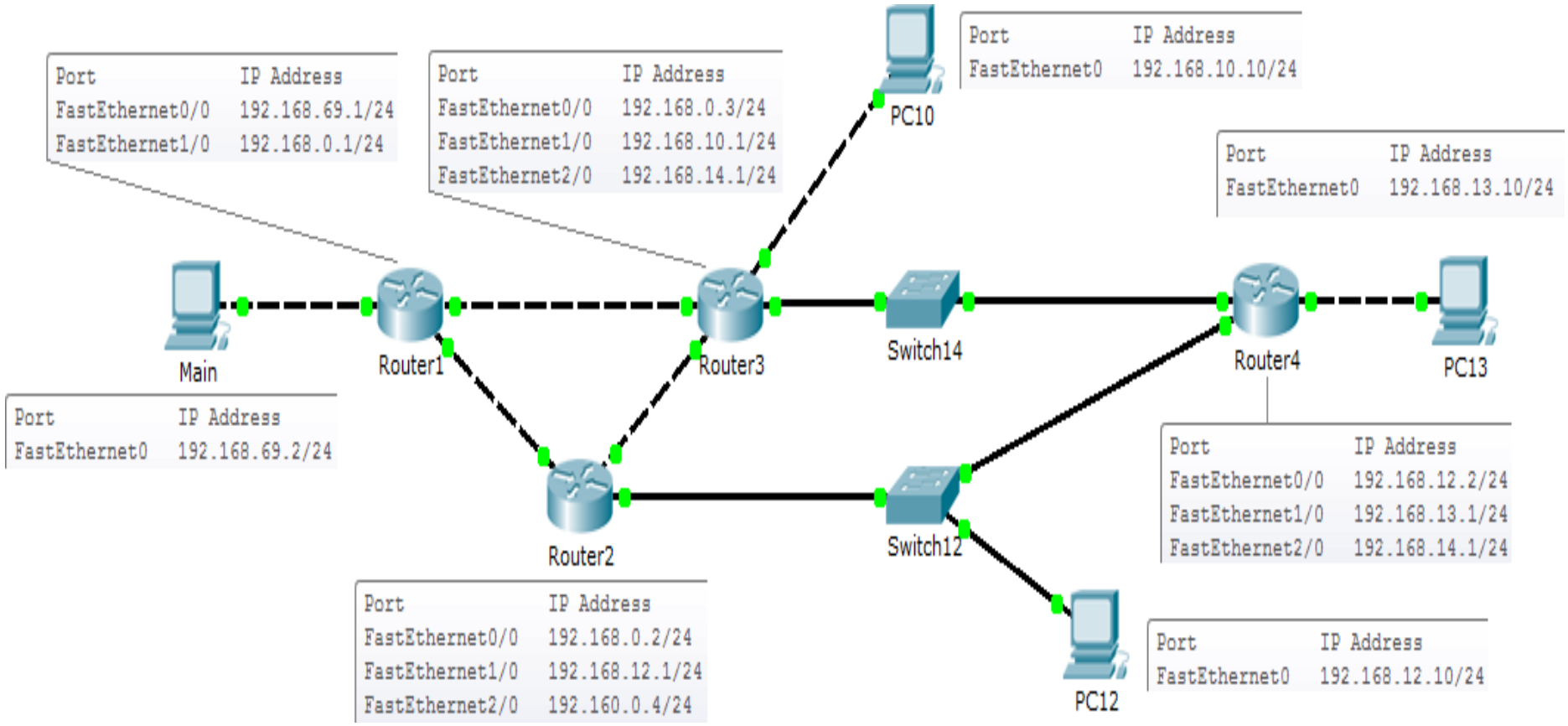
DNS servers: *bind9*.

ICMP and BGP routers: *Zebra* and *Quagga*.

Network topology: *PacketTracer*.

Attacking OS: *Kali Linux*.

Network topology



Branched topology: emulate real-world network

Dynamic routing: assess availability of nodes and services.

Model of DoS attack

At t==0 rogue user sends the first impulse, shuts down the system.

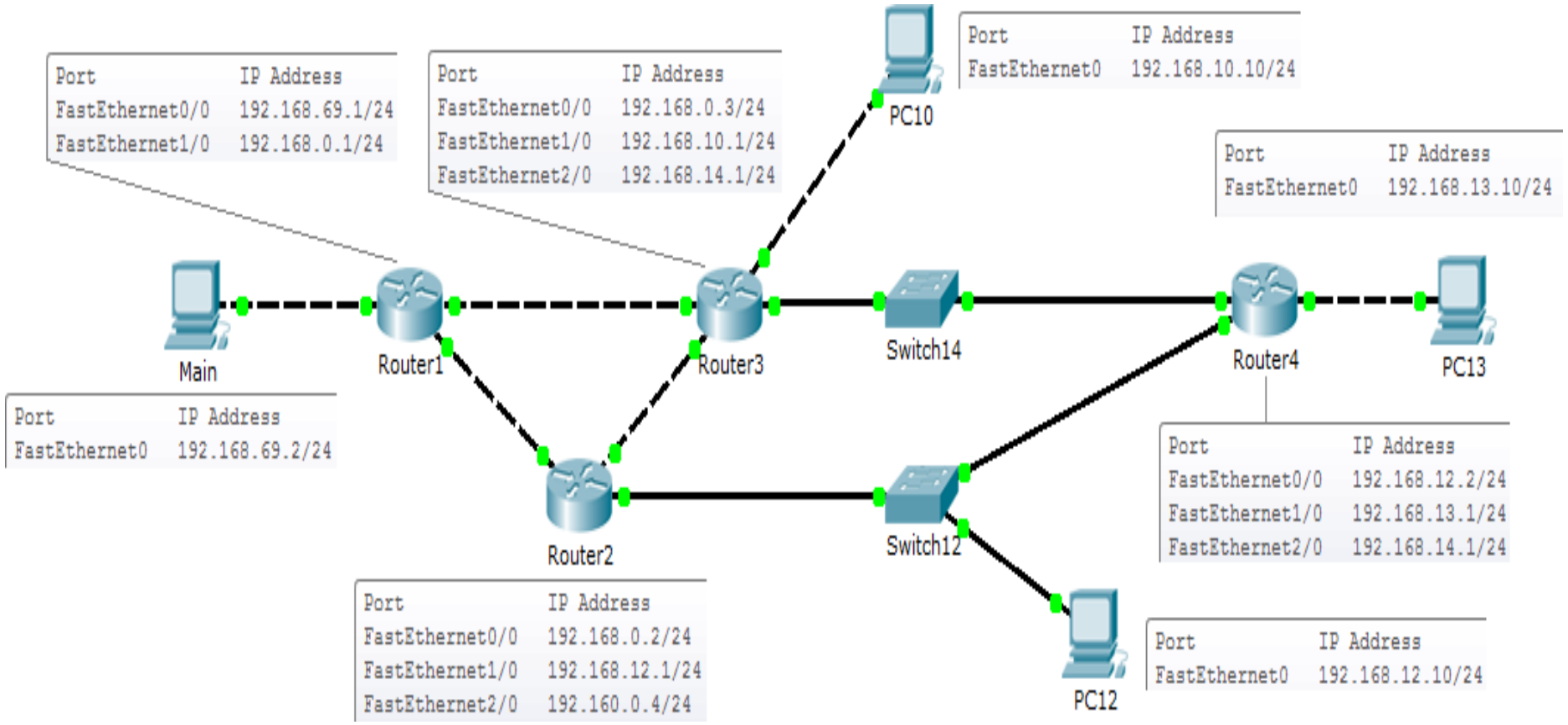
Legitimate user encounters time-out, forced to wait for retransmission, and double the RTO.

Rogue user repeats attack at t==1+2RTT (Round-Trip Time).

legitimate user encounters time-out, forced to wait for retransmission double the time, and double the RTO.

Rogue user will shut down the service by sending packets at low rate – every odd point in time.

HTTP attack



PC10 – target; PC12, PC13 – sources of attack.
Method of attack: SlowLoris. Main – monitor client.

HTTP attack

Attack made with the `slowhttptest` DoS simulator:

```
krewa@Abulafia:~$ slowhttptest -H -u https://192.168.10.10:8080 -p 100 -c 10000 -k 5
```

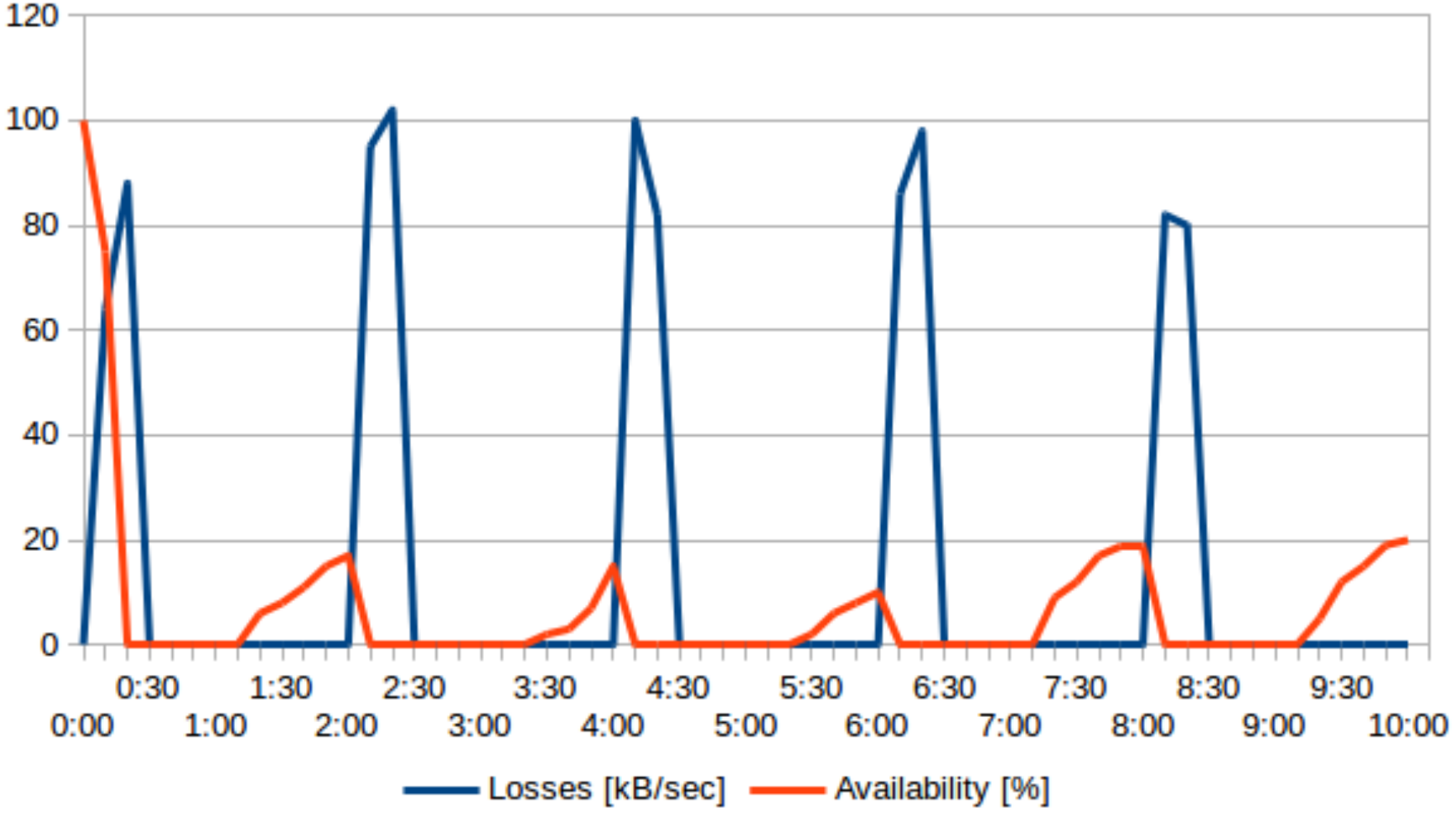
where: -H – SlowLoris mode; -u – attacked URL; -p – time-out;
-c number of connections; -k number of attempts.

Monitoring was made with `siege` stress tester:

```
krewa@Abulafia:~$ siege https://192.168.10.10 -c 10 -t 10M
```

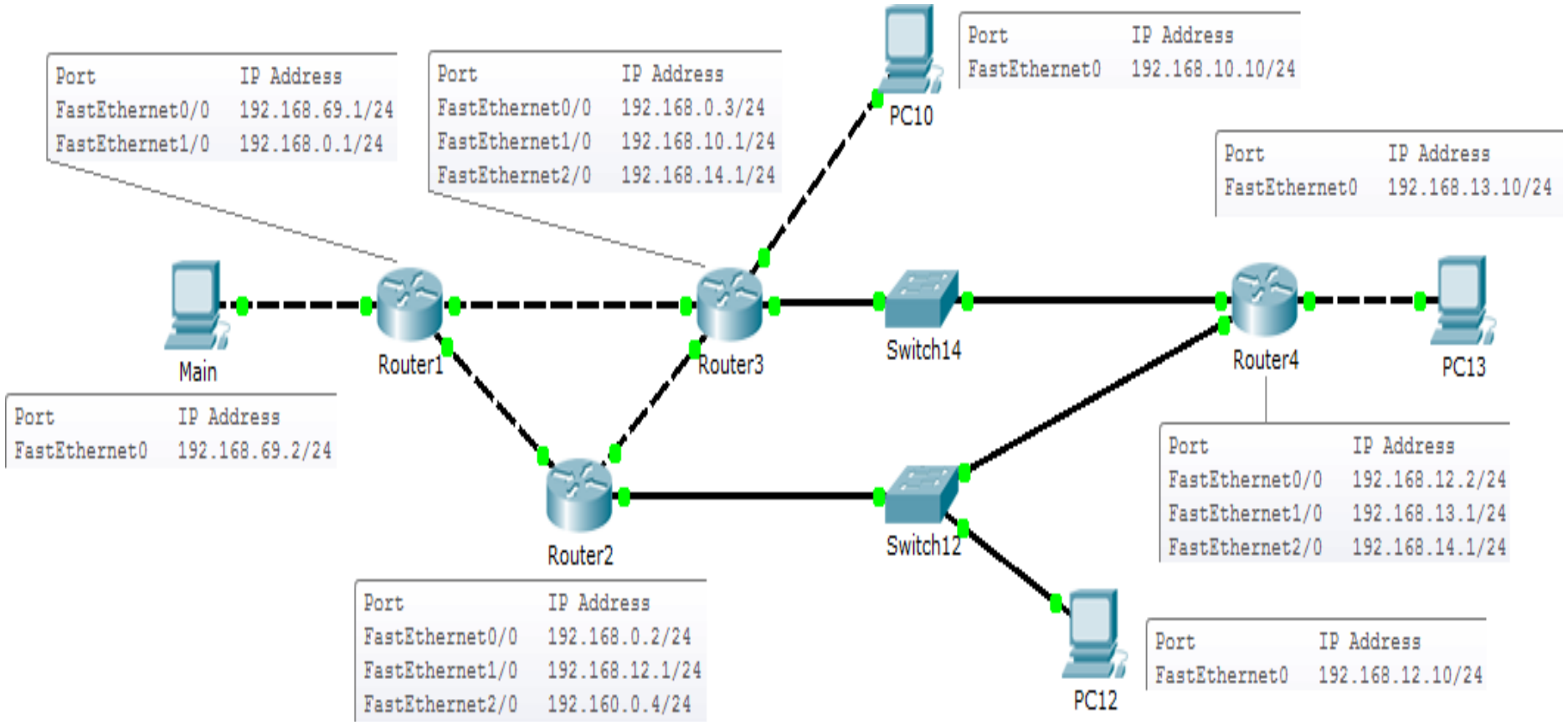
where: -c – concurrent number of simulated users;
-t – selected period of test time.

Losses vs. availability



Successful DoS attack w/o serious investment in the bandwith of attacking hosts.

DoS attack on BGP system



Attack was driven against the network segment on Router3 and Router4.

DoS attack on BGP system

Network throughput measured with `iperf` utility.

Attack:

Scenario 1:Direct attack on *Quagga*.

Scenario 2:Attack on BGP infrastructure behind Router4 to compromise routing channel.

Attack on *Quagga*

SYN-ACK packets sent with 5 sec. time-out.

Using `scapy` Python scripting utility:

```
krewa@Abulafia:~$ scapy
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> import time
>>> i = 1
>>> while i<10000:
...   SYN = sr(IP(dst="192.168.14.1",src="192.168.10.10")/TCP(sport=179,dport=179,flags="S",seq=40+i))
...   SYNACK = sr(SYN)
...   ACK = sr(IP(dst="192.168.14.1",src="192.168.10.10")/TCP(sport=SYNACK.dport,dport=179,flags="A",seq=SYNACK.ack,ack=SYNACK.seq+i))
...   time.sleep( 5 )
...   send(ACK)
...   i += 1
```

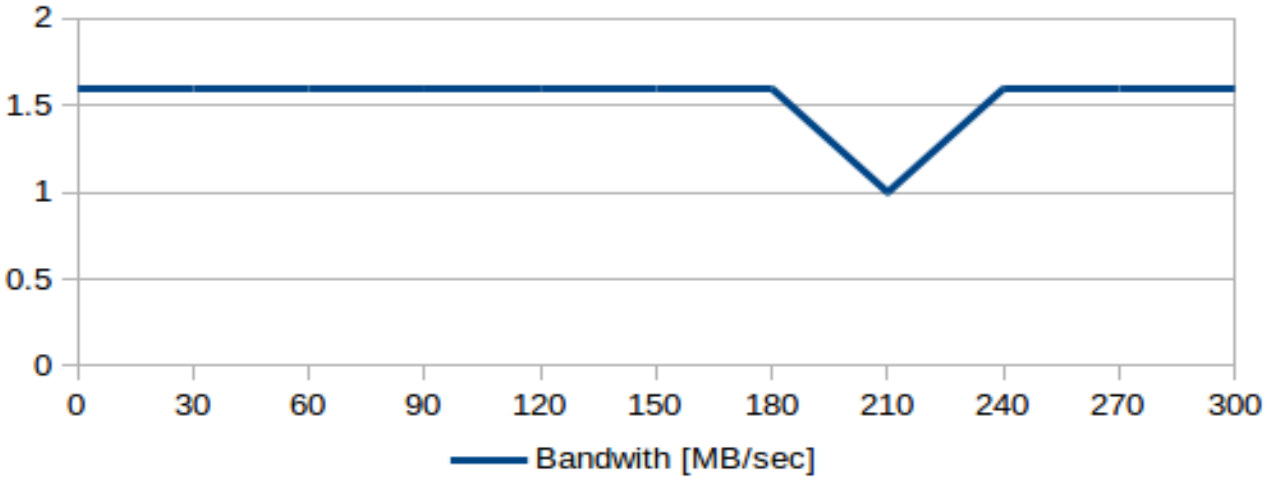
Attack on *Quagga*

Handshake initialized and processed except the `ESTABLISHED` status.

Quagga responds with `RST` packet to the rogue requests.

Changing `time.sleep()` parameter in the 1 to 300 range resulted in closing connection with `SYN-RECV` status.

No problems with availability:



Analysis

Successful low-intensity DoS attack requires BGP emulating software.

Legitimate connection to rogue requests possible only on misconfigured servers.

Data exchange between BGP neighbours based on Access Lists (ACL):

- permission to transmit routes to a neighbour,
- permission to receive routes from a neighbour.

Router-in-the-Middle attack

Attack driven at the server behind attacked router.

Goal: To force the router to lower the bandwidth due to processing rogue traffic generated from low-intensity DoS attack.

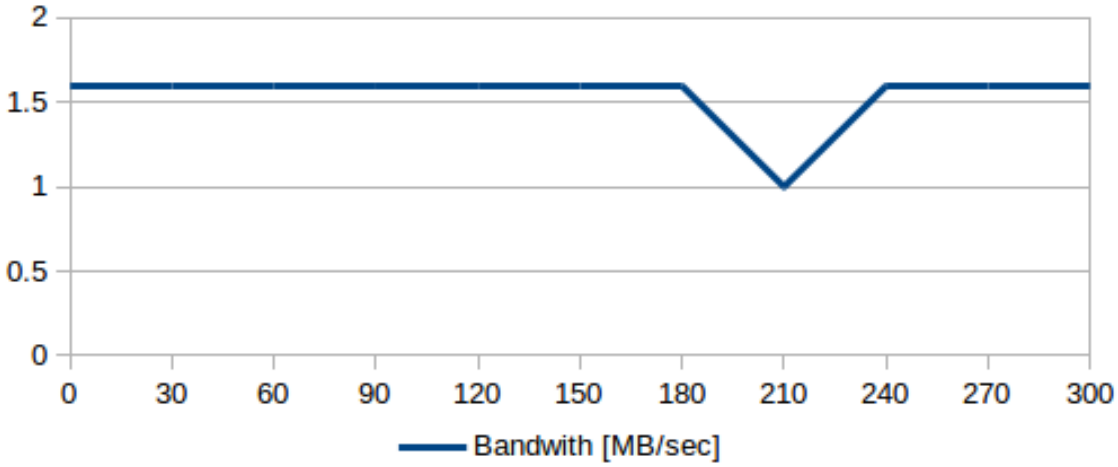
Attacked was PC13 behind Router4:

```
krewa@Abulafia:~$ slowhttptest -H -u https://192.168.10.13:8080 -p 100 -c 10000 -k 5
```

Network throughput measured with `iperf` utility.

Analysis

No changes in the throughput:



Slight droppage of the speed results from interface set-up to match real-world conditions.
Traffic generated from low-intensity DoS attack doesn't affect the border router's bandwidth.

Network throughput measured with `iperf` utility.

Analysis

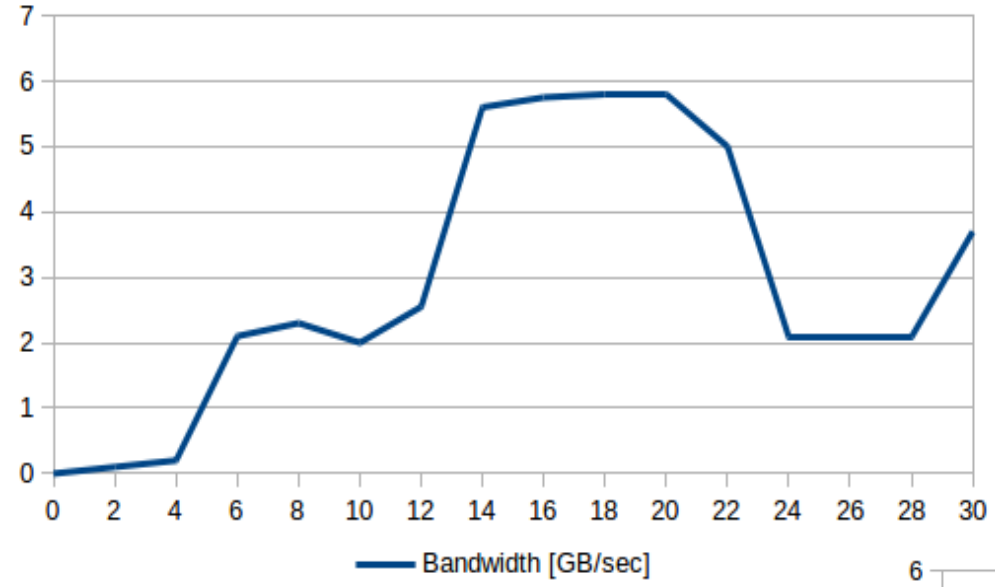
Attacks on systems with default configuration were successful.

As a rule default configurations ignore parameters to counter-act attacks. *Quagga* is a remarkable exception.

Low-intensity DoS attacks deteriorate channel bandwidth.

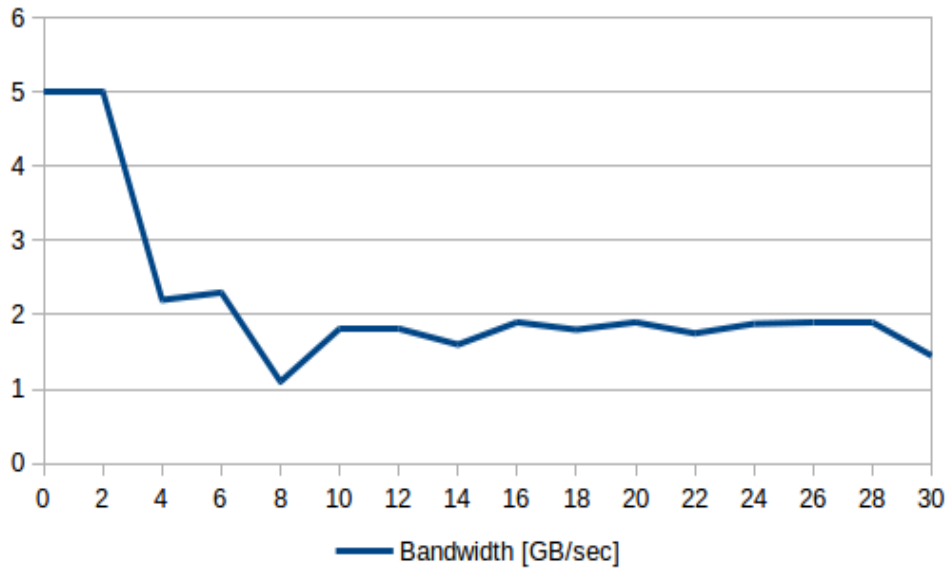
It results in denial of HTTP services to legitimate users.

Comparison



Normal traffic.

Traffic under attack.



Conclusions

Aleksandar Kuzmanovic, Edward W. Knightly. *Low-rate TCP-targeted denial of service attacks and counter strategies*. IEEE/ACM Trans. Netw. – 2006. – No 14 (4). – C. 683-696.

discusses how low-intensity DoS attacks on routing protocols may cause avalanche effect and destroy substantial segments of the Internet.

Experiment proves that such an attack may succeed only in the presence of many factors, including routers misconfiguration, substantial amount of computing resources, and well-coordinated scenario of the attack.

Questions?

Thank you for your attention!