DDoS: Undeniably a global Internet problem looking for a global solution

RIPE-41 EOF Tutorial, January 15, 2002, Amsterdam

Yehuda Afek and Hank Nussbacher
Wanwall Ltd.

wanwall

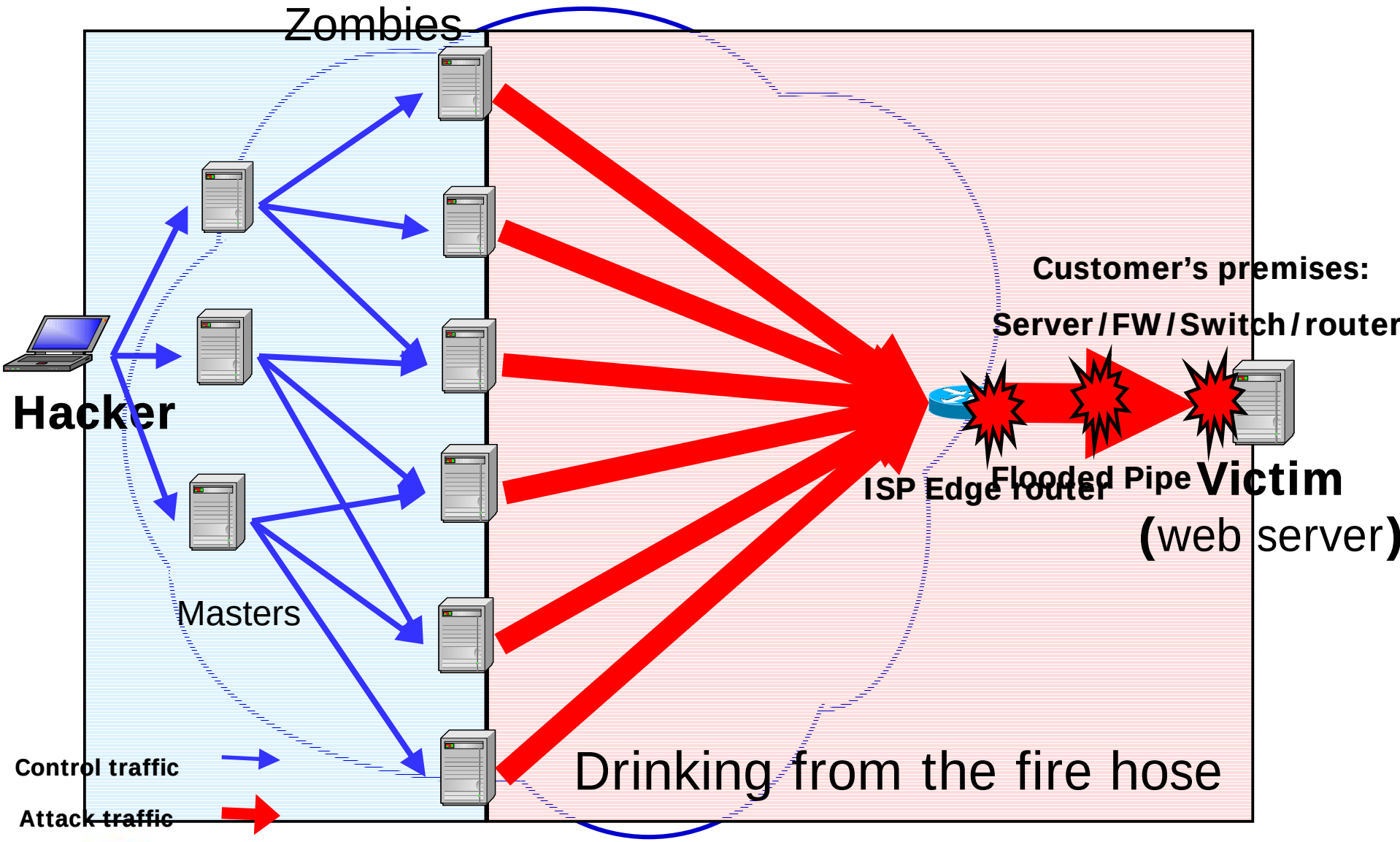
Goede morgen dames en heren, het is fijn om weer in A'dam te zijn

- Yehuda Afek
  - CTO & founder Wanwall Inc. and Professor of Computer Science at Tel-Aviv University
- Hank Nussbacher
  - Internet and security consultant with over 20 years of experience
- Wanwall Inc.
  - A company providing systems to defeat and stop DDoS attacks.

Outline

1. Outline
2. DDoS: What & How
3. Events, History and Today
4. DDoS attacks, physics & mechanisms
5. Measurements
6. Standards & Academia
8. Detection
9. Protection and Defense
10. DDoS companies
11. Links

DDoS


Zombies
Hacker
Masters
Control traffic
Attack traffic
Customer's premises:
Server / FW / Switch / router
ISP Edge router
Flooded Pipe
Victim
(web server)
Drinking from the fire hose

Who cares?

- **2/2000:** $1.2 Billion cost to US market
  - **$100 Million revenue loss**
- **1/2001:** $10's Million damage due to Microsoft attack
- **5/2001:** Whitehouse site down six hours
- **6/2001:** CERT down twice for > seven hours
- **6/2001:** Weather.com
- **7/2001:** Lufthansa.com
- **8/2001:** White House ('Code Red')
- **9/2001:** Deutsche Bank
- **10/2001:** NY Times
- **11/2001:** Attacks targeting routers (IDG News)

4,000 attacks per week *CAIDA*

Who cares? (2)

- Everybody is vulnerable
  - ISPs
  - Hosting centers
  - ASP's
  - Government
  - Banks, Financial institutions
  - E-commerce
  - DNS servers
  - Email accounts
- Easy to mount
- Download, click and launch

Background

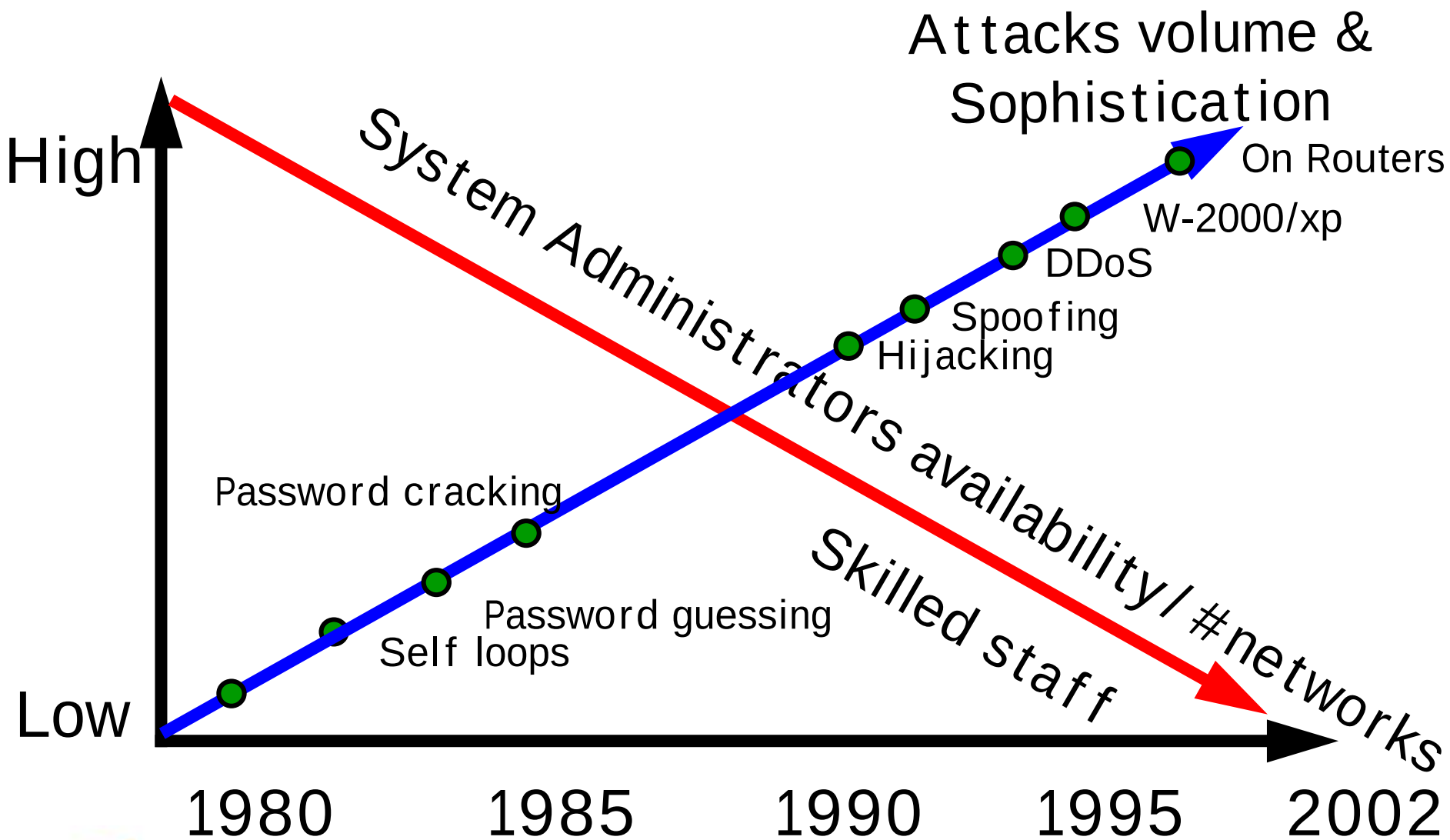
- Motives
  - Showoff
  - Terror
  - Cyberspace demonstrations
  - Ransom
  - Blackmailing
  - Get your aggression out in cyber space
  - Boredom
- Same as in real life


The Joy of Tech
by Nitrozac & Snaggy
I'm so proud of Jimmy. Apparently he's one of the world's best at a new computer game.
Oh really! Which game?
Something called "DDOS Attack"
HACKER MOM
© 2000 Geek Culture™
joyoftech.com

DDoS is NOT

- Information theft (passwords, credit cards)
  - Financial fraud
- System penetration
  - Obtain root permission
- System crashing by:
  - Buffer overflow
  - Bashing the stack
- Breaking crypto

Problem on the rise: Hackers



3. Events, History and Today

Events - prehistory

- Shoch & Hupp, "The `Worm' Programs--Early Experience with a Distributed Computation," Communications of the ACM, March 1982
  - Meant to be a memory diagnostic program
  - 100 Alto computers brought to a standstill on an Ethernet
  - Used forced multicast since multicast didn’t exist then

Evolvement of attacks

- Sep 1996: Panix under SYN attack
- Jan 1997: Romanian hacker SYN floods Undernet (IRC net)
  - "We have some of the greatest minds in Internet technology here, and they couldn't do anything [to stop the attack]" -Wired, Jan 14, 1997
- Jan 1998: Tribe flooding tool appears for mIRC
- Jan 1998: Smurf attacks cripple ISPs
- March 1998: Smurf attack on University of Minnesota
- Aug 1999: Trinoo and TFN appear

Major attack not long in coming!

Evolvement of attacks (2)

- 02-2000: Infamous DDoS attacks (Yahoo, eBay, CNN), TFN2K, Stacheldracht
- 03-2000: Shaft
- 04-2000: DNS amplification attacks, mstream
- 05-2000: VBS/Loveletter
- 07-2000: Hybris
- 08-2000: Trinity IRC-based DDoS tool (unix)
- 11-2000: Multiple IRC-based DDoS tools (Windows), NAPHTA

NANOG23: http://www.nanog.org/mtg-0110/ppt/houle

Mafiaboy timeline, Feb 7,8,9 2000

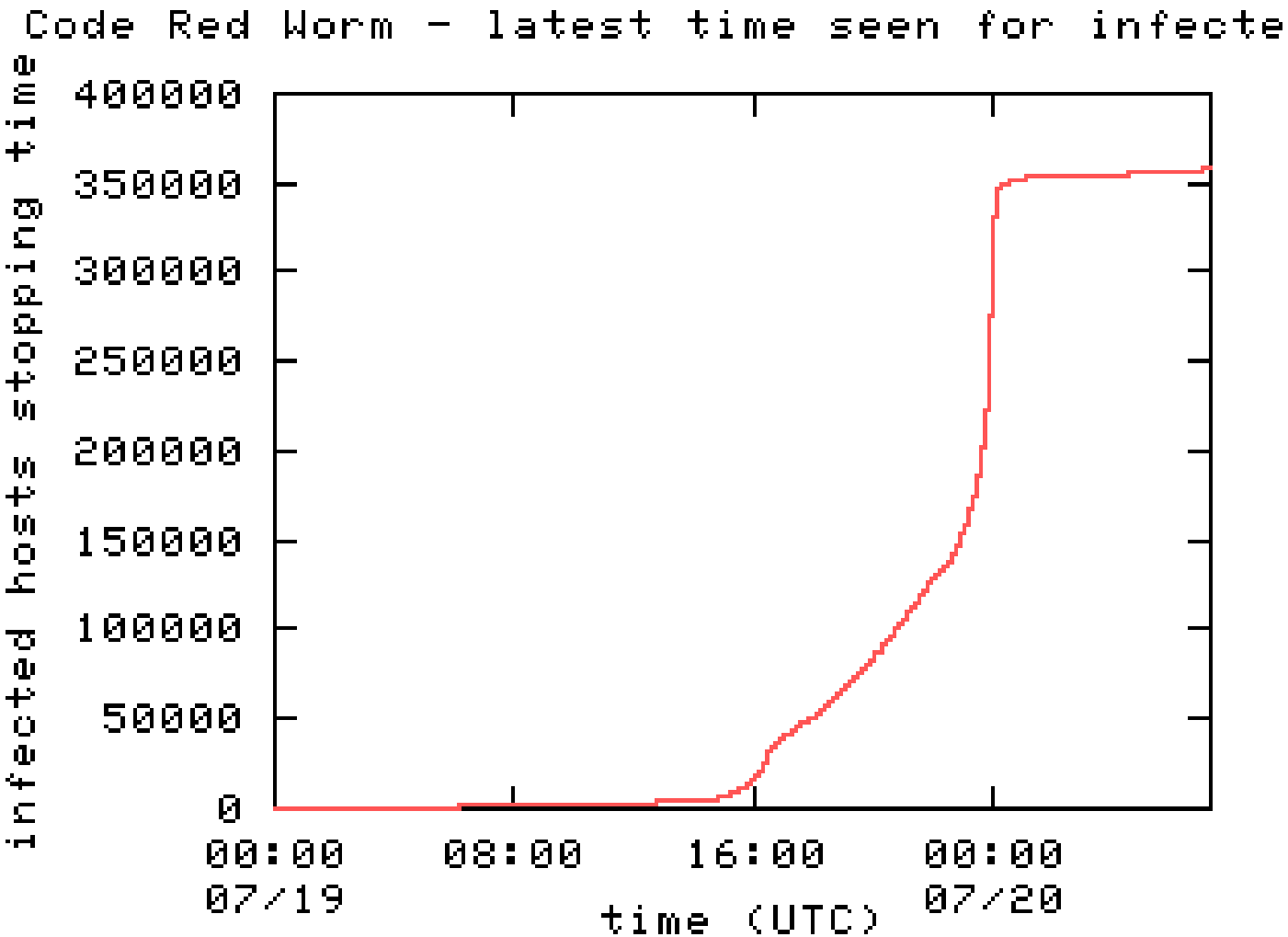
- Feb 7
  - Yahoo Mon 10:20 a.m. 3 hours
- Feb 8
  - Buy.com Tues 10:50 a.m. 3 hours
  - eBay Tues 3:20 p.m. 90 minutes
  - CNN.com Tues 4:00 p.m. 110 minutes
  - Amazon.com Tues 5:00 p.m. 1 hour
- Feb 9
  - E*Trade Wed 5:00 a.m. 90 minutes
  - Datek Wed 6:35 a.m. 30 minutes
  - ZDNet Wed 6:45 a.m. 3 hours

Tools evolvement: 2001

- 01-2001: Ramen worm
- 02-2001: VBS/OnTheFly (Anna Kournikova), 1i0n worm
- 03-2001: Stick
- 04-2001: Adore/Red worm, carko DDoS tool
- 05-2001: cheese worm, w0rmkit worm, sadmind/IIS worm
- 06-2001: Maniac worm, Code Red worm
- 07-2001: W32/Sircam, Leaves, Code Red II, various telnetd worms, various IRC-based DDoS tools (knight, kaiten)
- 09-2001: Nimda worm, Code Blue
- 12-2001: Goner worm

NANOG23: http://www.nanog.org/mtg-0110/ppt/houle/

Code Red spread



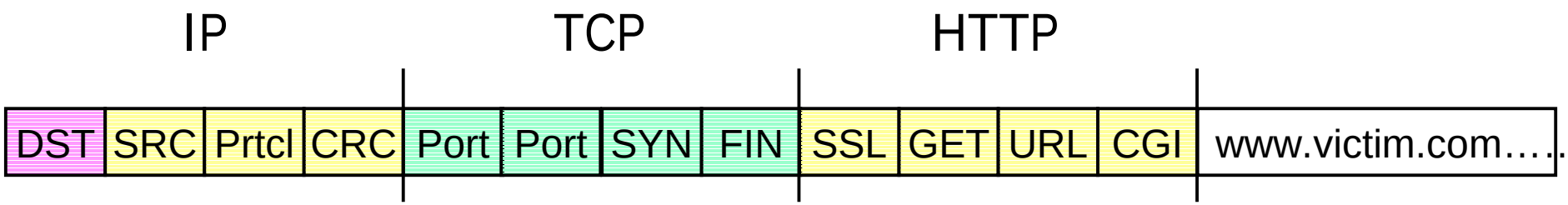
Over 350,000 IIS servers infected is less than 14 hours!

CAIDA stats

4.1 DDoS attacks - Ammunition

Ammunition: packet crafting

- Any field in any header *
- Any combination of fields
- Randomization



* except DST

Standard ammunition

TCP	SYN ACK FIN RST	SRC Spoofing
UDP	Diff sizes	Amplification
ICMP	Redirect Unreachable	Impossible flags
DNS	Requests Replies	Illegal headers

- Simple
- Effective
- Why to change?

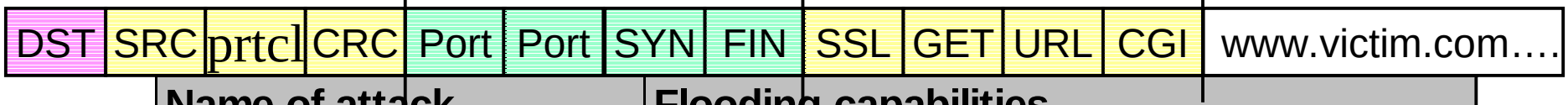
Additional types of ammunition

HTTP requests	Legal
Heavy application rqsts	
Many connections	Illegal
Incomplete connections	

Summary

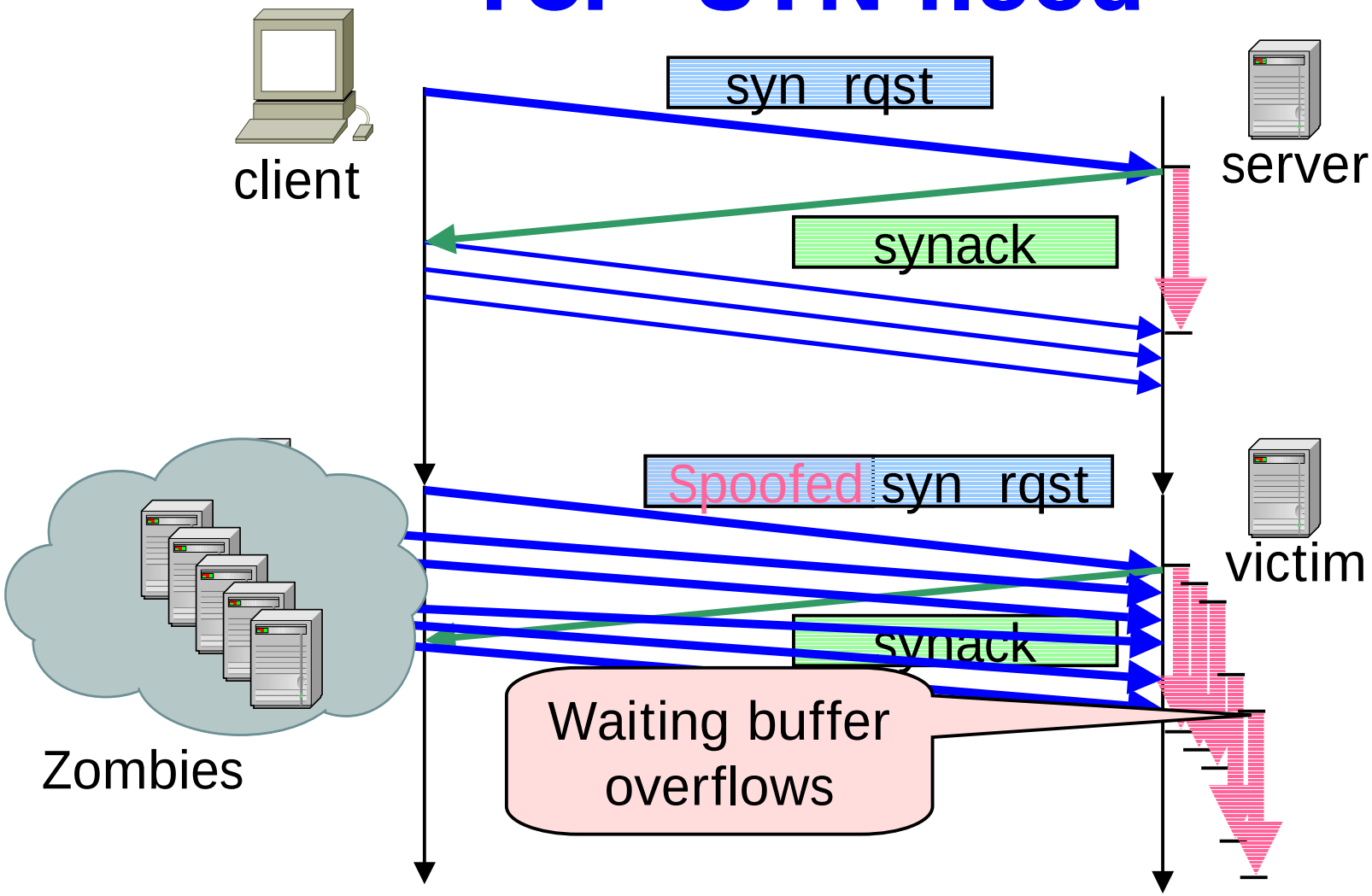
SYN	TCP
Smurf	ICMP
DNS Reply Queries flood	UDP
IGMP flood	IGMP
Fraggle (UDP loop)	UDP
TCP flood	TCP NUL, TCP RST, TCP ACK
UDP reflectors	UDP
TCP reflectors SYNACK	TCP
Client (URL) attacks Refresh and Error	HTTP

Generic attacks



Name of attack	Flooding capabilities
Land	TCP SYN (SRC=DST)
SYN	TCP SYN (spoofed SRC)
Smurf	ICMP via Amplifiers
ICMP redirect	ICMP
IGMP flood	IGMP
Fraggle (UDP loop)	UDP smurfing
TCP flood	TCP NUL, TCP RST, TCP ACK
UDP reflectors	UDP (ICMPs, unreachable, redirect)
URL client attacks	HTTP over TCP
VPN attacks	TCP, GRE or IPIP
Teardrop	TCP fragments (overlapping)
Ping of death	ICMP (> 65536 B)
Open/close	TCP, UDP (inetd)
ICMP Unreachable	spoofed ICMP unreachable
IRDP	ICMP router discovery, mass routing tables
ARP redirect	ARP

TCP SYN flood



- One of the first CERT DDoS advisories issued – 9/1996
- http://www.cert.org/advisories/CA-1996-21.html

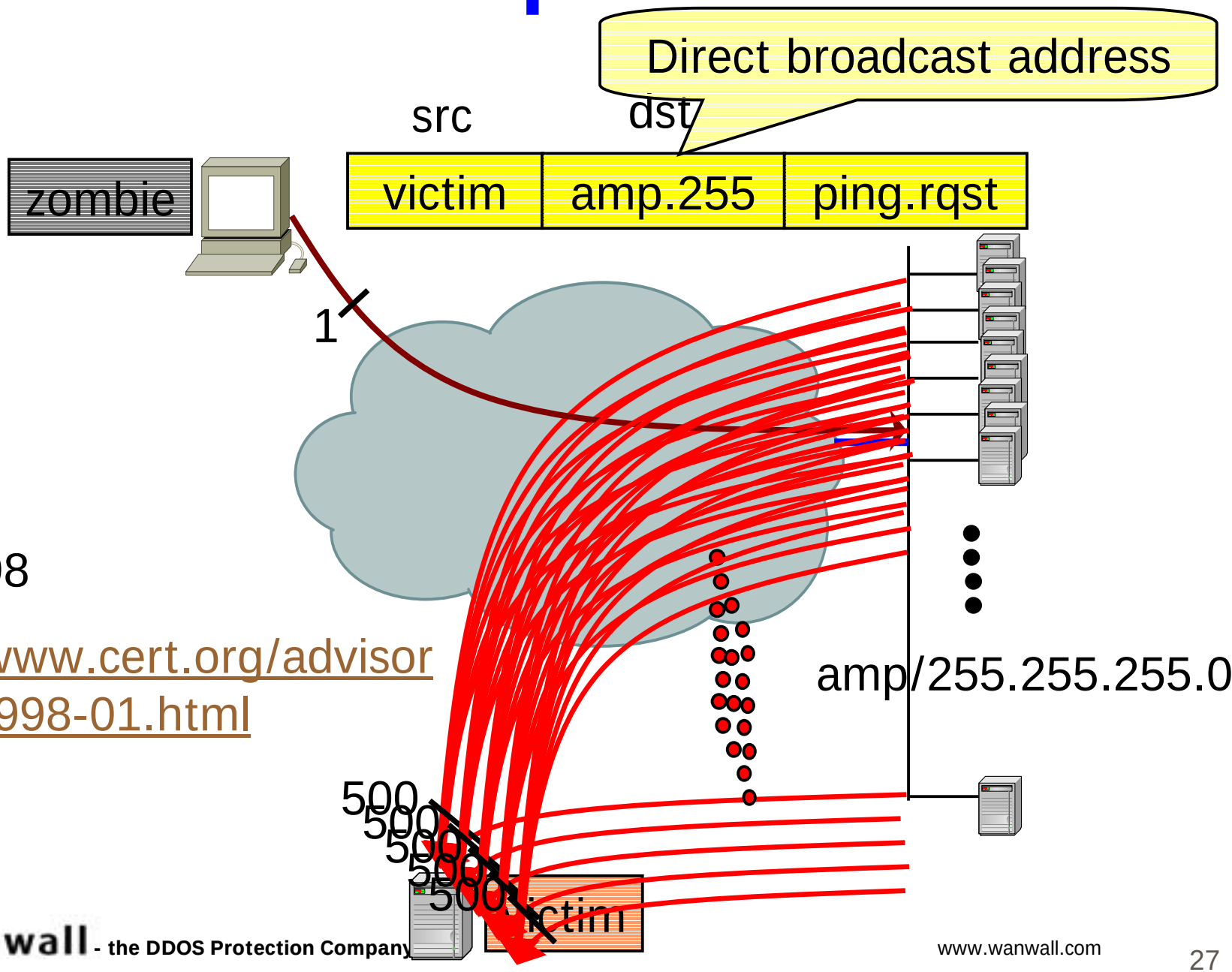
Teardrop/Land attack

- Dec 1997
- Land: source and destination IP are the same causing response to loop
- Teardrops: send overlapping IP fragments
- http://www.cert.org/advisories/CA-1997-28.html

NAPHTA: TCP connections

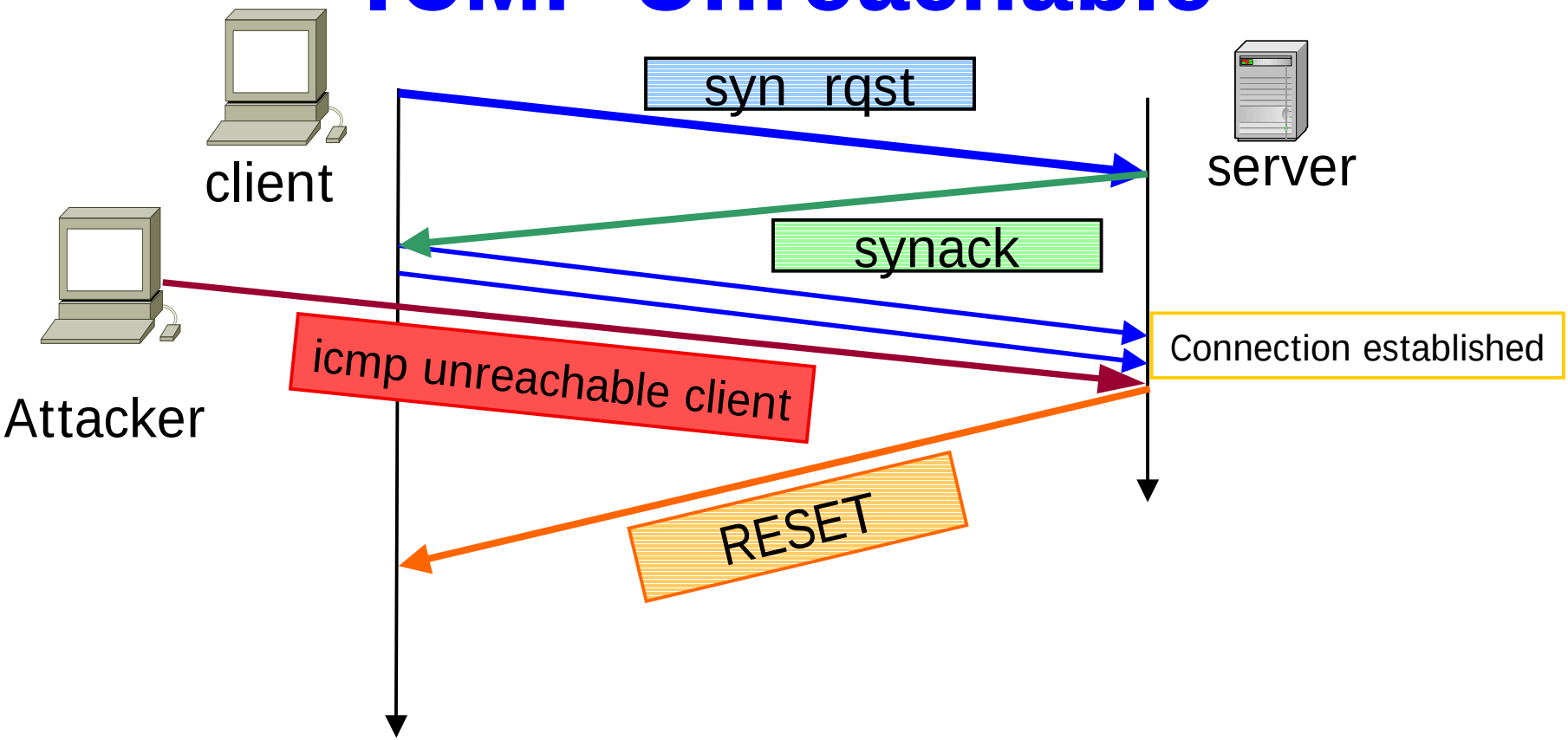
- Repeatedly establishing a connection and then abandoning it, an attacker can tie up resources. Fill up the TCP connections buffer.
- http://people.internet2.edu/~shalunov/netkill

Smurf Amplification



- Jan 1998
- http://www.cert.org/advisories/CA-1998-01.html

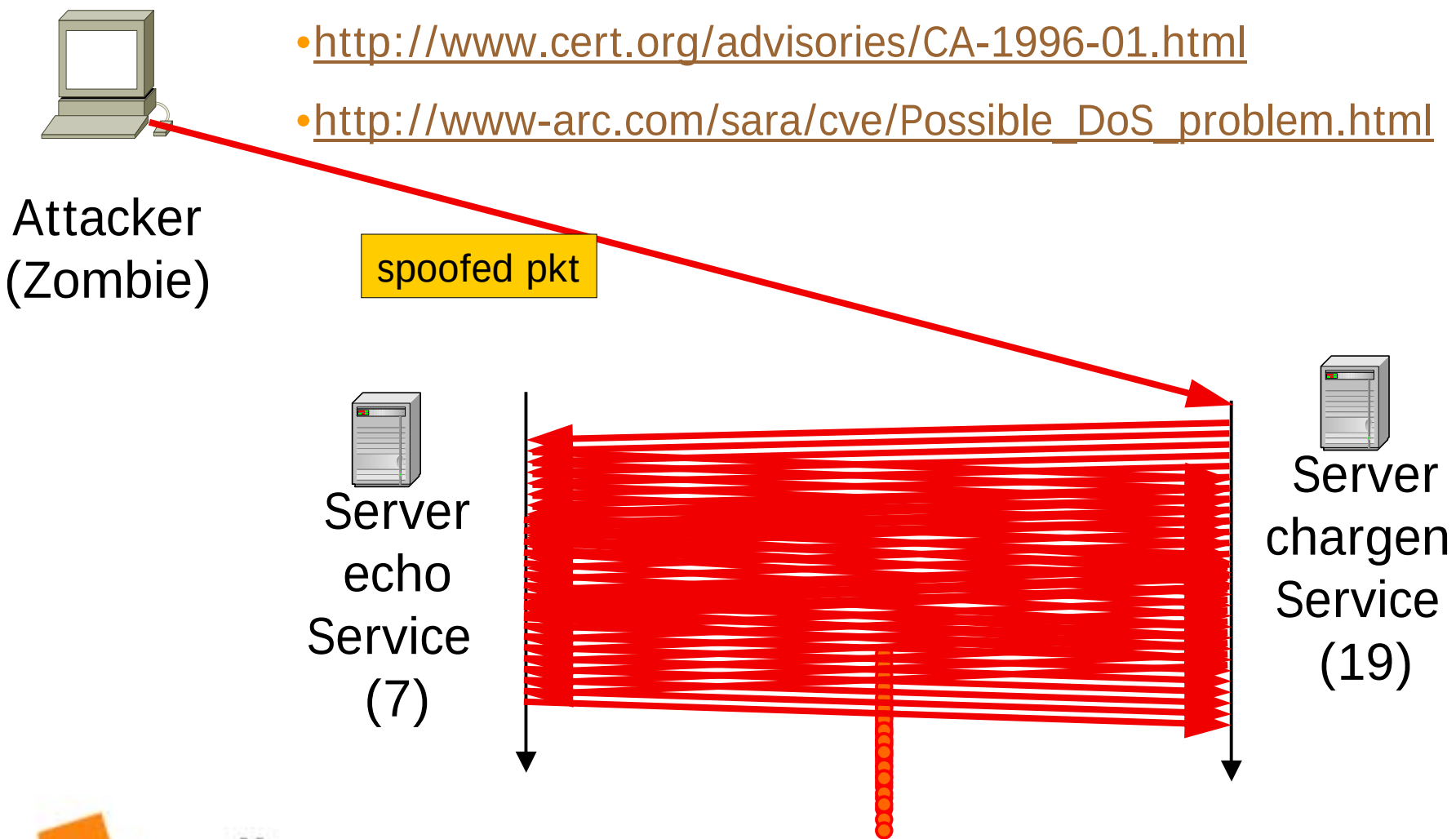
ICMP Unreachable



- Causes all legitimate TCP connections to the spoofed IP addresses, to be torn down
- http://www.networkice.com/Advice/Intrusions/2000104/default.htm

Looping UDP

- First known CERT DDOS advisory – Feb 1996
- http://www.cert.org/advisories/CA-1996-01.html
- http://www-arc.com/sara/cve/Possible_DoS_problem.html



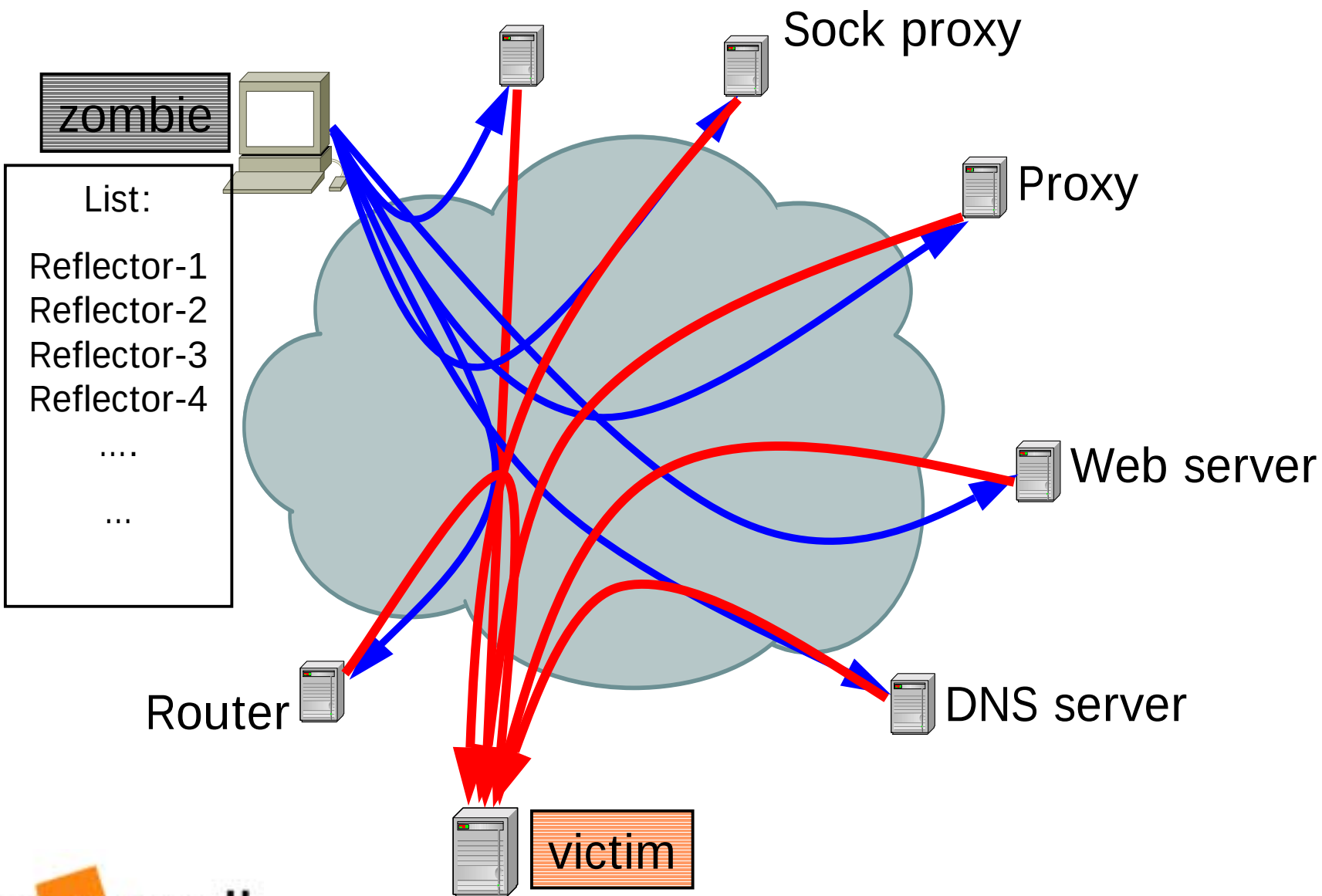
DNS attack

- DNS request
  - Spoofing
  - Random requests
  - Reflectors
- DNS replies
  - Spoofing
  - Junk

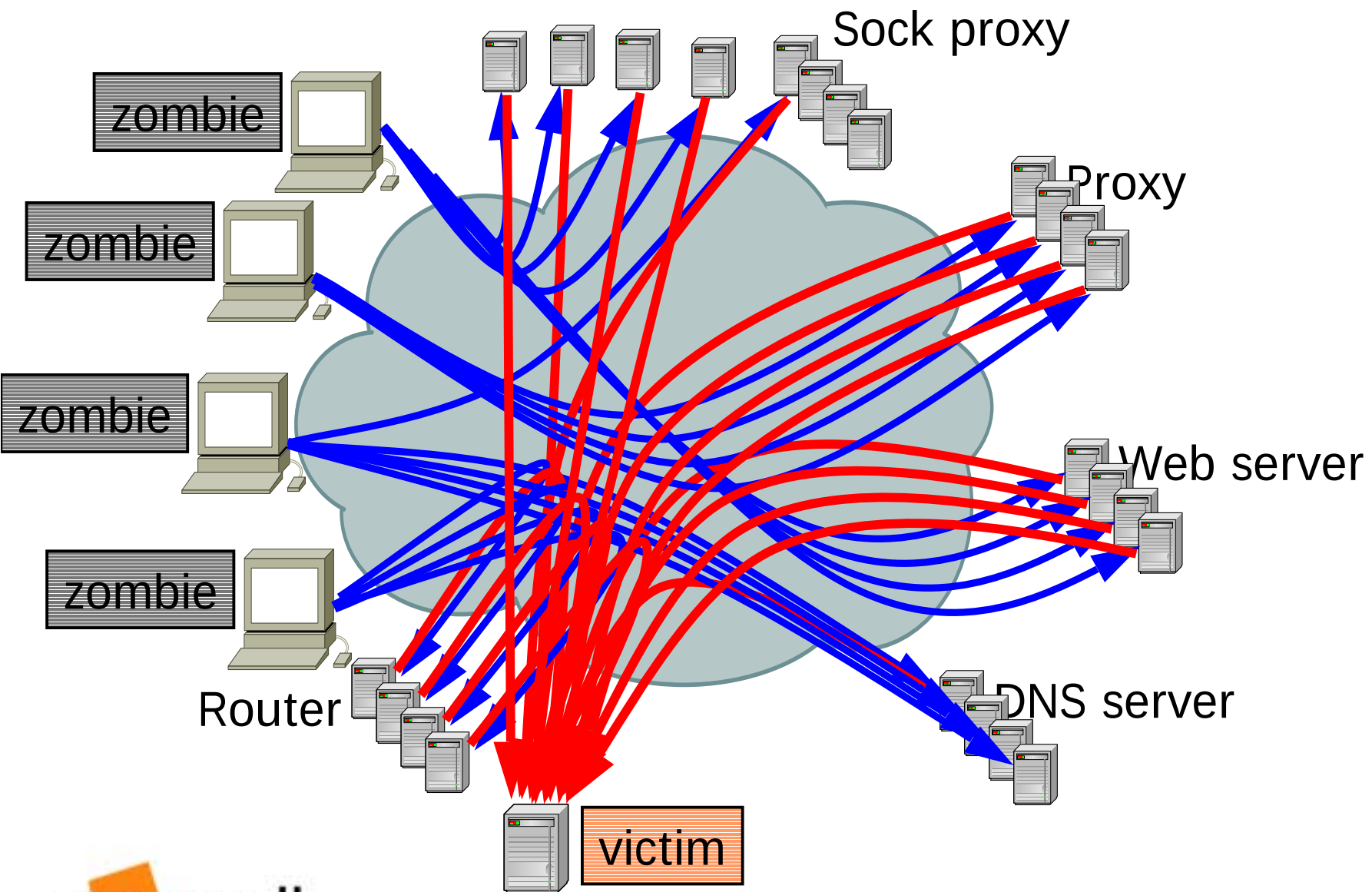
Reflectors -> Bandwidth attack

- Reflectors= returns a packet if one is sent
  - Web servers, DNS servers and routers
    - Returns SYNACK or RST in response to a SYN or other TCP packets with ACK
    - or query reply in response to a query
    - or ICMP Time Exceeded or Host Unreachable in response to particular IP packets
  - Attackers spoof IP addresses from a zombie
  - Vern Paxson research
  - http://www.aciri.org/vern/papers/reflectors.CCR.01.pdf

Reflectors

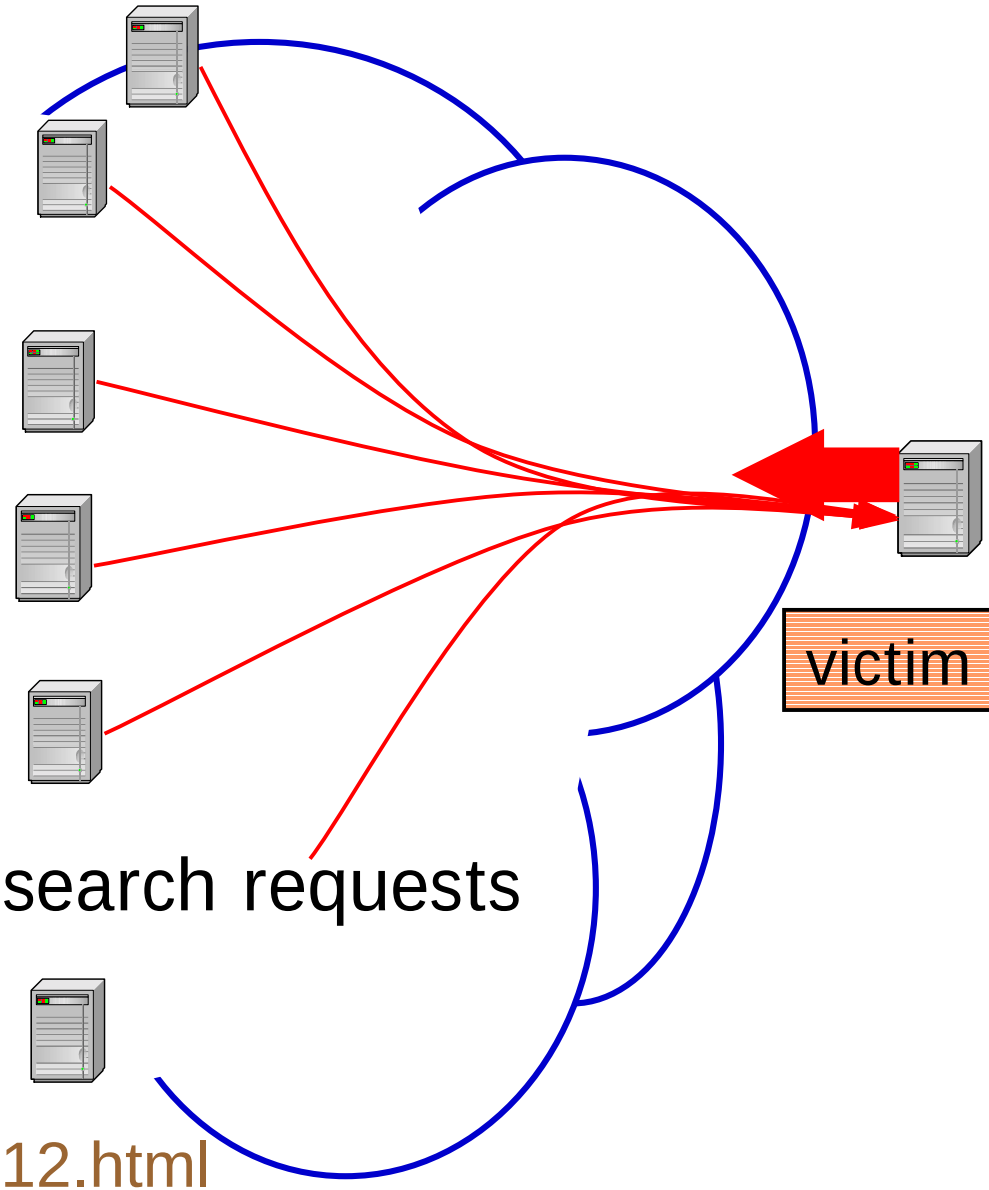

Sock proxy
zombie
List:
Reflector-1
Reflector-2
Reflector-3
Reflector-4
....
...
Proxy
Web server
Router
DNS server
victim

Reflectors

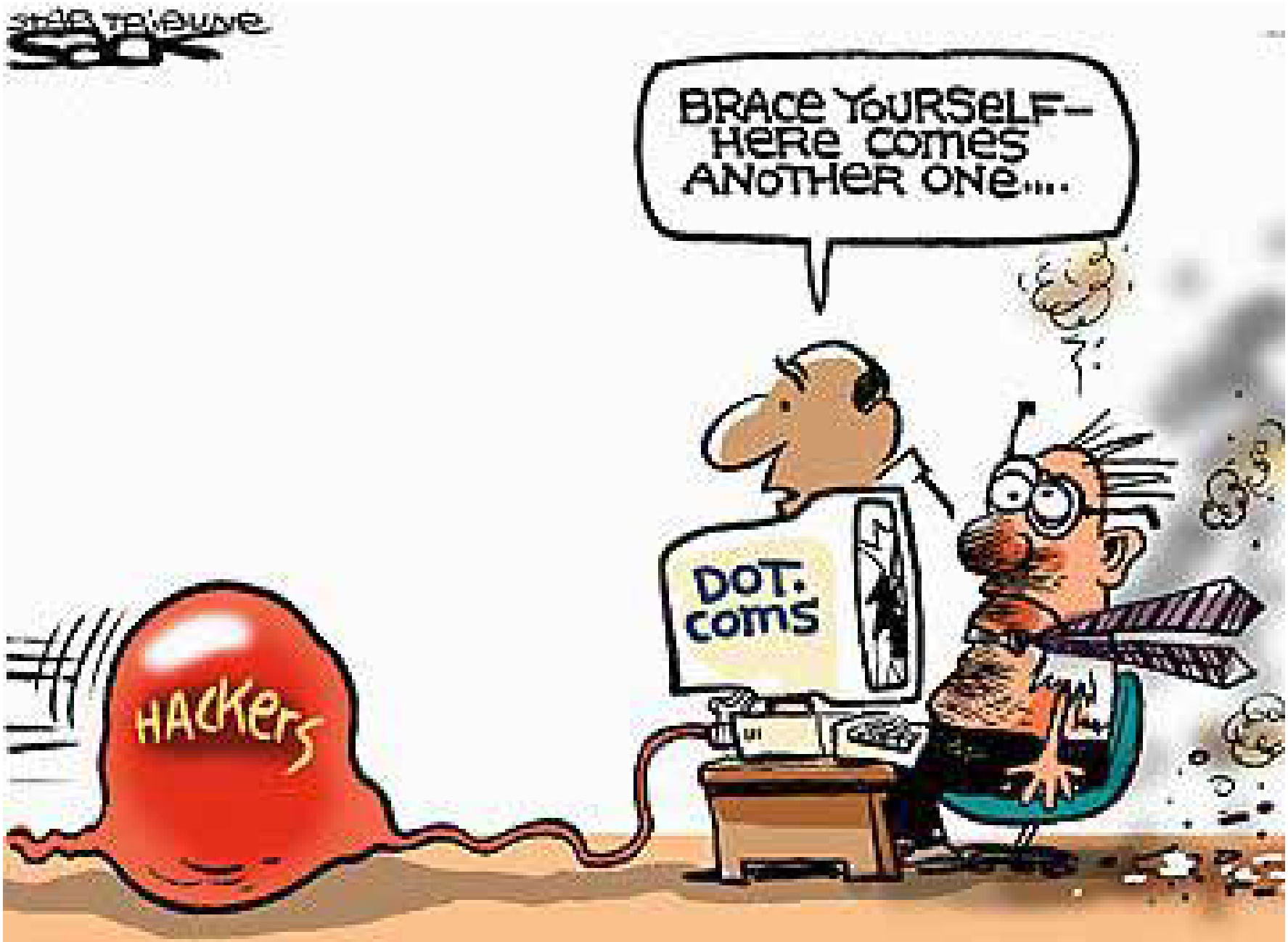

Sock proxy
zombie
zombie
zombie
zombie
Proxy
Web server
DNS server
Router
victim

Client attack

- URL attacks
  - Repeated request
  - Repeated REFRESH
  - Random URL
    - Avoids proxy
    - Works hard
    - Large log file
  - cgi, long forms, heavy search requests



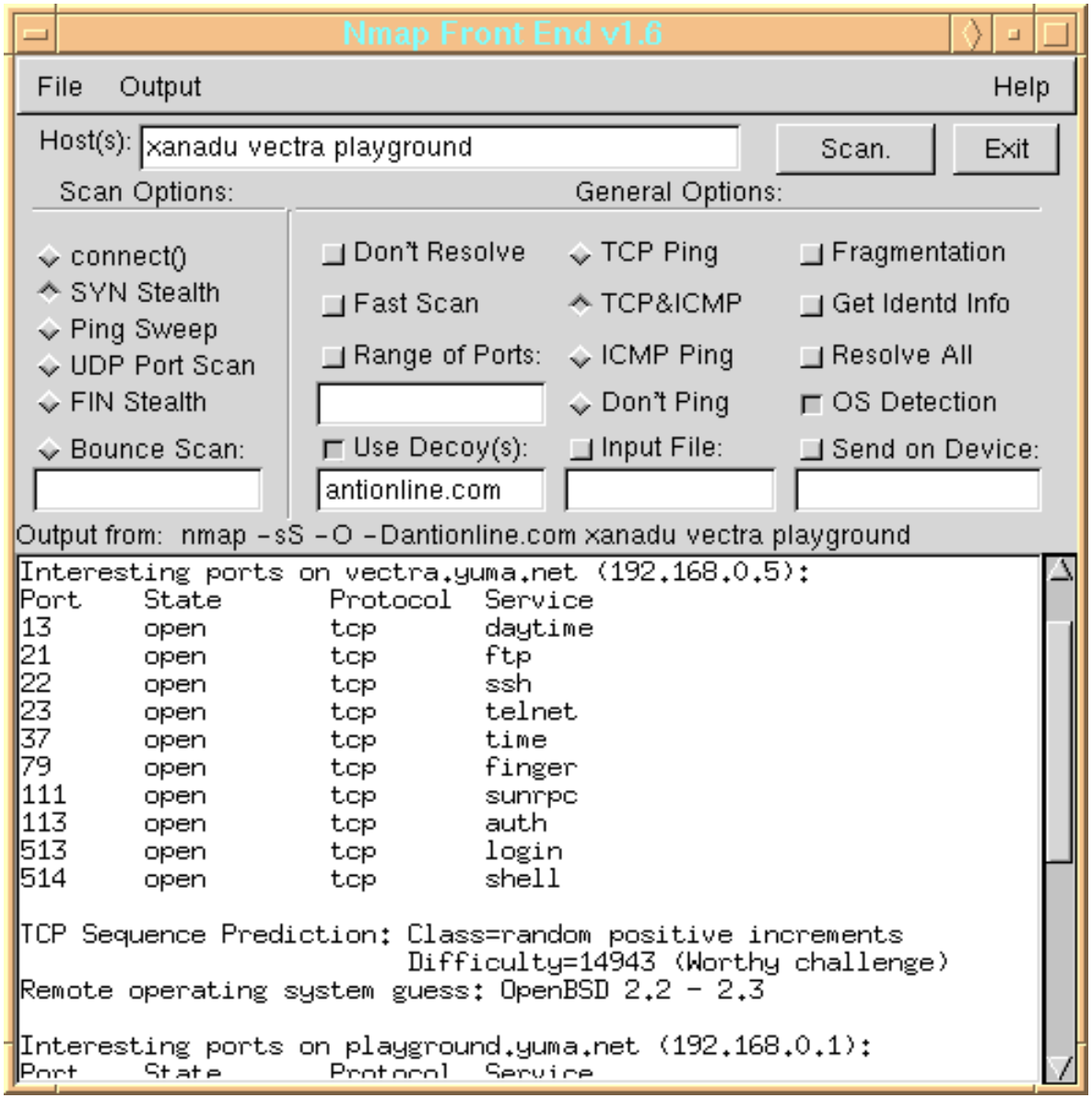
- http://all.net/journal/netsec/9512.html


BRACE YOURSELF— HERE COMES ANOTHER ONE....
DOT. COMS
HACKERS

4.2 DDoS attacks - tools

Probing stage

- Most DDOS attack tools are compromised computers
- Attackers would scan systems for non-secured services
- Many automated scanning tools around



Attack tools 1: FAPI

- Spoof IP addresses
- UDP packets to random or specified ports
- Automatic termination at specified time
- One of the first tools available in May 1998

Attack tools 2: Trinoo

- UDP attacks to random ports
- Defaults:
  - 120 seconds (max 1999 seconds)
  - Packet size: 1000 octets
- Master Slave communication clear TCP and UDP
- Does not support IP spoofing
- Link: http://xforce.iss.net/alerts/advise40.php

Attack tools 3: TFN

- Spoof IP addresses
- Master Zombie communicate by ICMP echo reply
- Flooding: ICMP echo, TCP SYN, UDP flood (trinoo emulation), Smurf
- Link: http://xforce.iss.net/alerts/advise43.php

TFN code

```
/* td.c - tribe flood network synflooder (c) 1999 by Mixter - PRIVATE */
char synb[8192];
void
syn (u_long victim, u_short port)
{
  struct sockaddr_in sin;
  struct iphdr *ih = (struct iphdr *) synb;
  struct tcphdr *th = (struct tcphdr *) (synb + sizeof (struct iphdr));
  srandom ((time (NULL) + random ()));
  ih->version = 4;
  ih->ihl = 5;
  ih->tos = 0x00;
  ih->tot_len = sizeof (ih) + sizeof (th);
  ih->id = htons (random ());
  ih->frag_off = 0;
  ih->ttl = 255;
  ih->protocol = 6;
```

TFN GUI

```
sun17>usage: tfn <options>
[-P protocol]    Protocol for server communication. Can be ICMP,
                      UDP or TCP.  Uses a random protocol as default
[-D n]           Send out n bogus requests for each real one to decoy
                      targets
[-i target string]        Contains options/targets separated by '@', see below
[-S host/ip]     Specify your source IP. Randomly spoofed by default,
                      use your real IP if you are behind spoof-filtering routers
[-f hostlist]    Filename with list of hosts with TFN servers to contact
[-p port]           A TCP destination port can be specified for SYN floods
<-c command ID>  0 - Halt all current floods on server(s) immediately
                 1 - Change IP antispoof-level (evade rfc2267 filtering)
                    usage: -i 0 (fully spoofed) to -i 3 (/24 host bytes spoofed)
                 2 - Change Packet size, usage: -i <packet size in bytes>
                 3 - Bind root shell to a port, usage: -i <remote port>
                 4 - UDP flood, usage: -i victim@victim2@victim3@...
                 5 - TCP/SYN flood, usage: -i victim@... [-p destination port]
                 6 - ICMP/PING flood, usage: -i victim@...
                 7 - ICMP/SMURF flood, usage: -i victim@broadcast@broadcast2@...
                 8 - MIX flood (UDP/TCP/ICMP interchanged), usage: -i victim@...
                 9 - TARGA3 flood (IP stack penetration), usage: -i victim@...
                 10 - Blindly execute remote shell command, usage -i command
```

TFN GUI (2)

```
sun18>tfn -r slaves -i victim-ip -c8

       Protocol       : random
       Source IP     : random
       Client input   : list
       Target(s)      : 192.168.252.5@192.168.252.5
       Command     : commence syn flood, port: random

Sending out packets: ..

       Command        : bind shell(s) to port 192
       Command        : commence udp flood
       Command        : commence icmp echo flood
       Command        : commence icmp broadcast (smurf) flood
       Command        : commence mix flood
       Command        : commence targa3 attack
```

Mixed attack

TFN: the result

```
17:21:04.506166 eth0 > 194.49.187.0.46704 > 192.168.252.5.1896:
                        S 5170376:5170396(20) win 2671 urg 12565
17:21:04.516166 eth0 > 234.63.125.0.37201 > 192.168.252.5.30309:
                        S 11047630:11047650(20) win 1997 urg 19011
17:21:04.516166 eth0 > 39.213.139.0.7910 > 192.168.252.5.43813:
                        S 2125087:2125107(20) win 14958 urg 60724
17:21:04.516166 eth0 > 43.105.6.0.4744 > 192.168.252.5.3424:
                        S 6254394:6254414(20) win 33694 urg 42255
17:21:04.516166 eth0 > 66.217.70.0.22670 > 192.168.252.5.6337:
                        S 13843234:13843254(20) win 11437 urg 24737
17:21:04.516166 eth0 > 235.178.30.0.45851 > 192.168.252.5.30524:
17:21:04.516166 eth0 > 90.254.119.0.25388 > 192.168.252.5.31123:
17:21:04.516166 eth0 > 119.74.222.0.16422 > 192.168.252.5.6950:
17:21:04.516166 eth0 > 97.62.6.0.42978 > 192.168.252.5.10888:
17:21:04.516166 eth0 > 4.205.185.0.54120 > 192.168.252.5.6432:
17:21:04.516166 eth0 > 217.96.68.0.59220 > 192.168.252.5.65030:
17:21:04.516166 eth0 > 35.109.153.0.22810 > 192.168.252.5.15604:
17:21:04.516166 eth0 > 37.200.46.0.32360 > 192.168.252.5.52882:
17:21:04.516166 eth0 > 60.174.10.0.23938 > 192.168.252.5.3478:
17:21:04.516166 eth0 > 245.117.36.0.34314 > 192.168.252.5.61235:
17:21:04.516166 eth0 > 210.91.134.0.20053 > 192.168.252.5.12545:
```

Attack tools 4: TFN2K

- Like TFN, but Zombie almost always silent
  - Difficult to spot
  - Master sends commands 20x to zombies in the hope that one will get through
- Master to zombie communication is encrypted
- Attack signatures:
  - TCP header is always 0 length
  - UDP packet length (as appears in the UDP header) is 3 bytes longer than the actual length of the packet
  - UDP and TCP checksums do not include 12 byte pseudo-header and therefore checksums will always be incorrect

Attack tools 5: Stacheldracht

- Stacheldracht (v4 and v2.666)
  - Attacks: UDP, ICMP, TCP SYN, Smurf
  - Use encryption for communication but not for ICMP heartbeat packets that zombie sends to master
  - Auto-update feature via rcp
  - Has ability to test (via ICMP echo) if it can use spoofed IP addresses
  - V2.666 has added TCP ACK and TCP NUL attacks
  - Link: http://xforce.iss.net/alerts/advise61.php

Attack tools 6: Shaft

- Optional IP spoofing capabilities
- Ports:
  - Master to zombie: 18753/udp
  - Zombie to master: 20433/udp
  - An attack timer
  - Provides statistics to the master
  - Can set ICMP and UDP packet size
- Link: http://www.adelphi.edu/~spock/lisa2000-shaft.pdf

Attack tools 7: Mstream

- TCP port 12754
- Master to zombie via telnet
  - Communication not encrypted
- Attack: TCP ACK
  - Target gets hits by ACK packets and sends TCP RST packets to non-existent IP addresses
  - Router returns ICMP unreachable causing more bandwidth starvation
- Link: http://xforce.iss.net/alerts/advise48.php

Attack tools 8: Omega

- Spoof IP addresses
- Zombies use “chat”
- Attacks:
  - TCP ACK, UDP, ICMP
  - Introduced IGMP flood (multicast)
    - Internet Group Management Protocol
    - provides a way for an Internet computer to report its multicast group membership to adjacent routers

Attack tools 9: Trinity

- Also known as Myserver and Plague
- Attacks: UDP, TCP fragments, TCP SYN, TCP RST, TCP random-flag, TCP ACK, TCP establish, TCP NUL
- Listens to TCP port 3370
- When zombie is idle it connects to Undernet IRC on port 6667
- Link: http://xforce.iss.net/alerts/advise59.php

Attack tools 10: Ramen

- Self-propagating worm
- Scans /16s for port 21 (FTP)
- SYN scanning by ramen causes DDoS on IP multicast range
- Link: http://xforce.iss.net/alerts/advise71.php

Attack tools 11: Naphta

- Exploits weaknesses in TCP stacks with large number of connections in states other than "SYN RECVD," including "ESTABLISHED" and "FIN WAIT-1."
- Links:
  - http://razor.bindview.com/publish/advisories/adv_NAPTHA.html
  - http://www.cert.org/advisories/CA-2000-21.html

Attack tools 12: IRC bots

- Zombie systems controlled via a central IRC channel
- Uses Sub7 trojan to maintain remote control on zombies
- Links:
  - http://grc.com/dos/grcdos.htm
  - http://www.astalavista.com/security/ddos/ddos.shtml
  - http://www.cert.org/advisories/CA-2001-20.html

Attack tools 13: Worms

- Worms
  - Code Red, Power Worm, Nimda, SQL Voyager
  - All exploit Microsoft holes turning systems into zombies
  - Links:
    - http://www.cert.org/advisories/CA-2001-19.html
    - http://www.cert.org/advisories/CA-2001-23.html
    - http://www.cert.org/advisories/CA-2001-11.html
    - http://www.cert.org/advisories/CA-2001-26.html

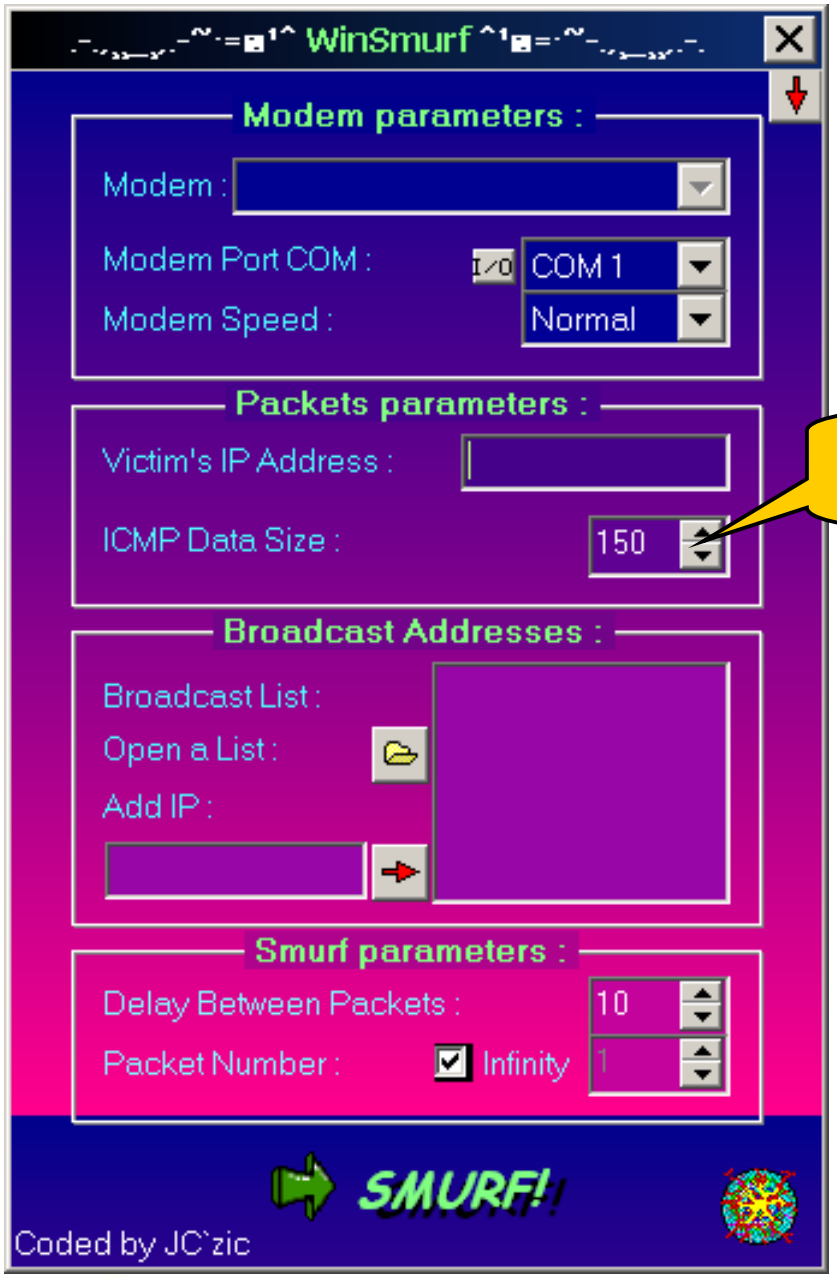
Attack tools 14: Routers

- Routers are being scanned
  - Pswd=cisco
- Using ICMP to packet a victim
  - Haven't discovered ttcp, yet!
- Juniper is FreeBsd derivative
  - Use your imagination

Hello y'all Jan 3, 2002

My name is Bubba, and down here in the south, we try some mighty fine things with these here Junipers. One day, I sat me down and thought long and hard about what to do with my router. Hect, you've got yourself a powerfur FreeBSD system on dat dare routing engine, and it's a bitching thing to use. Her are some of my ideas o how to use all of them thar idle cpu cycles:

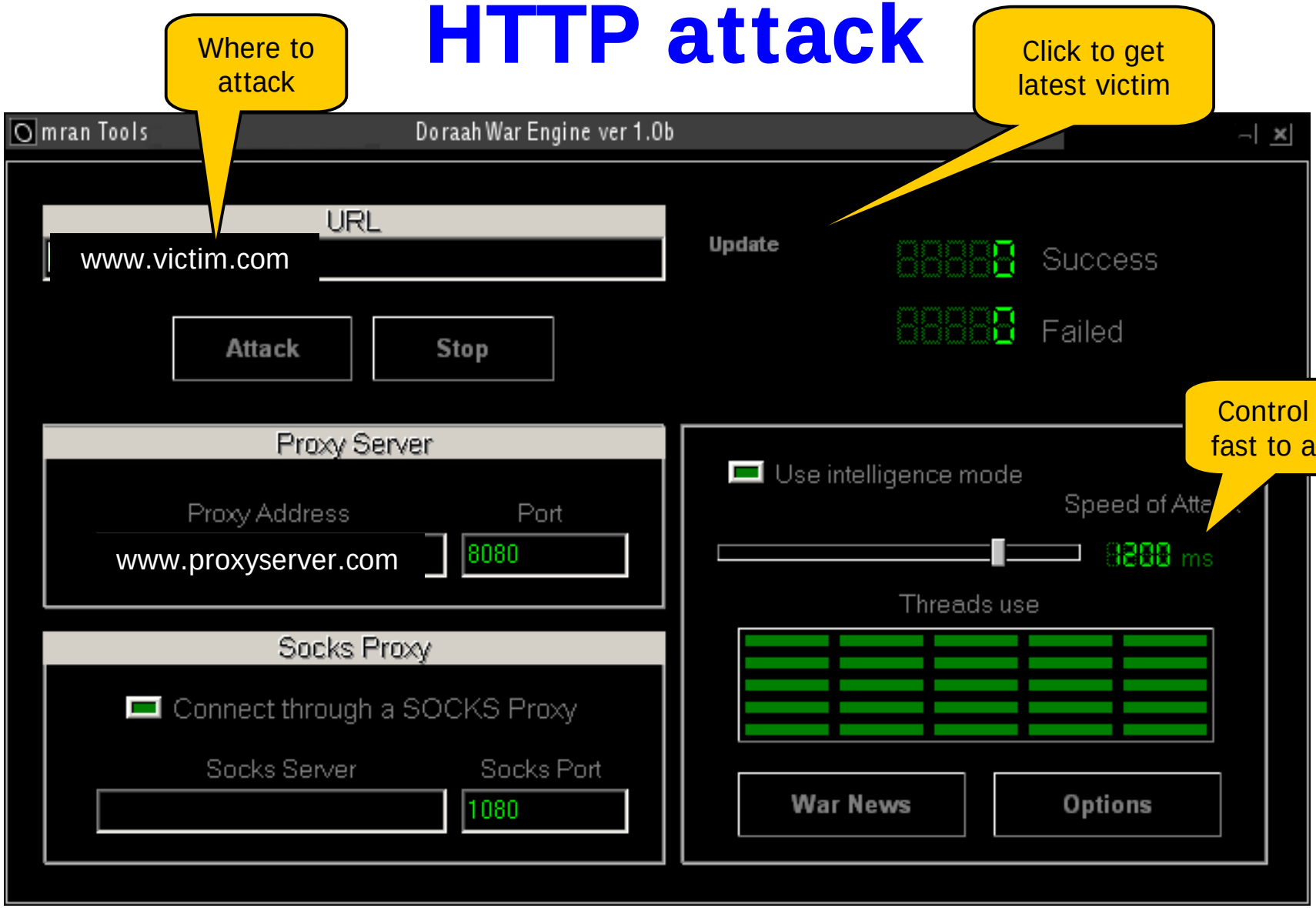
Smurf



Came out in March 1999!

Set packet size from 10 to 1300 octets

HTTP attack



First came out in January 1999!

Attack tools

- Others not covered:
  - Blitznet
    - http://packetstorm.decepticons.org/distributed/tfn3k.txt
  - Trank
  - Carko
    - http://www.securityfocus.com/archive/75/177265
  - Freak88
    - http://www.tlsecurity.net/backdoor/freak88.htm
  - Spank
  - Stick
    - http://xforce.iss.net/alerts/advise74.php

Summary of tools (1)

Name	Ammunition
Trinoo	UDP random ports
TFN/TFN-2K	Spoofed UDP/ICMP/TCP,SYN/Smurf
Stacheldracht v4/v2.666	SpoofedUDP, ICMP, TCP SYN, Smurf, TCP ACK, TCP NUL
FAPI	UDP, TCP SYN, TCP ACK, ICMP
Carko (Stacheldraht v1.666 + antigl + yps)	UDP, ICMP, TCP SYN, Smurf, TCP ACK, TCP NUL
Freak88	ICMP
Shaft	UDP, ICMP, TCP SYN
Mstream	TCP ACK
Blitznet	Spoofed IP floods
Ramen	Worm Multicast
Targa	Random ALL(TCP, UDP, long headers)
Spank	Multicast

Summary of tools (2)

Name	Ammunition
Trinoo	UDP random ports
TFN/TFN-2K	Spoofed UDP/ICMP/TCP,SYN/Smurf
Stacheldracht v4/v2.666	SpoofedUDP, ICMP, TCP SYN, Smurf, TCP ACK, TCP NUL
FAPI	UDP, TCP SYN, TCP ACK, ICMP
Carko (Stacheldraht v1.666 + antigl + yps)	UDP, ICMP, TCP SYN, Smurf, TCP ACK, TCP NUL
Freak88	ICMP
Shaft	UDP, ICMP, TCP SYN
Mstream	TCP ACK
Blitznet	Spoofed IP floods
Ramen	Worm Multicast
Targa	Random ALL(TCP, UDP, long headers)
Spank	Multicast

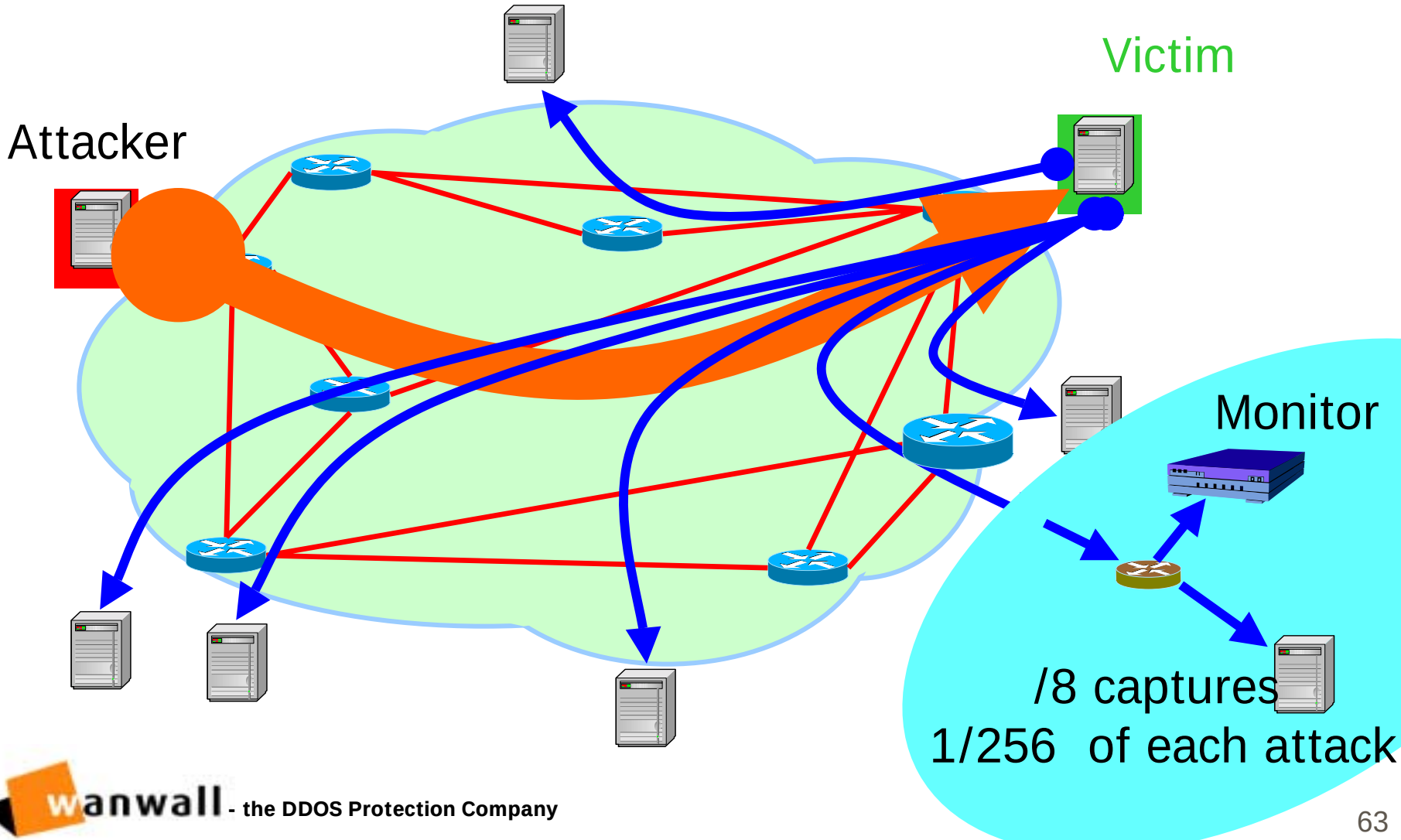
5. Statistics

Statistics CAIDA/UCSD

- 4,000 attacks per week
- 40 - 200 concurrent attacks / hour
- Most last 10 min's - 2 hours (avg 1/2 hour)
- Romania (15%) and Brazil (7%)

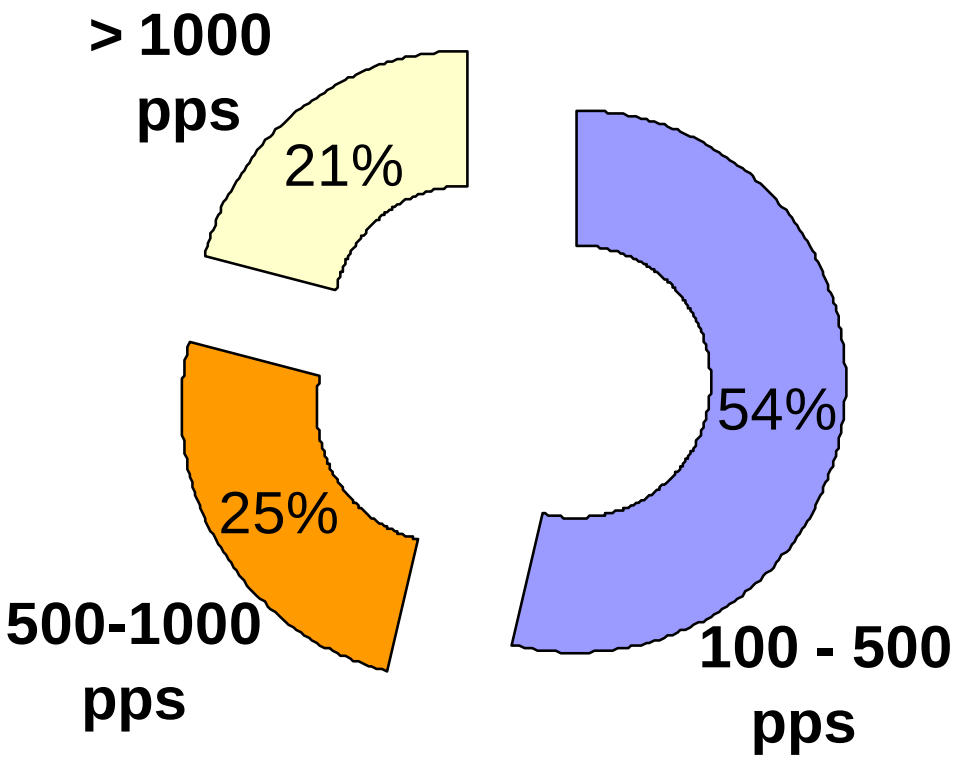
Backscatter CAIDA/UCSD

Moore, Voelker, Savage

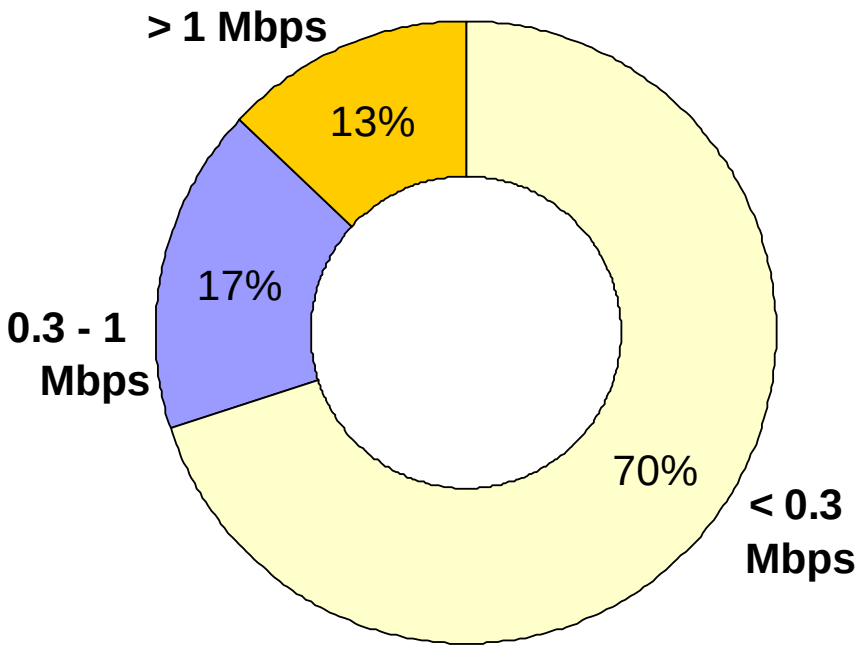


Attacks B/W

From: David Harmelin, DANTE



Highest: 27000 pps

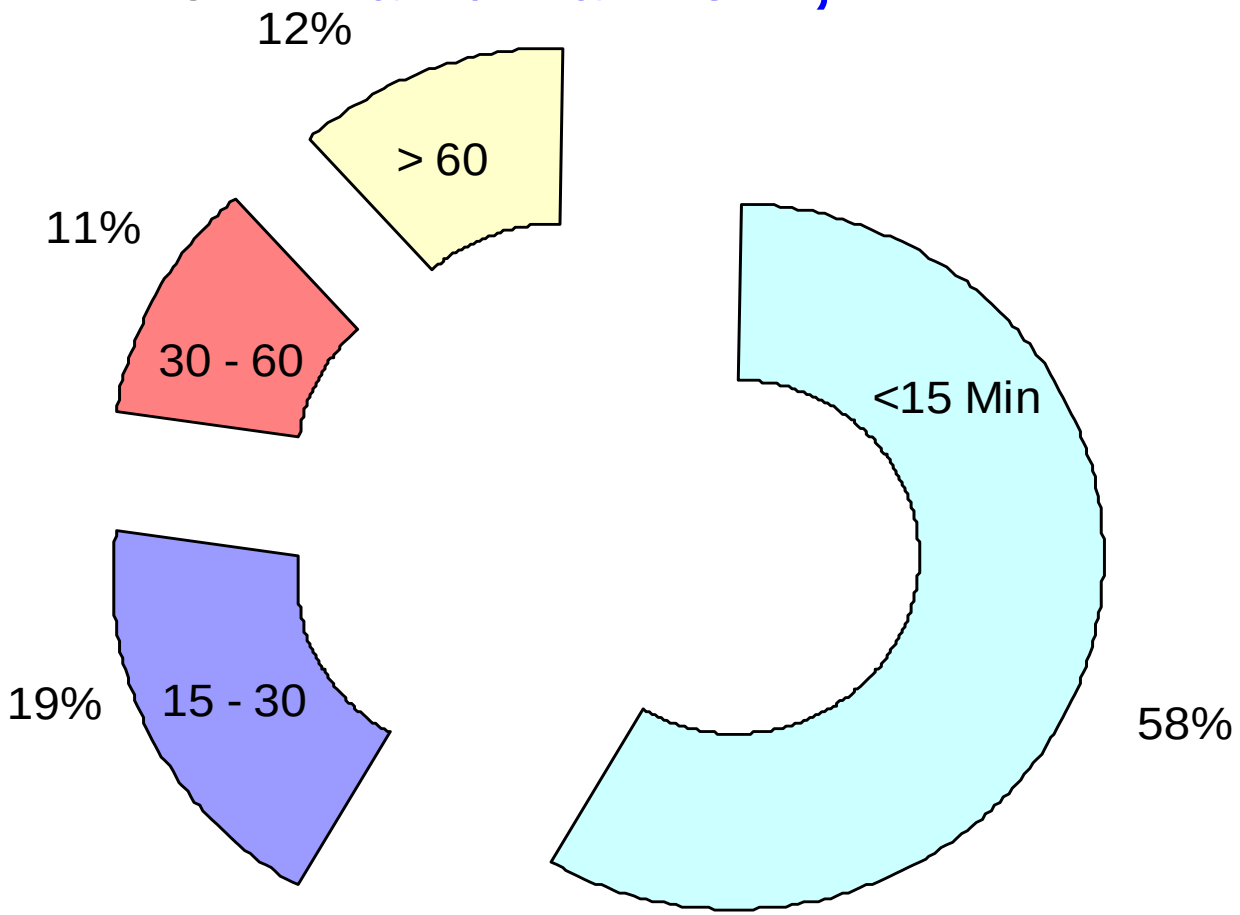


Highest: 32 Mbps

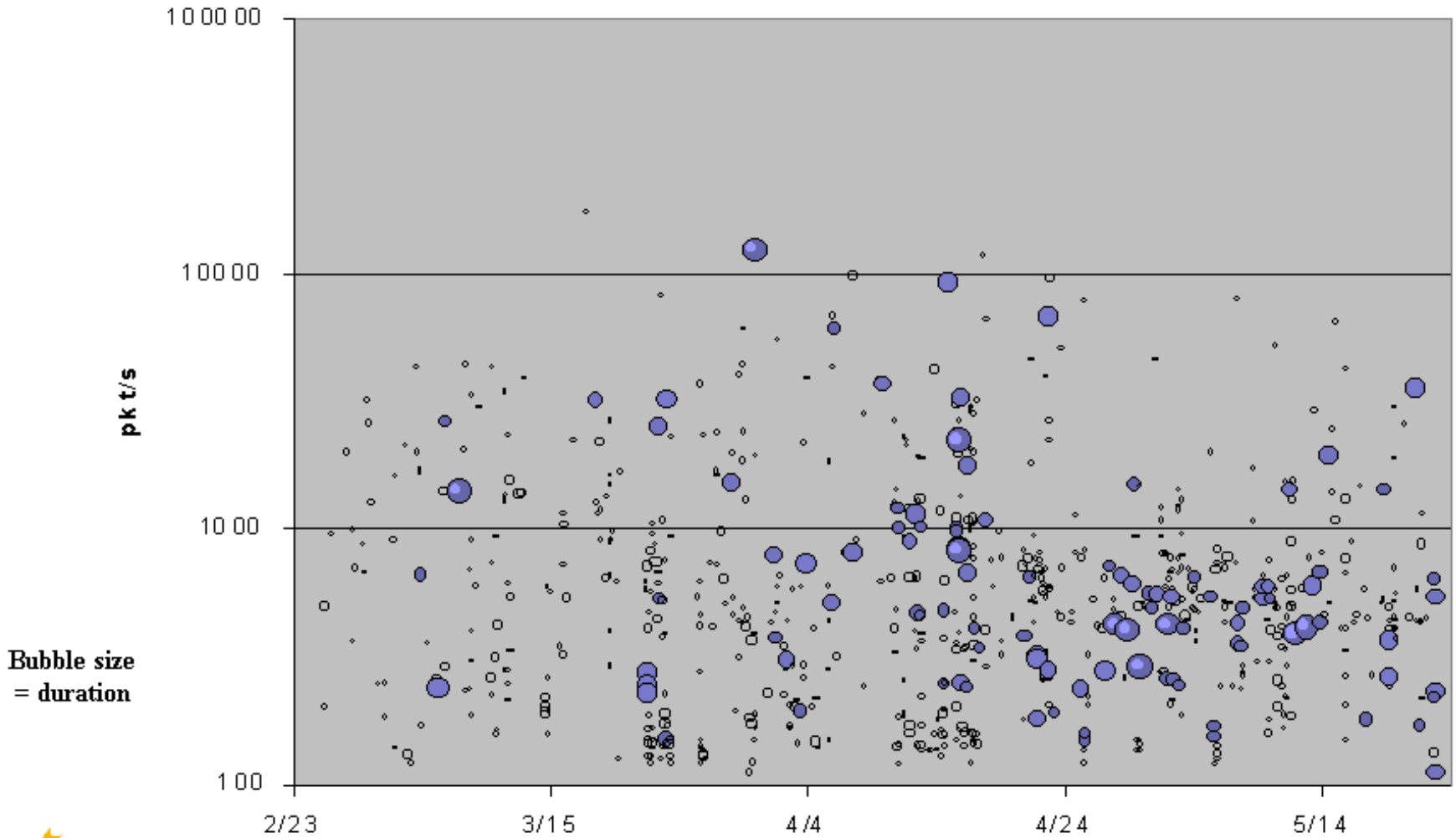
Approximate values only. Low accuracy due to sampling.

Attacks Duration

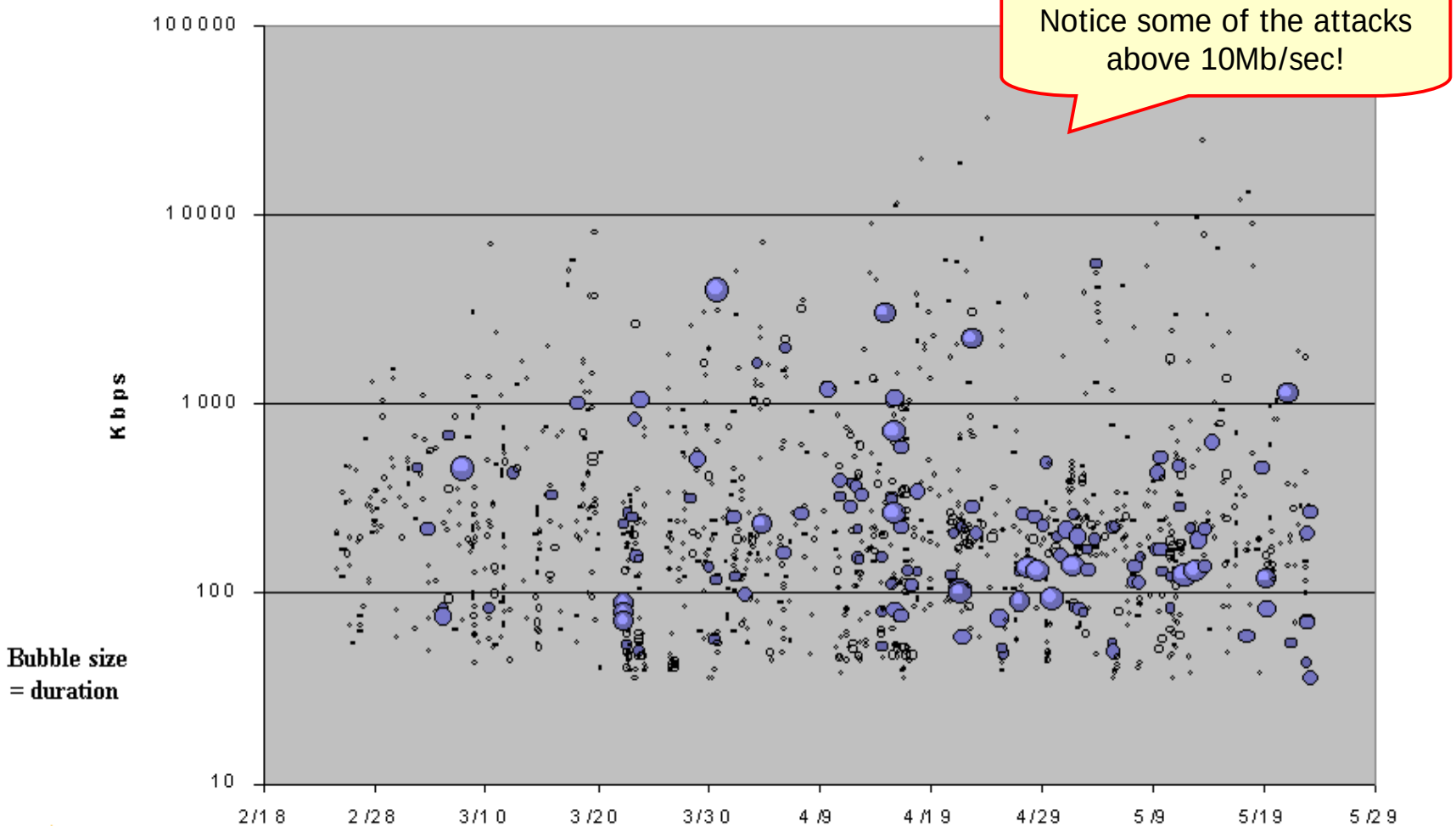
From: David Harmelin, DANTE



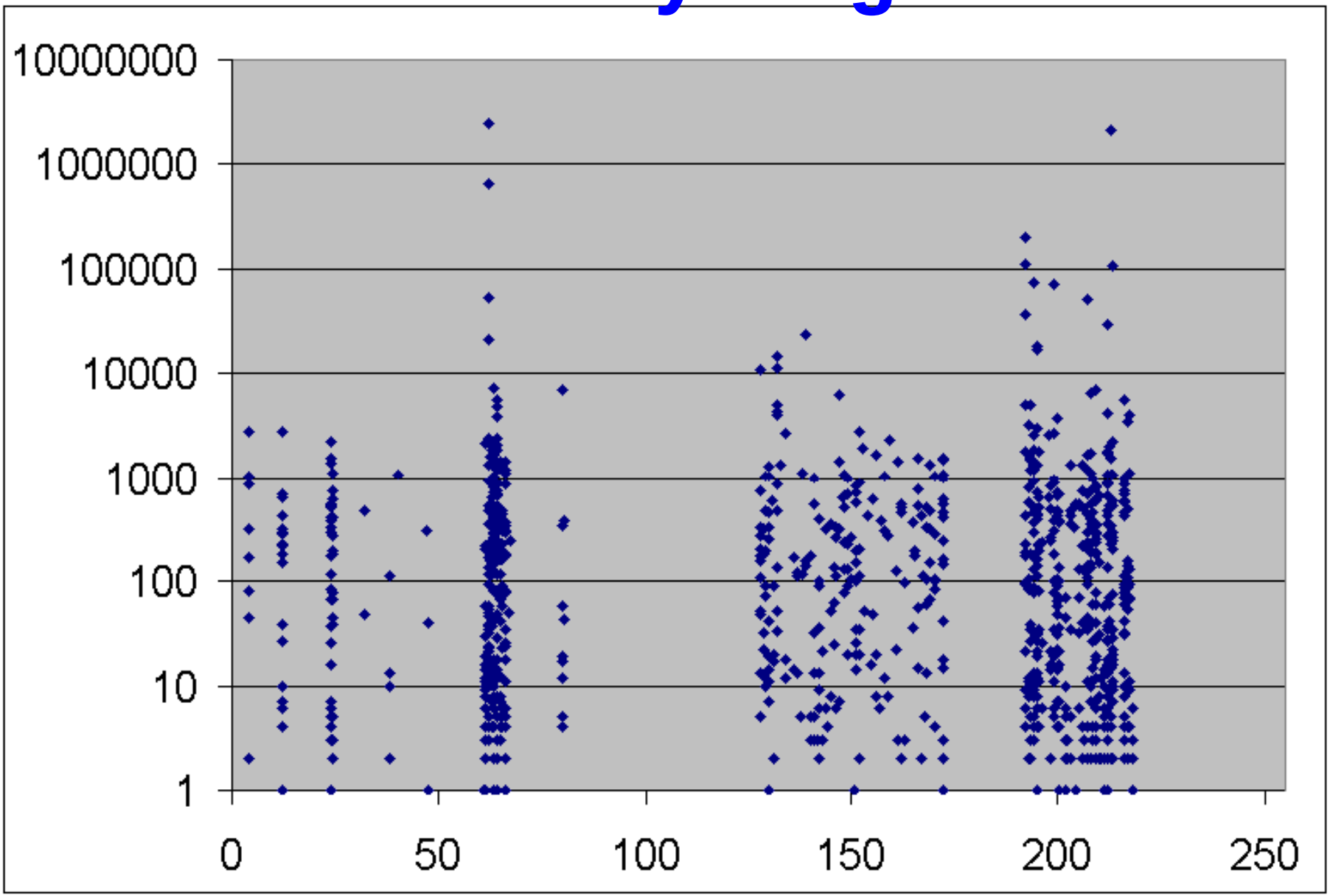
Results - All attacks (pkts/s)



Results - All attacks (Kbps)



Traffic history: Signature




SO, WHADDA YA WANT TO DO TODAY?... HANG OUT AT THE MALL, GO TO THE MOVIES OR SCREW UP HALF THE WORLD'S COMPUTERS?!...
PIZZA

6. Standards

Standards

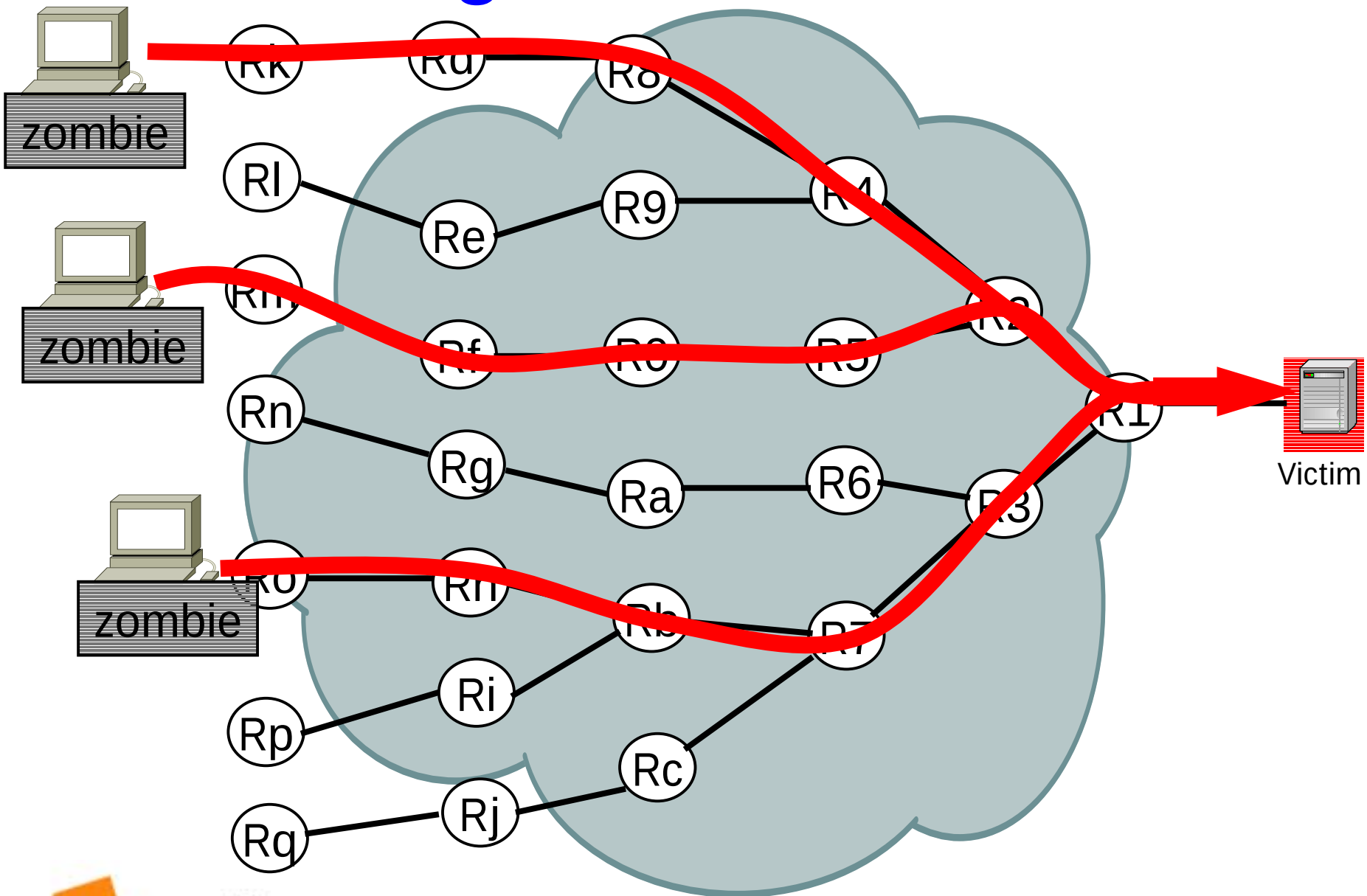
- Itrace IETF working group
  - Chaired by Steve Bellovin
- Two drafts
  - ICMP Traceback Messages [Bellovin]
  - Intention-Driven ICMP Trace-Back [Massey, Mankin]
- http://www.ietf.org/html.charters/itrace-charter.html

7. Academia

Lots of academic/research work

- Traceback [Bellovin00, Savage et.al.00, Burch&Cheswick99, Wu et.al.00, Snoeren et. al.01]
- CenterTrack [Stone]
- MULTOPS [Gil&Poletto]
- Pushback [Bellovin]

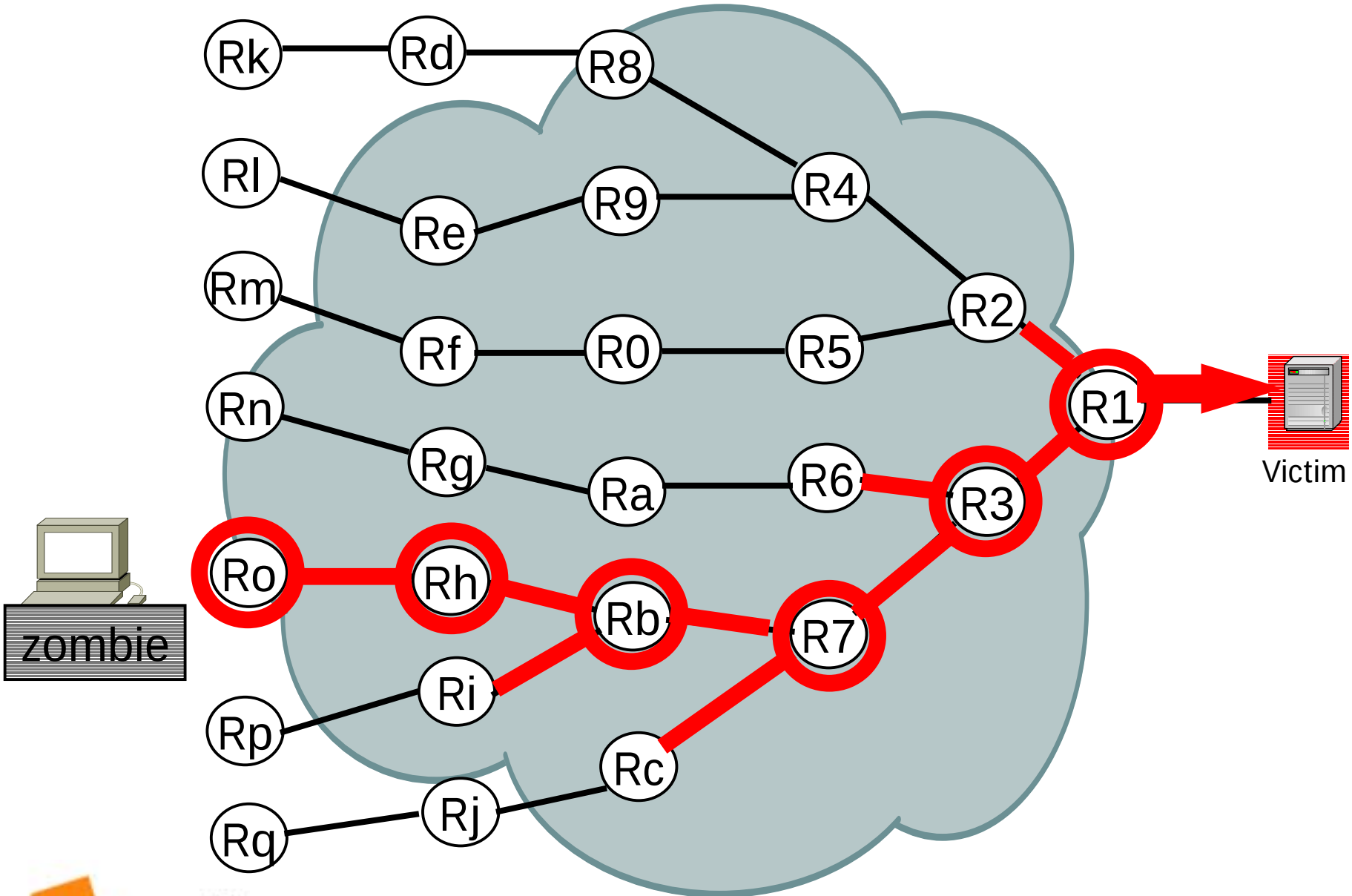
Tracing the attack route



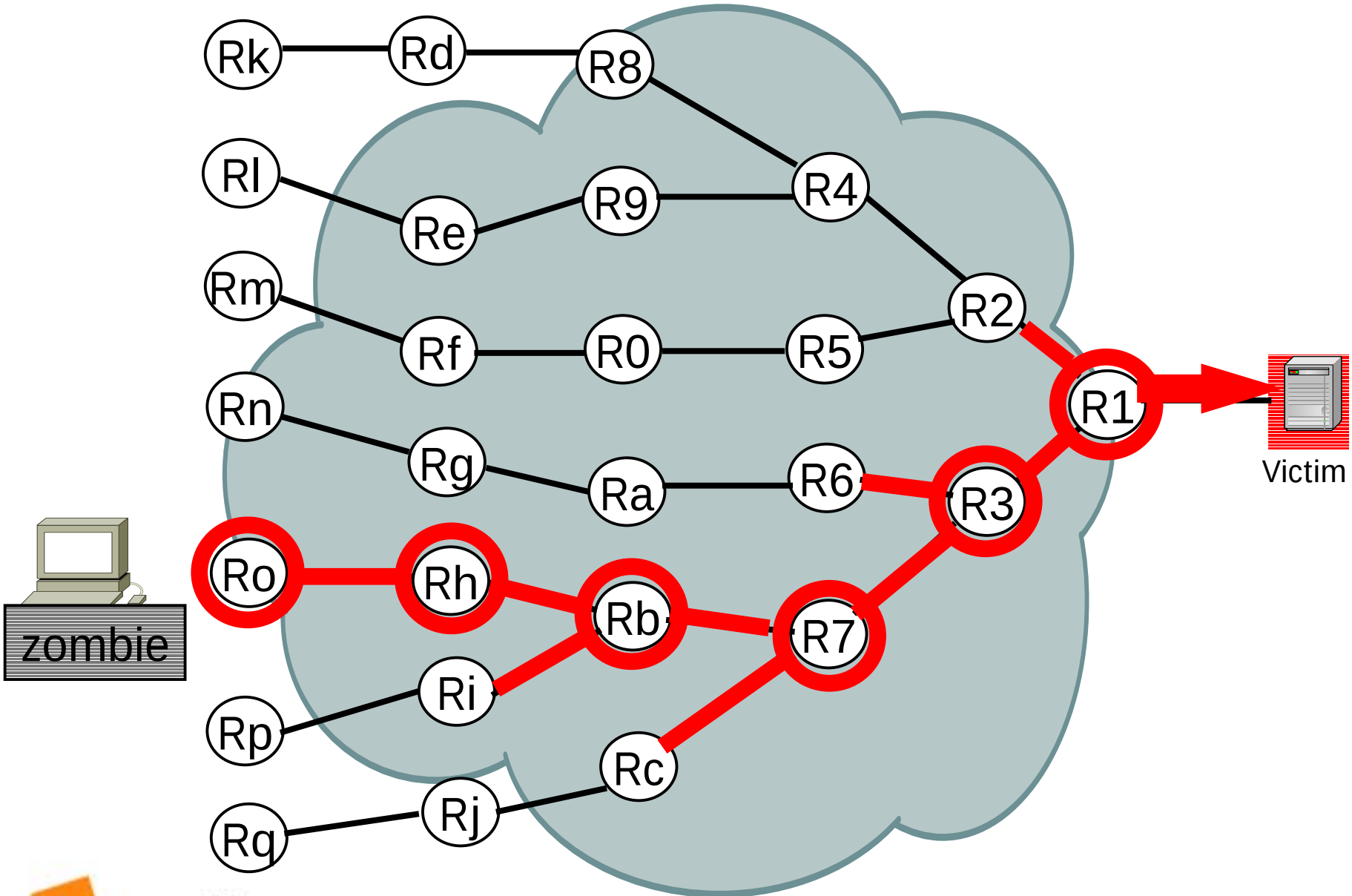
Tracing the attack route

1. Hop by hop, router by router [Cisco,Juniper]
2. Effect on network [Burch&Cheswick99]
3. Auditing / probing the route
   - Packet marking [Savage et.al.00]
   - ICMP reports [Bellovin]
   - Out of band [Stone CenterTrack]
4. Logging [Snoeren et. al.]

Hop by hop, router by router



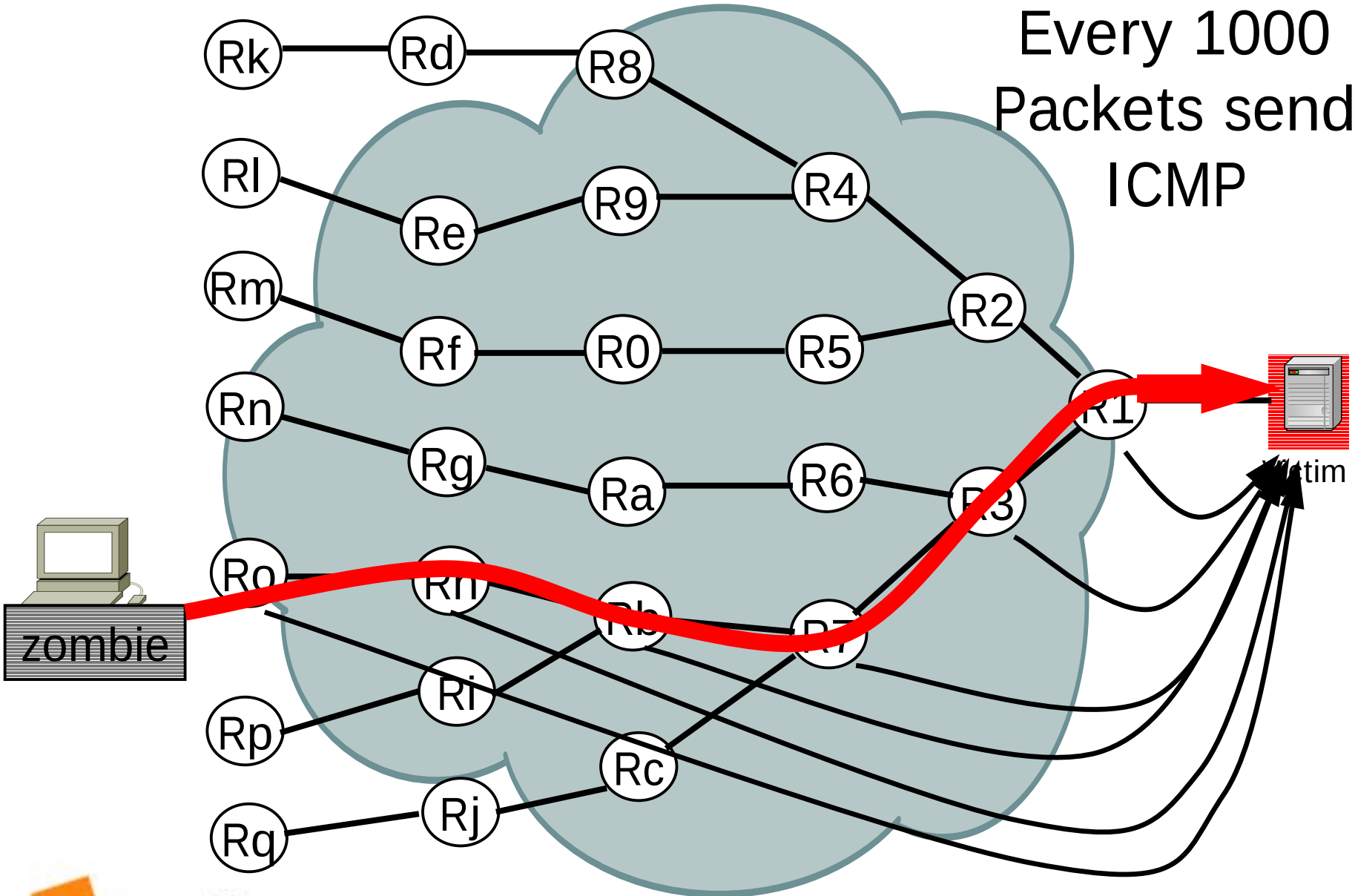
Effect on links BurchCheswick



Traceback (1)

- ICMP traceback (Bellovin)
- For very few packets (1/20000) every router, copy the content into a special ICMP traceback message containing the info about the previous/next routers along the path
- Victim reconstructs the path to the attacker
- Problems:
  - Creates more traffic – about .1%
  - Authentication
  - Load on routers
  - Some firewalls block ICMP traffic

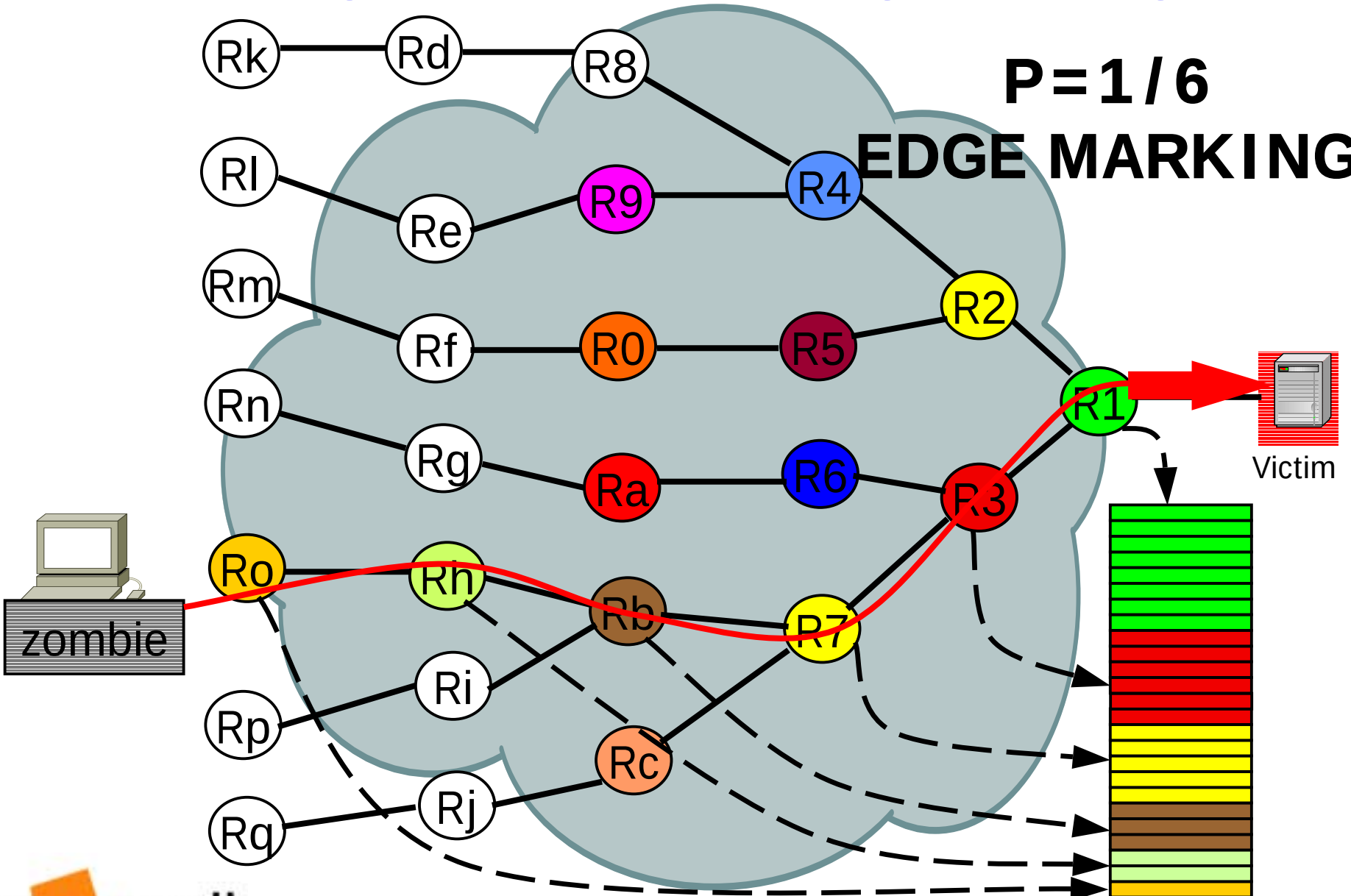
Bellovin, Wu et.al: TraceBack ICMP



Traceback Savage et.al.

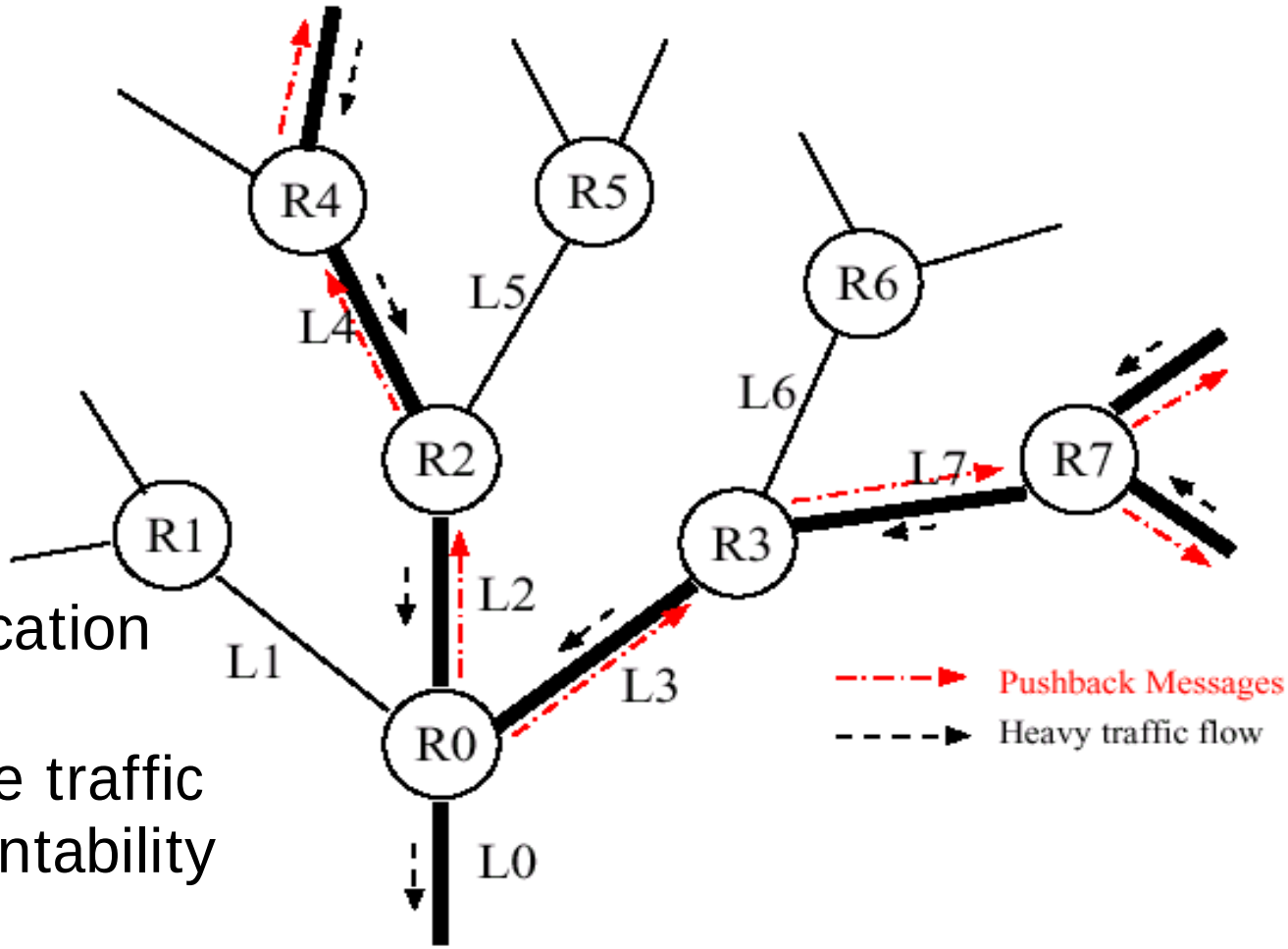
- Savage, Wetherall, Karlin & Anderson
- Router marks the packet, with probability p, with the next hop the packet will flow thru
- For a large flow of packets this method can determine the path and source
- Marking in the IP header in ID field
  - Problematic for fragmentation

Savage et. al., Song-Perrig



Pushback - Bellovin AT&T

Data Flow



- Authentication
- Fairness
- legitimate traffic
- Implementability

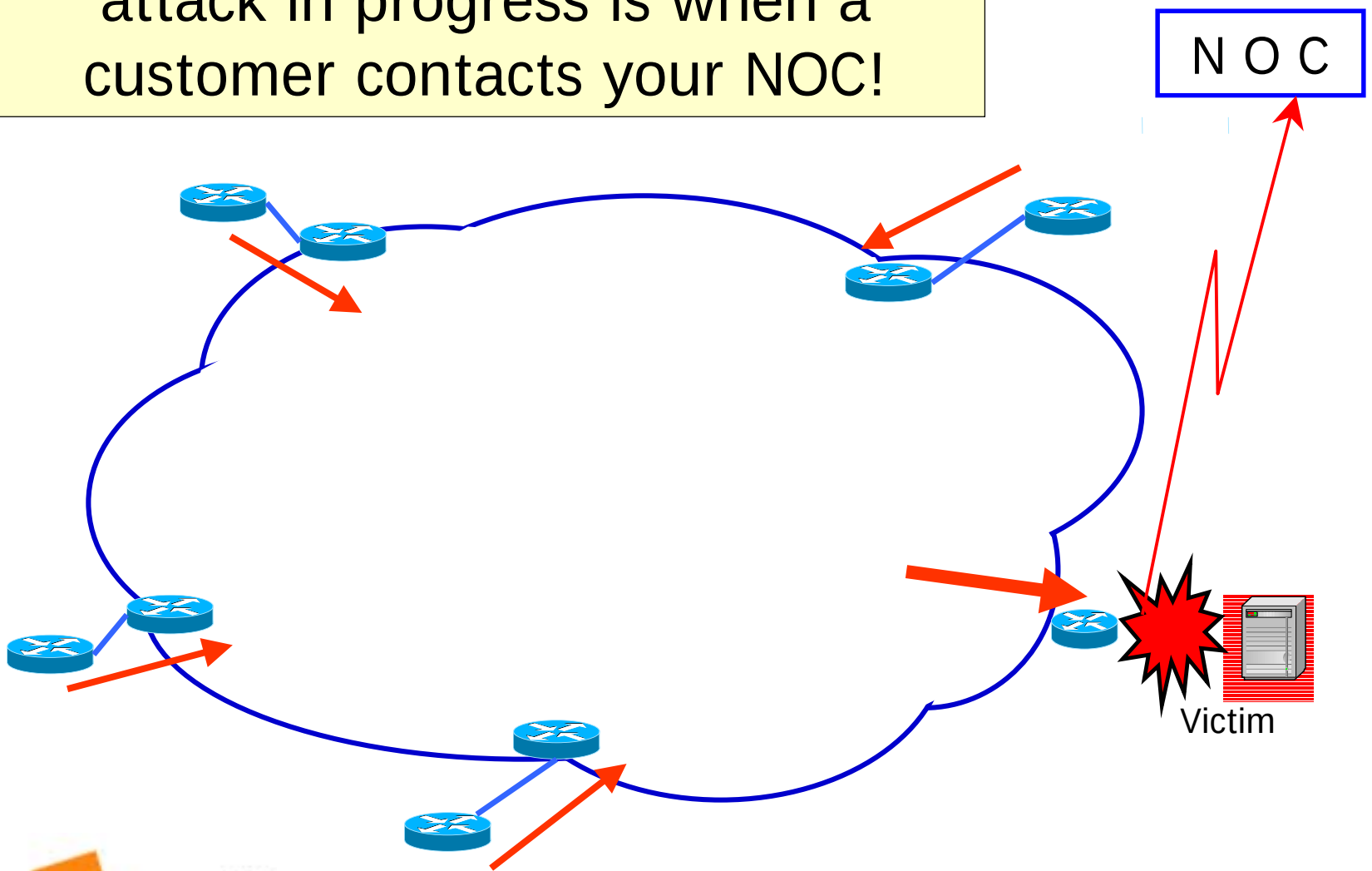
8. Detection

Detection four approaches

- Tracing
- Netflow
- Optical splitters / port mirroring
- Remote monitoring

NOC

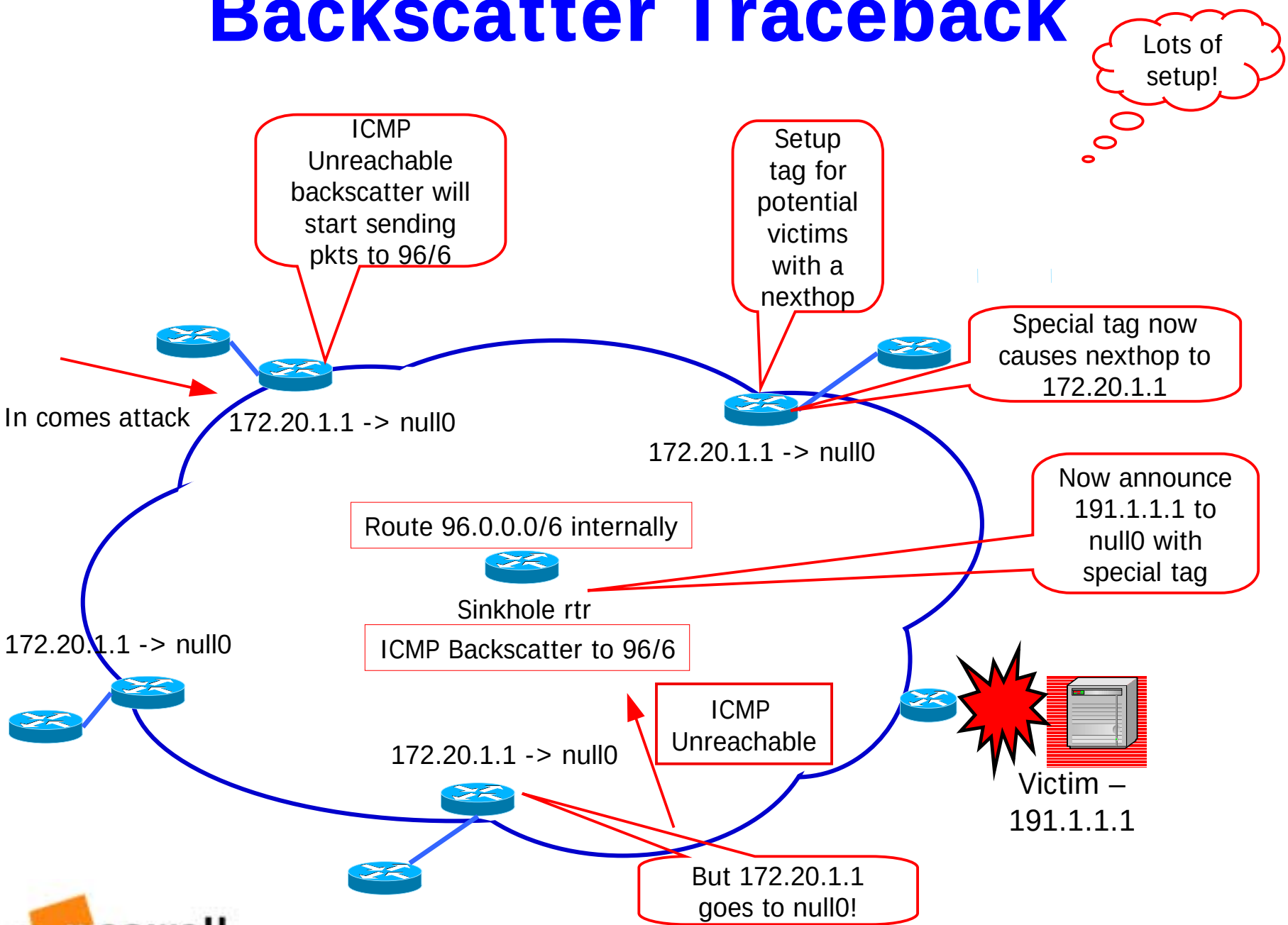
The #1 way to know there is an attack in progress is when a customer contacts your NOC!



Backscatter Traceback

- Technique designed by Chris Morrow and Brian Gemberling of UUnet
  - http://www.secsup.org/Tracking/
- **Concept**: Packets whose destination is unreachable will have ICMP Unreachable sent back to the source.
  - This “unreachable noise” is Backscatter Traceback
  - Requires a large “unused” block to be only internally routed

Backscatter Traceback



Backscatter Traceback (2)

- Routers require ICMP Unreachables working
  - **`no ip unreachables`** has to be turned on
- Sinkhole router advertises the prefix under attack (/32)
  - **`ip route victimip 255.255.255.255 null0 tag 666`**
- Cons
  - Complex method
  - Time consuming
  - Doesn't stop the attack – just tells you from where it is coming
  - Routers meant to forward – not drop packets

Cisco Netflow - 1

- Operates in conjunction with CEF
  - Enabled on a per interface basis
  - If CEF not running then Netflow switching will be enabled

    ```
    interface FastEthernet0/0
    ip route-cache flow
    ```

  - Shows flows into the interface
    - Number of flows, packet size, activity, etc.

Cisco Netflow - 2

Most pkts are small

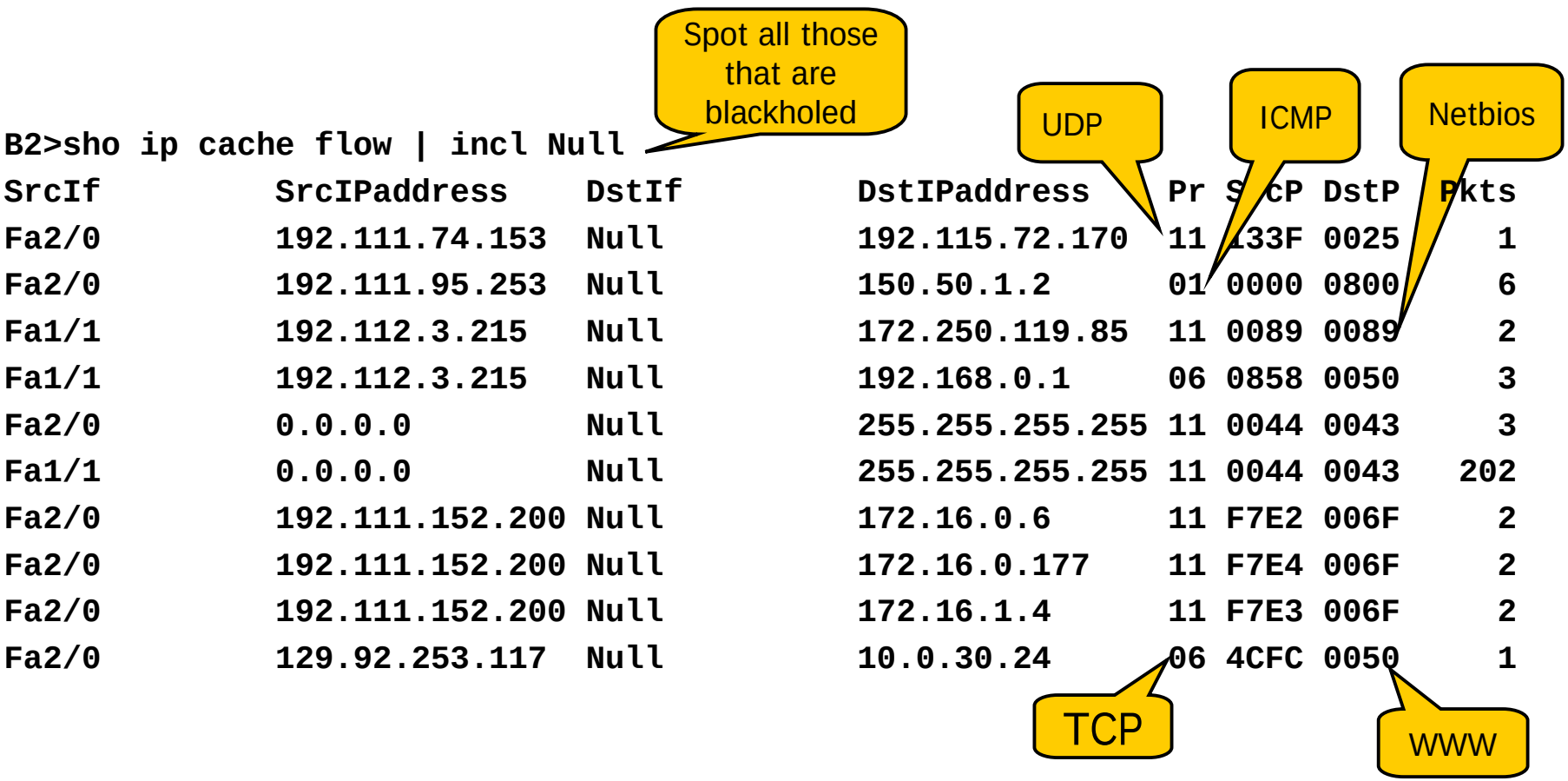
```
B2>sho ip cache flow
IP packet size distribution (71156M total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .002 .581 .090 .024 .011 .010 .010 .006 .003 .004 .003 .003 .003 .003 .003

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .004 .003 .124 .011 .093 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 4456704 bytes
  17047 active, 48489 inactive, 4010292907 added
  2115225614 ager polls, 0 flow alloc failures
```

Protocol	Total Flows	Flows /Sec	Packets /Flow	Bytes /Pkt	Packets /Sec	Active(Sec) /Flow	Idle(Sec) /Flow
TCP-Telnet	5903492	1.3	8	156	12.3	9.3	19.9
TCP-FTP	41468046	9.6	5	252	49.1	10.1	18.4
TCP-WWW	2473587049	575.9	8	345	4882.8	4.0	18.7
TCP-BGP	885358	0.2	5	179	1.1	19.5	20.2
TCP-Frag	60544	0.0	7	101	0.1	5.1	19.6
TCP-other	564343726	131.3	28	444	3680.2	14.1	18.8
UDP-DNS	296006951	68.9	3	78	214.6	5.0	21.7
UDP-Frag	213461	0.0	143	320	7.1	60.7	21.5
UDP-other	365140346	85.0	72	73	6142.9	10.3	20.9
ICMP	183652930	42.7	2	221	113.3	4.0	21.6
IGMP	126	0.0	2186	700	0.0	93.9	23.5
GRE	533375	0.1	1144	384	142.1	50.7	21.4
IP-other	5632527	1.3	191	445	250.4	55.9	21.1
Total:	4010276236	933.7	17	275	16566.4	6.5	19.3

Cisco Netflow - 3




```
B2>sho ip cache flow | incl Null
SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Fa2/0         192.111.74.153  Null          192.115.72.170  11 133F 0025     1
Fa2/0         192.111.95.253  Null          150.50.1.2      01 0000 0800     6
Fa1/1         192.112.3.215   Null          172.250.119.85  11 0089 0089     2
Fa1/1         192.112.3.215   Null          192.168.0.1     06 0858 0050     3
Fa2/0         0.0.0.0         Null          255.255.255.255 11 0044 0043     3
Fa1/1         0.0.0.0         Null          255.255.255.255 11 0044 0043   202
Fa2/0         192.111.152.200 Null          172.16.0.6      11 F7E2 006F     2
Fa2/0         192.111.152.200 Null          172.16.0.177    11 F7E4 006F     2
Fa2/0         192.111.152.200 Null          172.16.1.4      11 F7E3 006F     2
Fa2/0         129.92.253.117  Null          10.0.30.24      06 4CFC 0050     1
```

Cisco Netflow - 4

- Can use Unix to find attackers
  - Capture complete **`sho ip cache flow`** data
  - Sorted by column 2 (source)
    - **`awk '{print $2}' /tmp/data | sort | uniq –c | sort –rn | head`**

```
842 123.1.1.1
234 191.2.2.2
212 192.4.4.4
```

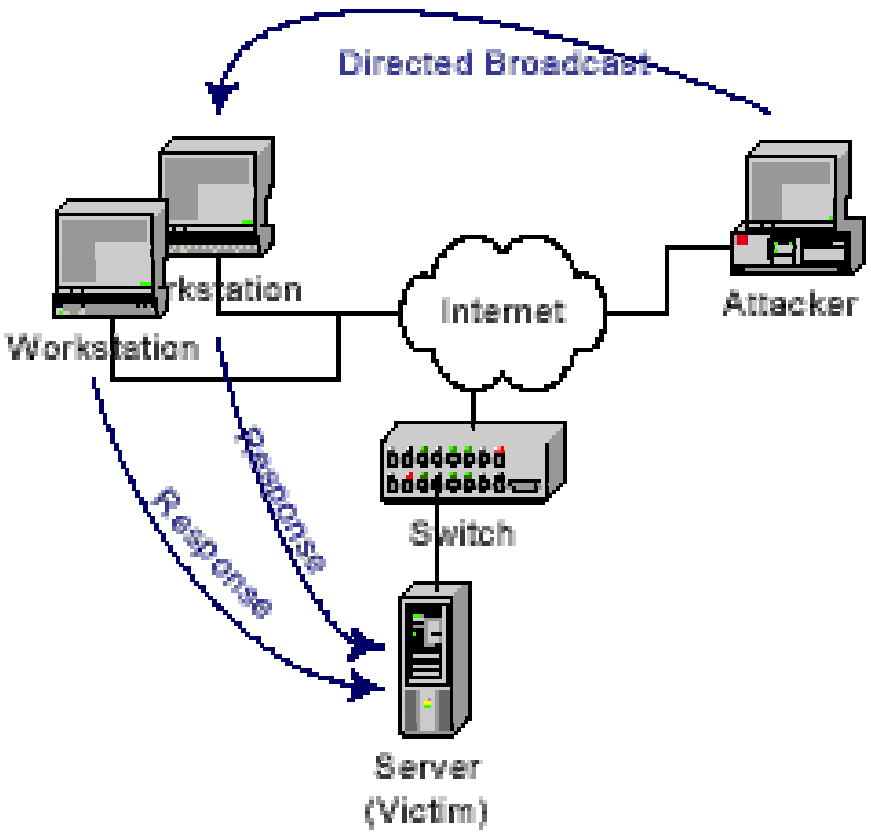
Could be proxy servers

  - Sorted by column 4 (destination)
    - **`awk '{print $4}' /tmp/data | sort | uniq –c | sort –rn | head`**

```
2341 192.111.2.2
1563 192.110.1.1
1211 125.2.3.1
```

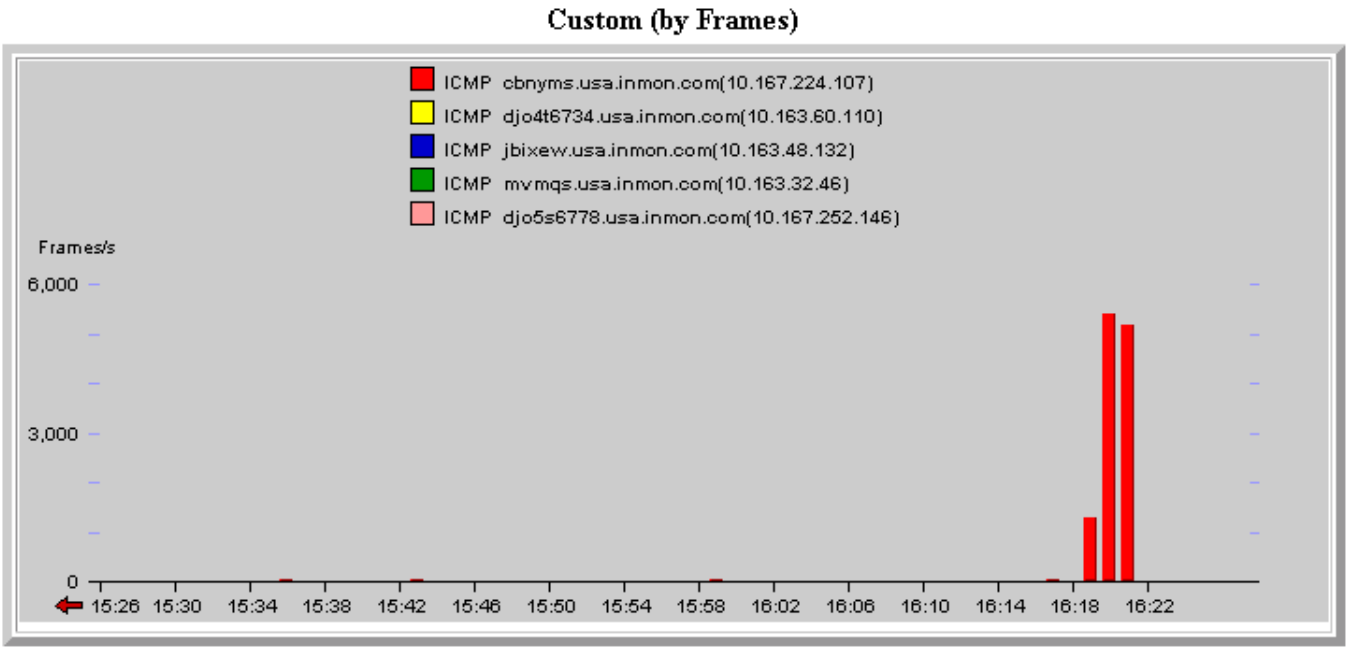
Inmon - 1

- Traffic Server – simplified Netflow processor



Inmon -2

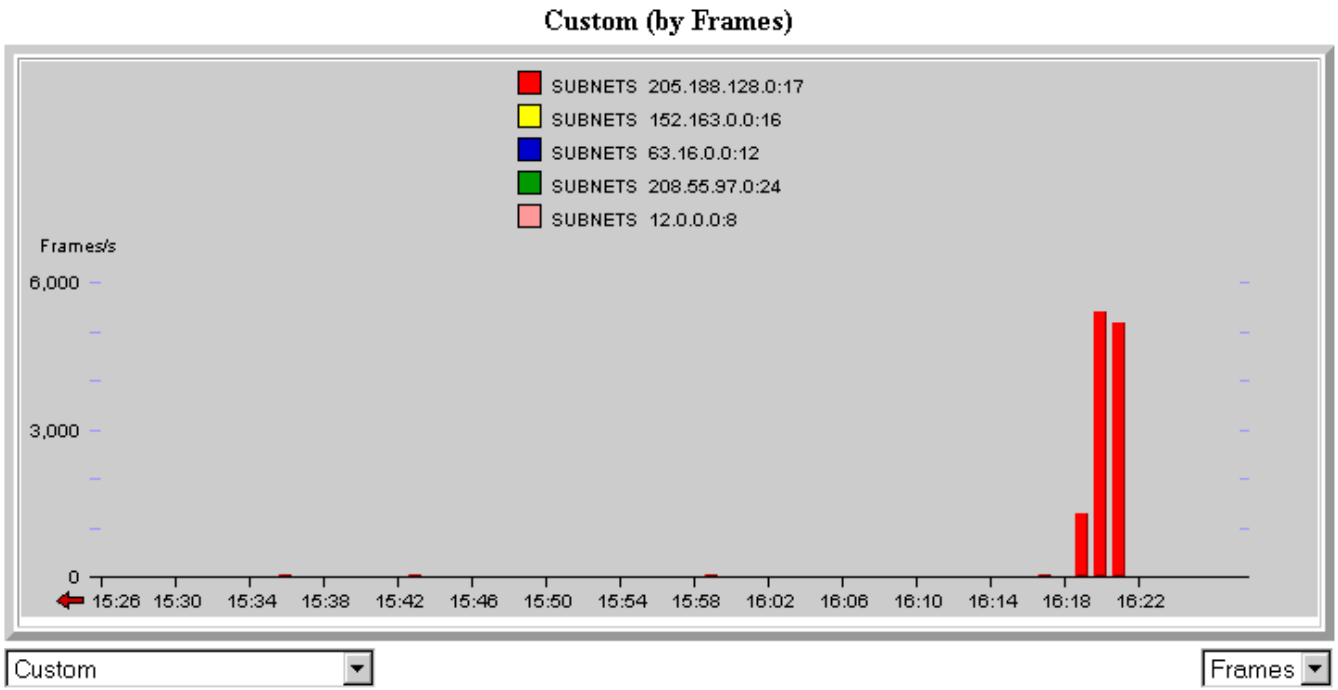
- Who is being attacked?



		Source		Destination	
	Protocol	Address	Port	Address	Port
Filter	ICMP				Echo_Rep
Output	ALL	☐ Include	☐ Include	☑ Include	☐ Include
		Submit	Reset		

Inmon - 3

- Where is the attack coming from?

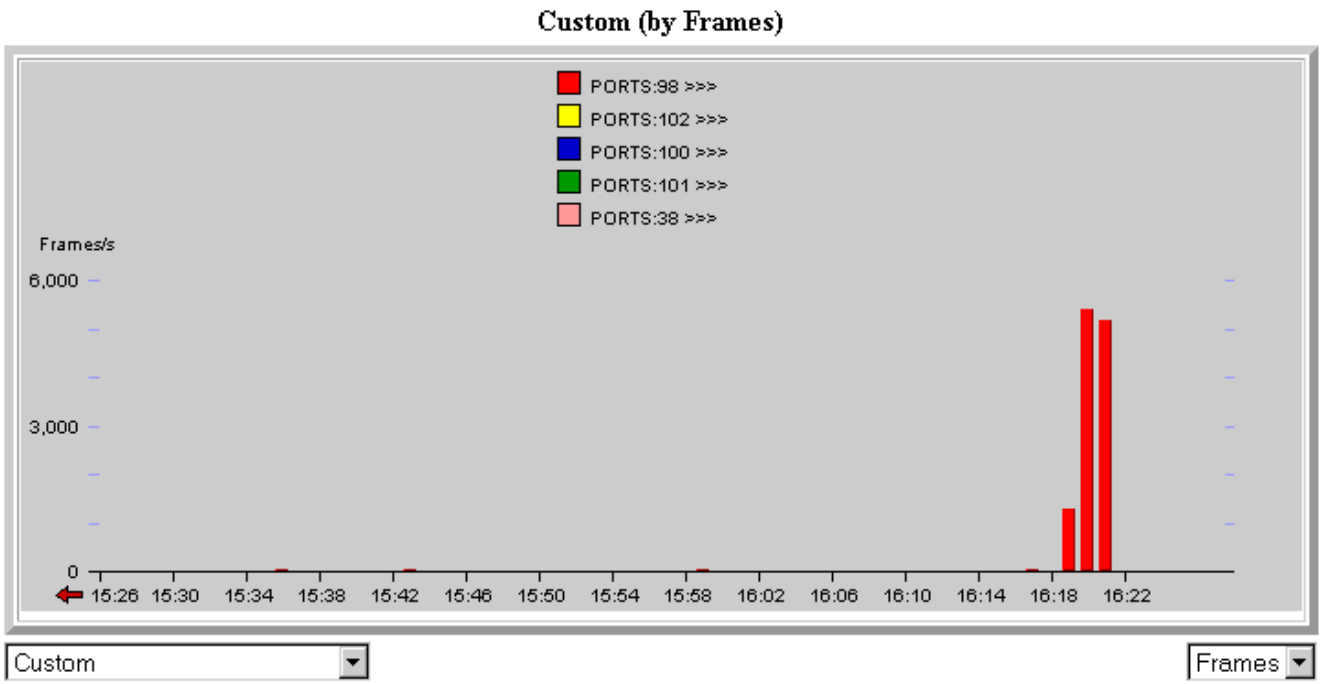


		Source		Destination	
	Protocol	Address	Port	Address	Port
Filter	ICMP			10.167.224.107	Echo_Rep
Output	SUBNETS	☑ Include	☑ Include	☐ Include	☐ Include

Submit Reset

Inmon - 4

- On which interface?

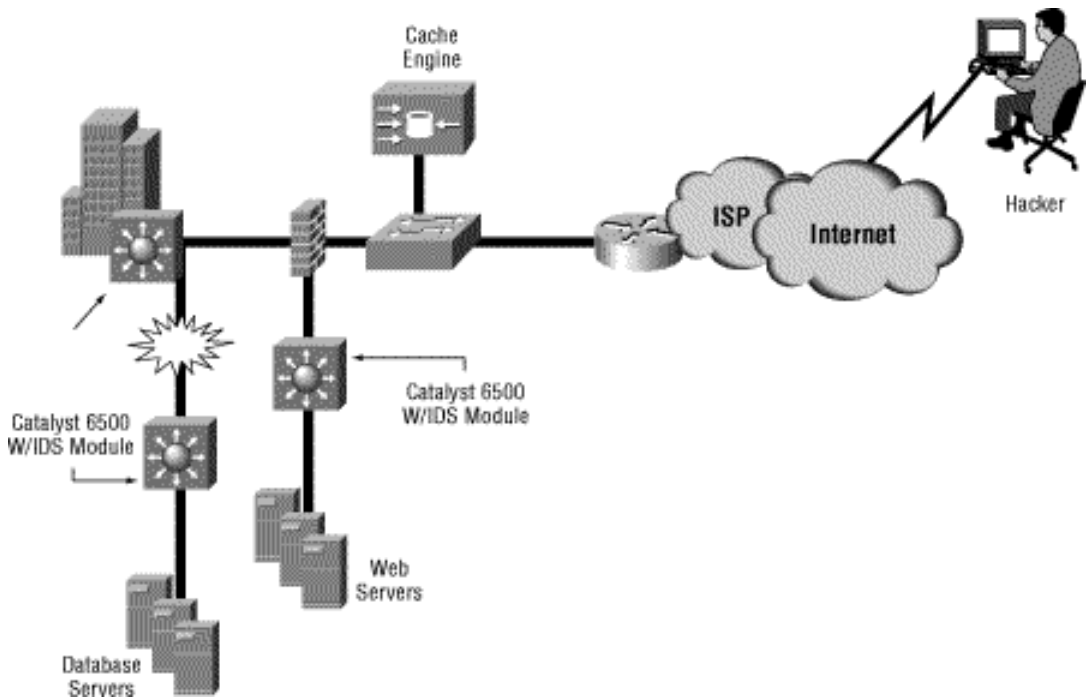


		Source		Destination	
	Protocol	Address	Port	Address	Port
Filter	ICMP			10.167.224.107	Echo_Rep
Output	PORTS	☐ Include	☑ Include	☐ Include	☐ Include

Submit Reset

IDS

- Cisco Catalyst IDS
- Handles 47Kpps



Optical Splitter

NLANR
National Laboratory for Applied Network Research

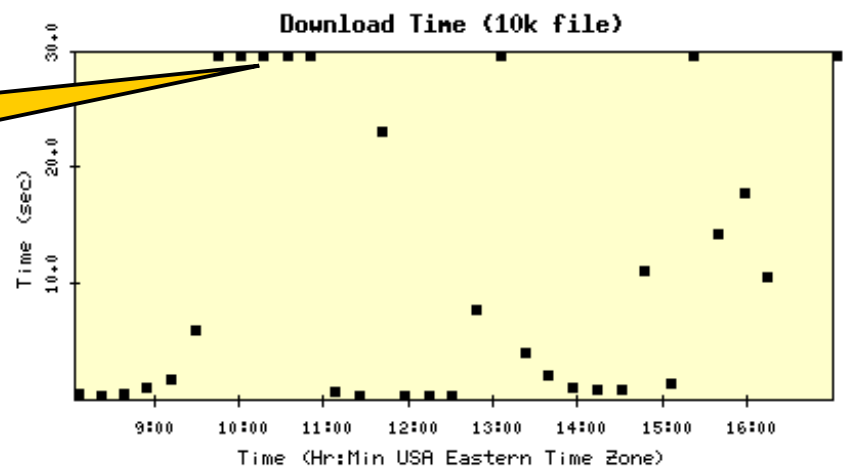
Optical splitters

Alert services

www.websitepulse.com
www.internetseer.com
www.alerta.com
www.siteprobe.com
www.sitepatrol.com
www.mirror-image.com
www.empirix.com
www.siteseer.com
www.isitrunning.com

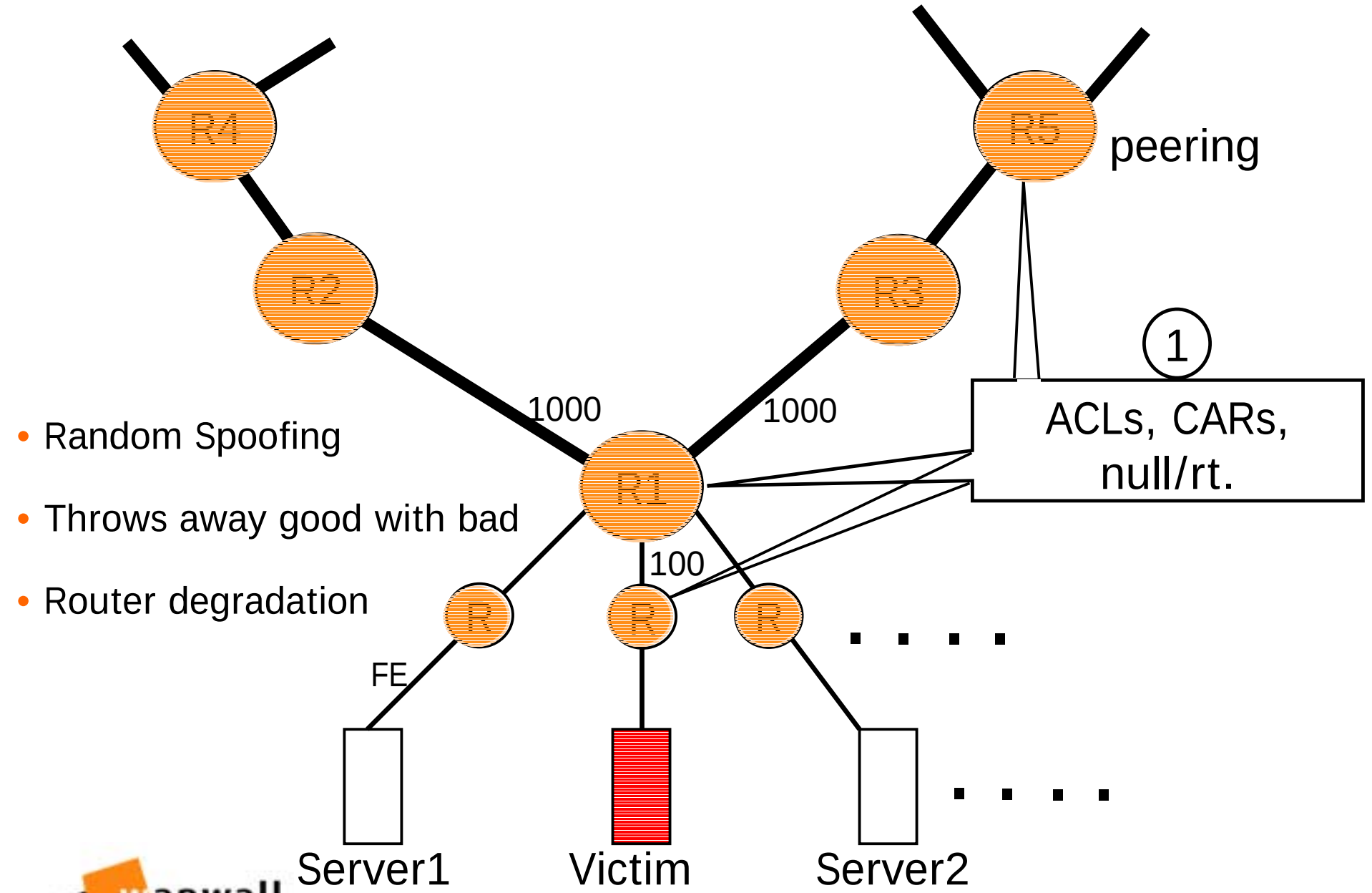
- Keynote
  - Red Alert
    - Email or pager alerts if site becomes unavailable
- NetMechanic
  - Server Check
    - Customized by user: server too slow

Response time went too high!



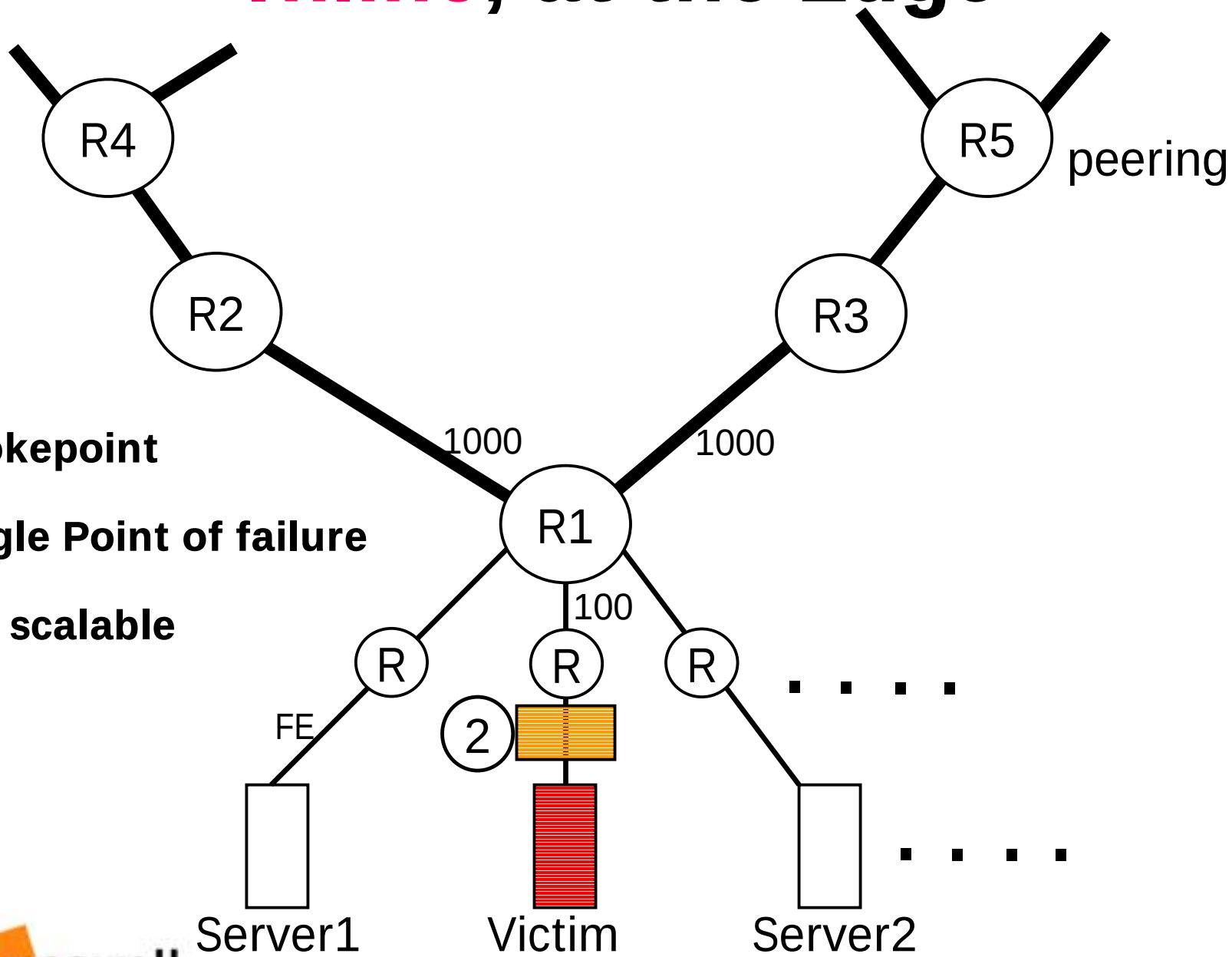
9. Protection and Defense

At the Routers



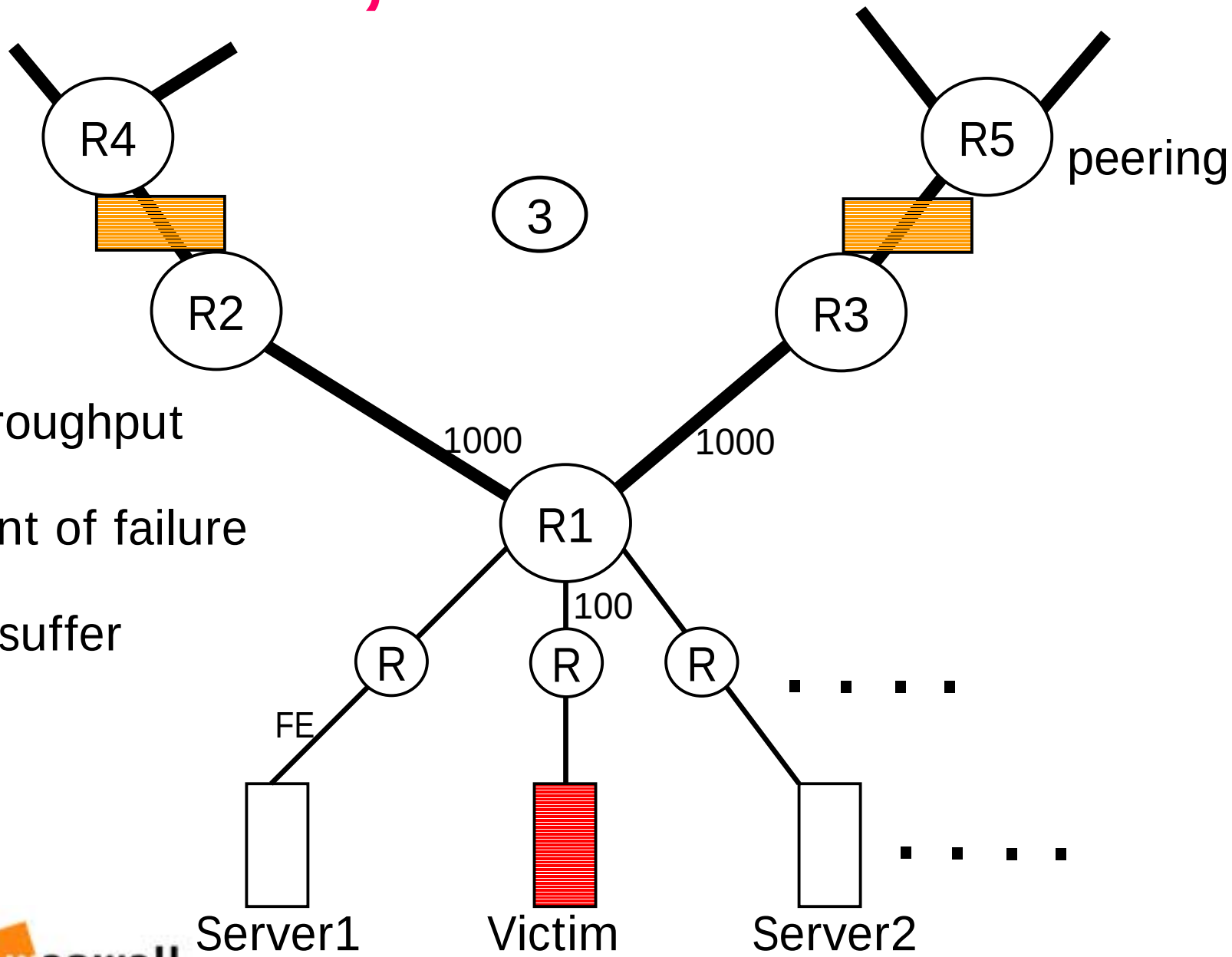
- Random Spoofing
- Throws away good with bad
- Router degradation

Inline, at the Edge



- **Chokepoint**
- **Single Point of failure**
- **Not scalable**

Inline, on the Back Bone



- Throughput
- Point of failure
- All suffer

Router based protection

Cisco ACLs - 1

- Use ACL to determine which interface is being attacked and characteristics of attack
  - Initial ACL to determine what type of attack

```
access-list 101 permit icmp any any echo
access-list 101 permit icmp any any echo-reply log-input
access-list 101 permit udp any any
access-list 101 permit tcp any any
access-list 101 permit ip any any

interface serial 1/1
ip access-group 101 out
! Wait 10 seconds
no ip access-group 101 out
```

Cisco ACLs - 2

- `sh access-l 101`

```
Extended IP access list 101
    permit icmp any any echo (2 matches)
    permit icmp any any echo-reply (21374 matches)
    permit udp any any (18 matches)
    permit tcp any any (123 matches)
    permit ip any any (5 matches)
```

- Indications are that there is some sort of ICMP attack
  - Need to place ACL on each successive router in upstream path

Cisco ACLs - 3

- Next use ‘`log-input`’ to determine from where – via ‘`sho logging`’:

```
%SEC-6-IPACCESSLOGDP: list 101 permit icmp 192.168.1.1
  (Serial1/1) -> 128.139.19.5 (0/0), 1 packet
%SEC-6-IPACCESSLOGDP: list 101 permit icmp 172.17.3.34
  (Serial1/1) -> 128.139.11.2 (0/0), 1 packet
%SEC-6-IPACCESSLOGDP: list 101 permit icmp 192.168.2.15
  (FastEthernet1/0/0) -> 128.139.6.1 (0/0), 1 packet
%SEC-6-IPACCESSLOGDP: list 101 permit icmp 192.168.3.4
  (Serial1/1) -> 128.139.6.1  (0/0), 1 packet
```

Serial 1/1 is our prime suspect!

Link: http://www.cisco.com/warp/public/707/22.html

Cisco ACLs - 4

- From 12.0(6)S – TurboACLs – compiled ACLs – gives superior performance

Cisco CAR - 1

- CAR – Committed Access Rate

```
interface ATM1/1/0.21 point-to-point
rate-limit input access-group 180 96000 24000 32000
   conform-action continue exceed-action drop
 rate-limit input access-group 190 128000 30000
   30000 conform-action transmit exceed-action drop
!
access-list 180 deny   icmp 128.139.252.0 0.0.0.255
   any
access-list 180 permit icmp any any
access-list 190 deny   tcp any any established
access-list 190 permit tcp any any
```

b/w

Normal Burst in bytes

Max Burst in bytes

SYN Defender

No one really understands “burst” – best to read:
http://www.nanog.org/mtg-9811/ppt/witt/index.htm

Cisco CAR - 2

- sho int rate

```
router#sho int rate
ATM1/1/0.21
Input
    matches: access-group 180
      params:  96000 bps, 24000 limit, 32000 extended limit
      conformed 112068188 packets, 53953M bytes; action:
  transmit
      exceeded 8299587 packets, 10421M bytes; action: drop
      last packet: 1ms ago, current burst: 49119 bytes
      last cleared 2w6d ago, conformed 88000 bps, exceeded 20000
  bps
```

Dropped traffic

Null0 routing - 1

- Also known as blackholing
- Works only on destination addresses
- Cisco ASICs are optimized to work with null0
- Simple blackhole:

```
ip route 191.1.1.1 255.255.255.255 null0
```

  - Will appear in Netflow “null” list
  - Caveat: routers can forward faster than they can drop packets
  - Blackholes good packets with bad packets

Null routing - 2

- But ICMP Unreachables can overload CPU

```
interface null0
  no ip unreachables
```

Solution

- ICMP rate-limiting

```
ip icmp rate-limit unreachable [DF]<1-4294967295 millisecond>
```

Illegal addresses

Note: Many types of network attacks are dependent on spoofing the source IP address

Block inbound traffic sourced from your own address space:

```
access-list 110 deny ip 192.200.0.0 0.0.255.255 any
```

Block outbound traffic *not* sourced from your own address space:

```
access-list 111 permit ip 192.200.0.0 0.0.255.255 any
```

Block inbound traffic sourced from unroutable IP addresses:

```
access-list 110 deny ip 10.0.0.0    0.255.255.255 any
access-list 110 deny ip 172.16.0.0  0.15.255.255  any
access-list 110 deny ip 192.168.0.0 0.0.255.255   any
access-list 110 deny ip 127.0.0.0   0.255.255.255 any
access-list 110 deny ip 255.0.0.0   0.255.255.255 any
access-list 110 deny ip 1.0.0.0     0.255.255.255 any
... more [see next slide]...
```

RFC1918

Broadcast

Unallocated

Special IP Addresses

Addresses reserved for networks not connected to the Internet (RFC 1918)

```
10.0.0.0 – 10.255.255.255
172.16.0.0 - 172.31.255.255
192.168.0.0 – 192.168.255.255
```

Bogons: IP address as yet unallocated (some listed below)

1.0.0.0/8	58.0.0.0/8
2.0.0.0/8	59.0.0.0/8
27.0.0.0/8	127.0.0.0/8
31.0.0.0/8	169.254.0.0/16
36.0.0.0/8	197.0.0.0/8
41.0.0.0/8	223.0.0.0/8
49.0.0.0/8	224.0.0./8

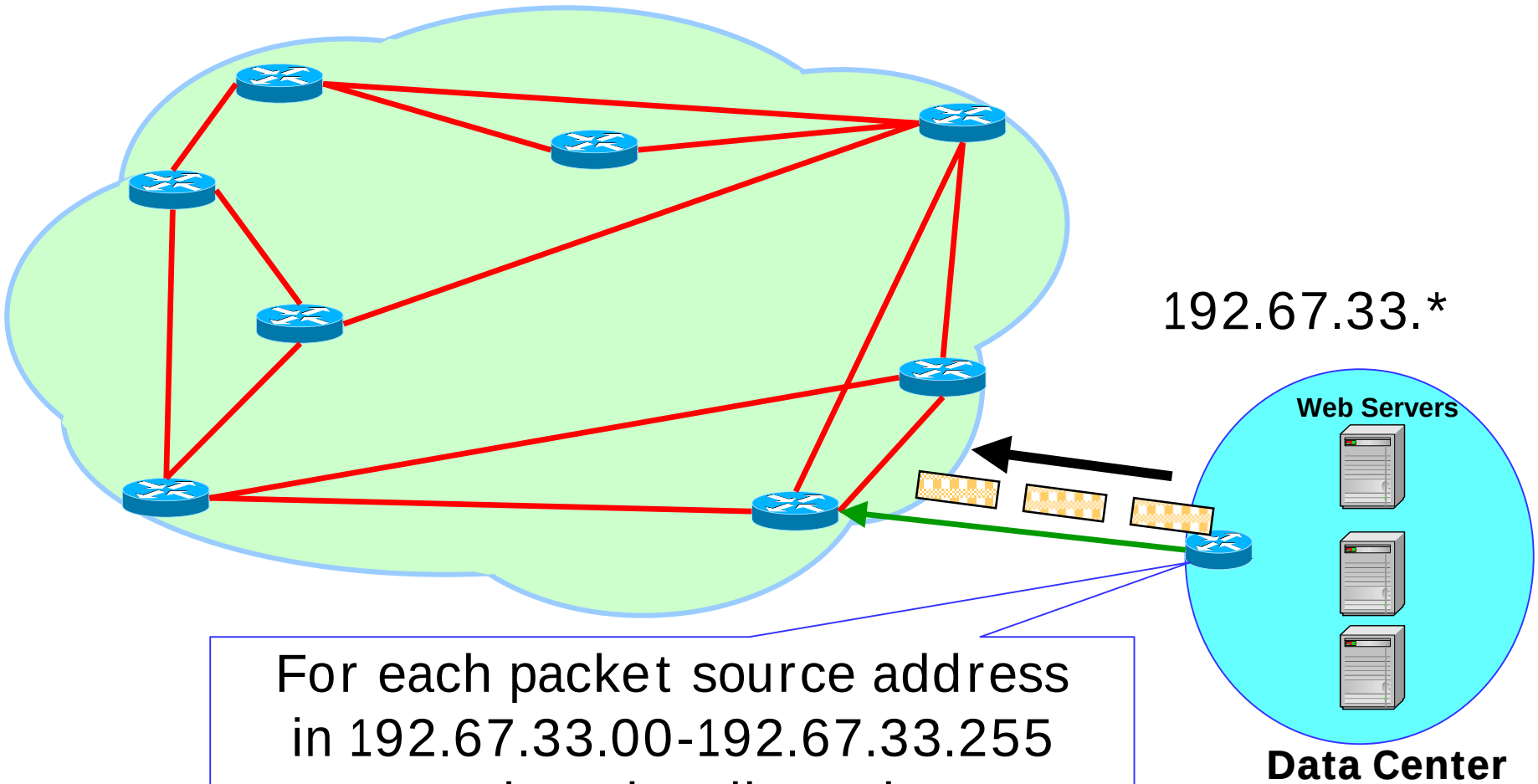
Complete list:
http://www.cymru.com/~robt/Docs/Articles/secure-ios-template.html
http://www.iana.org/assignments/ipv4-address-space

RFC2827: Network Ingress Filtering:Defeating Denial of Service Attacks which Employ IP Source Address Spoofing

Cisco – stopping Smurf

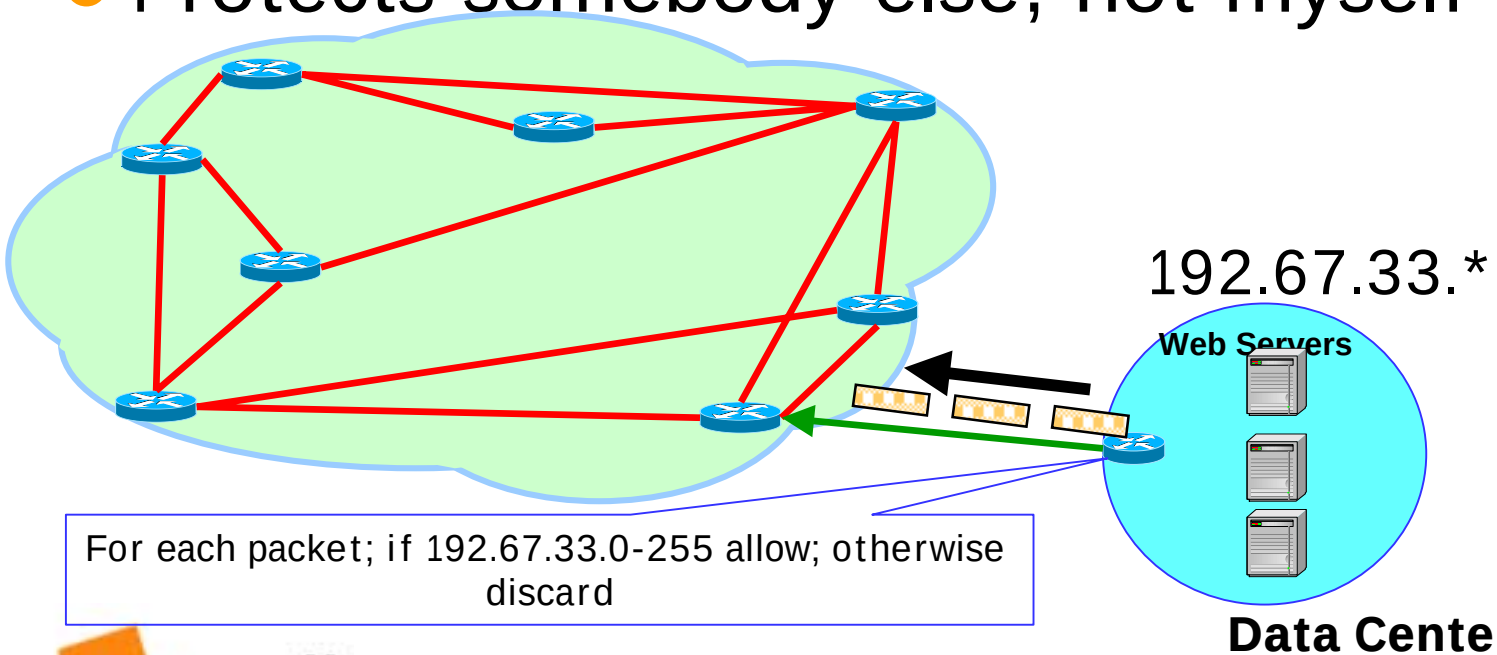
- **`no ip directed-broadcast`**
  - Translation of directed broadcast to physical MAC broadcasts is disabled
  - As of 12.0 this is the default

Ingress Filtering

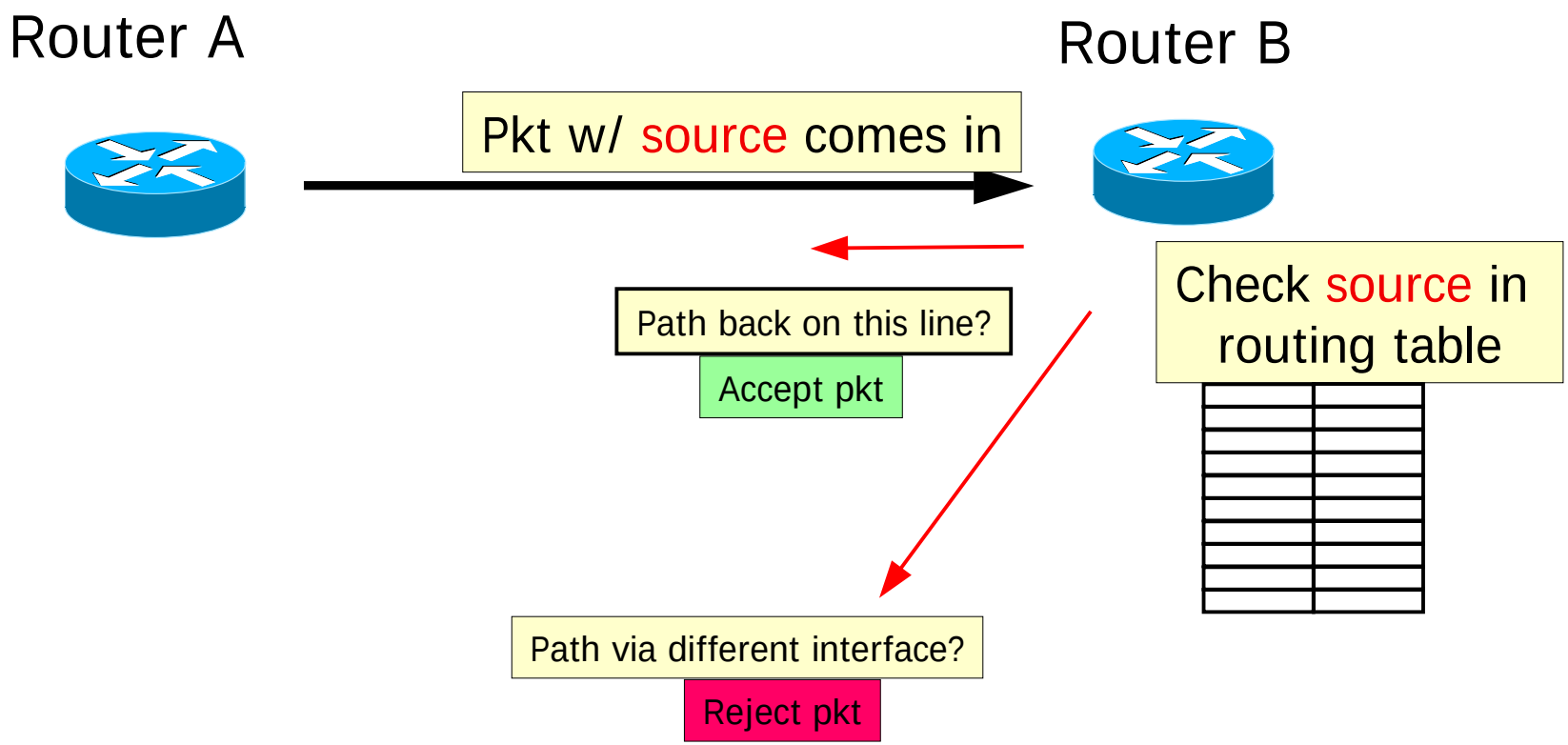


Ingress Filtering Cons

- Only anti-spoofing
- Does not stop internal spoofing
- Does not stop port spoofing
- Protects somebody else, not myself



Cisco uRPF



Does routing back to the source go through same interface ?

Cisco uRPF - 1

- Unicast Reverse Path Forwarding
  - Requires CEF
  - Available starting in 11.1(17)CC, and 12.0
    - Not available in 11.2 or 11.3 images
- Cisco interface command:

```
ip verify unicast rpf
```

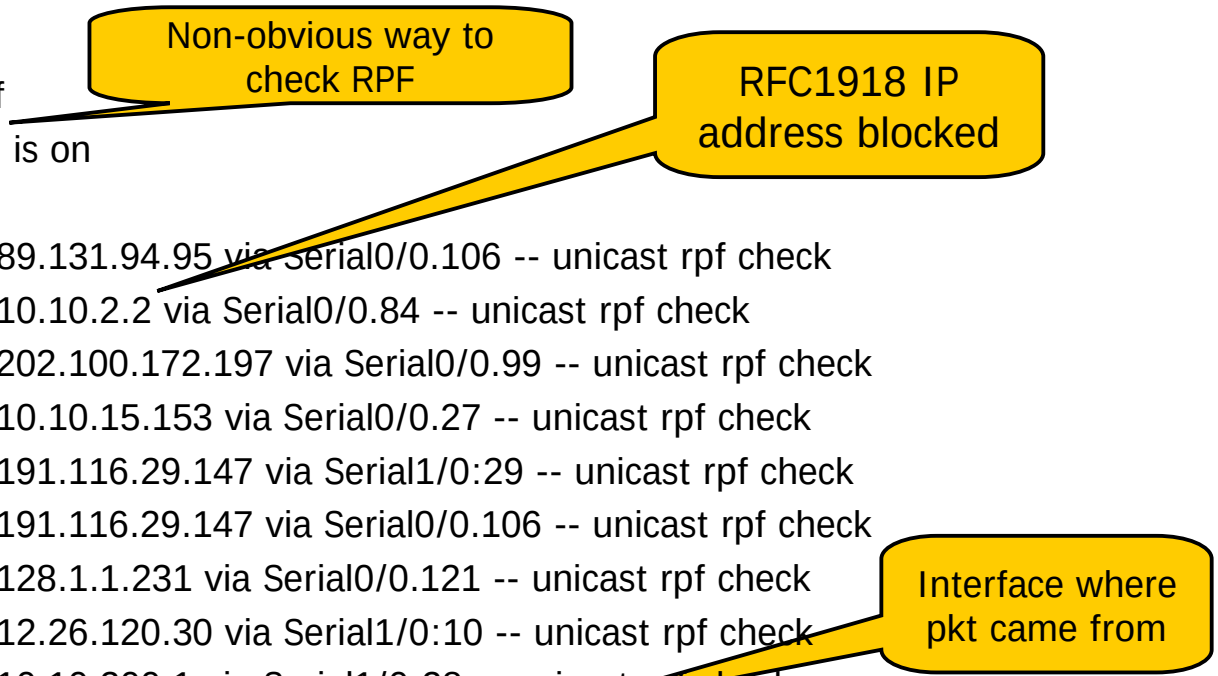
Cisco uRPF - 2

- Problem: Asymmetric routes
- Many ISPs may announce the same prefix - RPF checks only one of them
- Exceptions to uRPF checking:
  - 0.0.0.0 and 255.255.255.255
    - Needed for BOOTP and DHCP

Cisco uRPF -3

- Loose check:
  - **`ip verify source reachable via any`**
- Is there a way to route to the source using any interface?
  - NO - block
  - YES - allow
- Eliminates any spoofed IPs from the restricted prefixes list RFC 1918
- Eliminates any unallocated prefixes
- Does not completely solve the problem
  - To be used on edge – not backbone
  - Enhancements allow it to be deployed on ISP edge

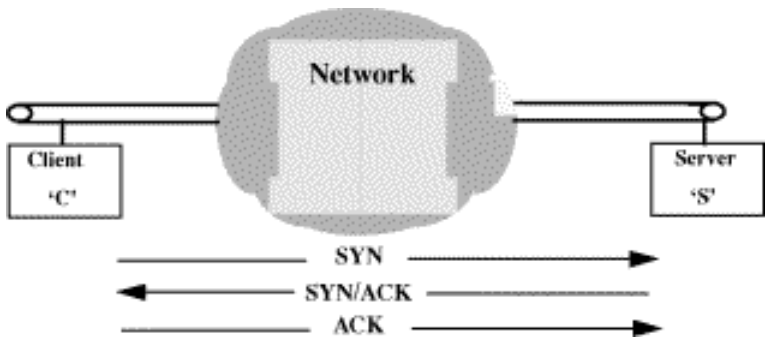
Cisco uRPF - 4




```
access-1#debug ip cef drops rpf
IP CEF drops for RPF debugging is on
access-1#term mon
18w0d: CEF-Drop: Packet from 89.131.94.95 via Serial0/0.106 -- unicast rpf check
18w0d: CEF-Drop: Packet from 10.10.2.2 via Serial0/0.84 -- unicast rpf check
18w0d: CEF-Drop: Packet from 202.100.172.197 via Serial0/0.99 -- unicast rpf check
18w0d: CEF-Drop: Packet from 10.10.15.153 via Serial0/0.27 -- unicast rpf check
18w0d: CEF-Drop: Packet from 191.116.29.147 via Serial1/0:29 -- unicast rpf check
18w0d: CEF-Drop: Packet from 191.116.29.147 via Serial0/0.106 -- unicast rpf check
18w0d: CEF-Drop: Packet from 128.1.1.231 via Serial0/0.121 -- unicast rpf check
18w0d: CEF-Drop: Packet from 12.26.120.30 via Serial1/0:10 -- unicast rpf check
18w0d: CEF-Drop: Packet from 10.10.200.1 via Serial1/0:28 -- unicast rpf check
18w0d: CEF-Drop: Packet from 191.116.29.147 via Serial1/0:10 -- unicast rpf check
18w0d: CEF-Drop: Packet from 200.73.138.16 via Serial0/0.99 -- unicast rpf check
18w0d: CEF-Drop: Packet from 201.136.29.114 via Serial0/0.27 -- unicast rpf check
18w0d: CEF-Drop: Packet from 191.116.29.147 via Serial1/0:24 -- unicast rpf check
18w0d: CEF-Drop: Packet from 201.228.107.191 via Serial0/0.18 -- unicast rpf check
18w0d: CEF-Drop: Packet from 60.150.47.35 via Serial0/0.106 -- unicast rpf check
18w0d: CEF-Drop: Packet from 201.52.115.129 via Serial1/0:10 -- unicast rpf check
```

Cisco TCP Intercept - 1

- Method used to stop SYN flooding
- Gets in the middle of the TCP 3-way handshake



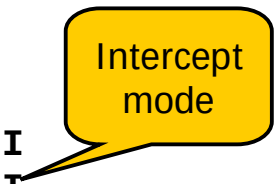
Cisco TCP Intercept - 2

```
! Enable TCP Intercept to protect against SYN flooding.
ip tcp intercept list 120
! Watch the "flow" for only 60 seconds
ip tcp intercept connection-timeout 60
! Keep half-open sockets only 10 seconds.
ip tcp intercept watch-timeout 10
! Set the low water mark to 1500 active opens per minute.
ip tcp intercept one-minute low 1500
! Set the high water mark to 6000 active opens per minute.
ip tcp intercept one-minute high 6000
! Configure an ACL for TCP Intercept.   Protect only a /24
access-list 120 permit tcp any 192.111.1.0 0.0.0.255
```

Cisco TCP Intercept - 3

- Monitoring
  - **show tcp intercept connections**

```
Incomplete:
Client Server State Create Timeout Mode
172.19.160.17:58190 10.1.1.30:23 SYNRCVD 00:00:09 00:00:05 I
172.19.160.17:57934 10.1.1.30:23 SYNRCVD 00:00:09 00:00:05 I
Established:
Client Server State Create Timeout Mode
171.69.232.23:1045 10.1.1.30:23 ESTAB 00:00:08 23:59:54 I
```



  - **show tcp intercept statistics**

```
intercepting new connections using access-list 120
543 incomplete, 16 established connections (total 3)
1 minute connection request rate 24 requests/sec
```

Cisco NBAR

- Network-Based Application Recognition
  - Only available on 12.1(5)T and later
- Can be done via 3 methods:
  - ACLs
  - Policy Based Routing
  - Policing policy
- Many restrictions on use
  - Not fragmented packets
  - Not on tunnels
  - Not on VLANs
  - Only first 400 bytes
  - Many more…

Cisco NBAR

```
class-map match-any http-attacks
  match protocol http url "*.ida*"
  match protocol http url "*cmd.exe*"
  match protocol http url "*root.exe*"
  match protocol http url "*readme.eml*"
  match protocol http url "*httpdodbc.dll*"
  match protocol http url "*Admin.dll*"
!
policy-map Trash-it
  class http-attacks
    set ip dscp 1
!
Interface n/n
 service-policy input Trash-it
 ip policy route-map null_policy_route
!
access-list 104 permit ip any any dscp 1
!
route-map null_policy_route permit 10
 match ip address 104
 set interface Null0
```

Patterns to match on

Mark the pkt w/ something unique

Anything that matches ACL 104 – throw away

Juniper

- Internet Processor II - Filtering, sampling, and rate limiting capabilities (same as Cisco but faster) (JUNOS 4.4)
  - Firewall filtering done in hardware (from 3.2)
- Independent Processor – no effect on the router performances
- Blocks legitimate traffic as well

Juniper – Stopping Smurf

- M-series routers rate limit ICMP echo requests directed to the router so that no more than 1,000 per second reach the Routing Engine
- M-series routers do not support directed broadcast
- http://www.juniper.net/techcenter/app_note/350001.html

Why Routers can't Protect

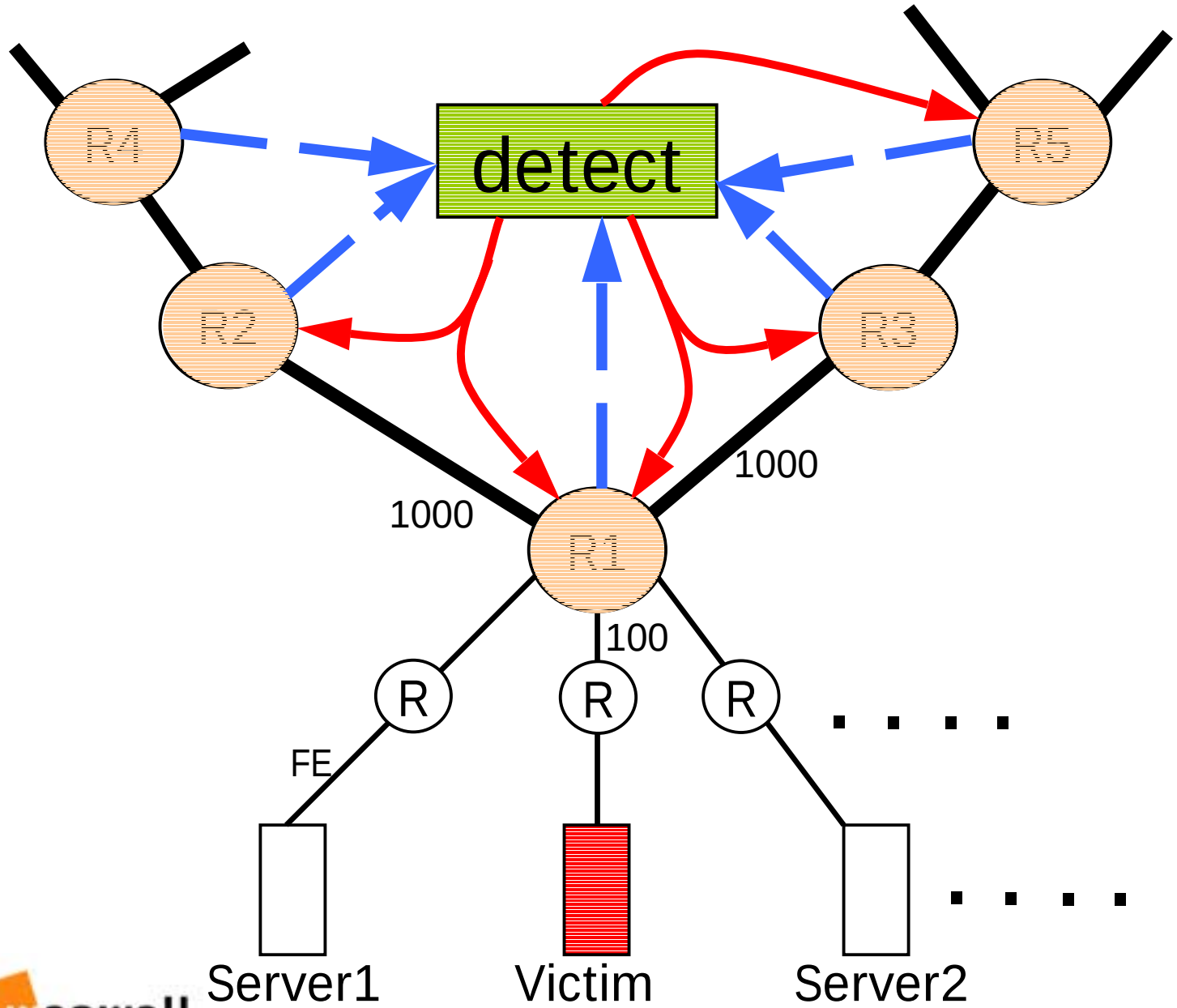
- ACL and CAR
  - Throws away good with the bad
  - Performance degradation
    - Central CPU being hit
    - During DDoS router non-responsive
  - Requires dynamic reconfiguration during attack
- Weak in defending the following attacks
  - Random everything (Targa)
  - Incomplete connections (Naphta)
  - Spoofed SYN floods
  - DNS attacks
  - Client attacks (http)
  - Zombie behind a proxy

10. DDOS companies

Three major categories

1. Detection boxes + Router filtering
2. On the critical path detection and filtering box
   - Special device
   - Firewalls, Load balancers, Switches
3. Detection & Diversion

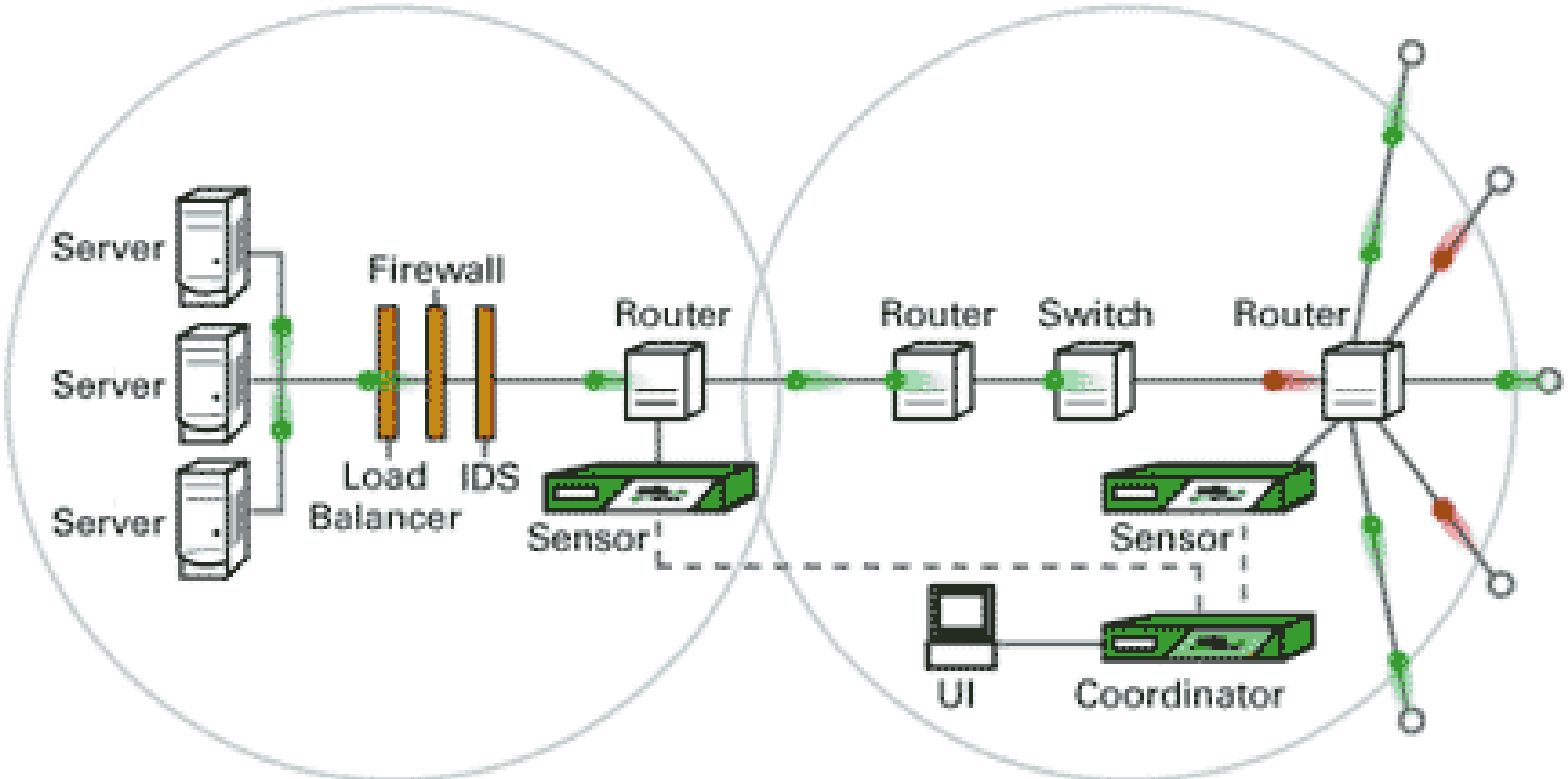
Detection boxes + Router filtering



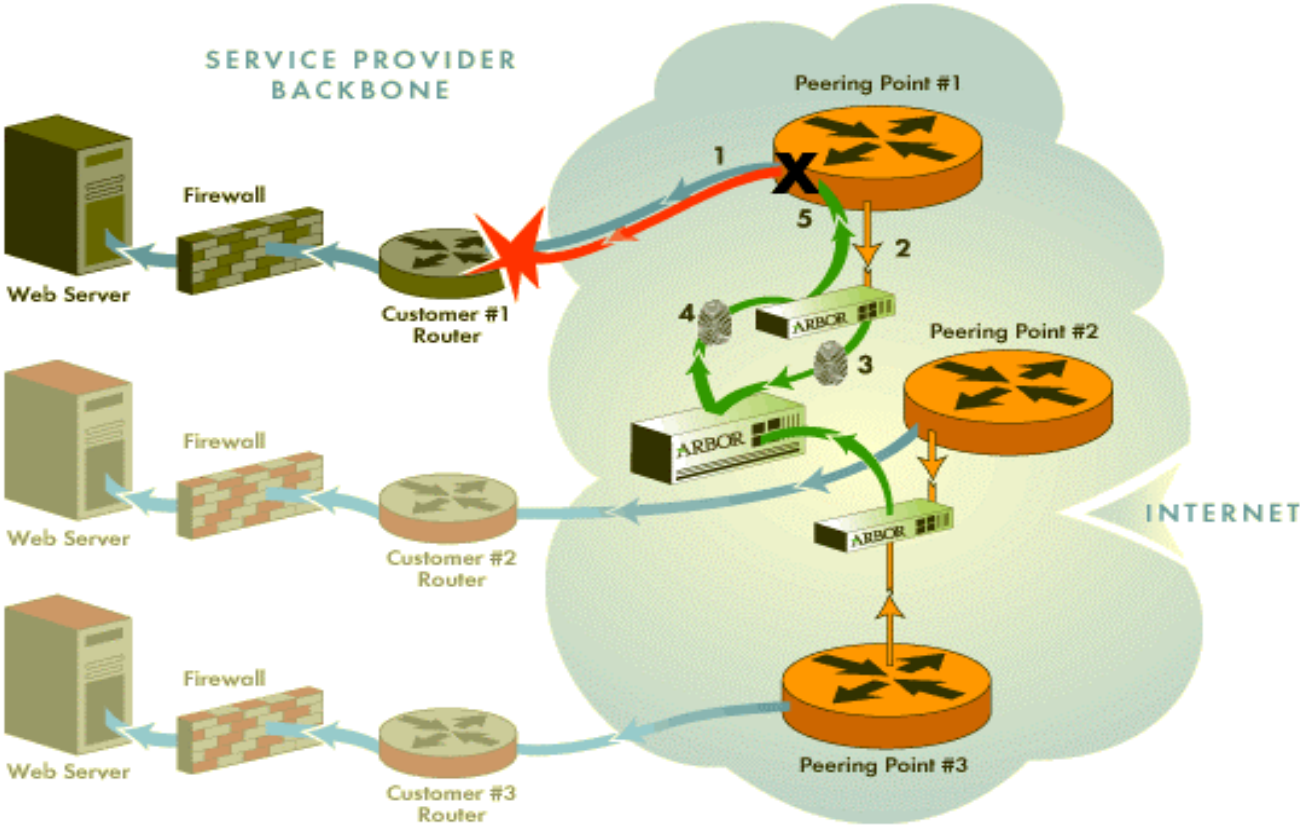
Asta Networks

- **Vantage System**
  - Runs on hardened Linux system
- Netflow / Optical splitters
- Signature and Anomaly detection
- Recommends ACL filters

Asta Networks



Arbor Networks



1.Traffic enters the Service Provider network. 2.Monitor: Peakflow DoS Collectors analyze traffic for anomalies without disrupting traffic flow to routers. 3.Detect: Peakflow DoS Collectors create and forward unique anomaly fingerprints to Peakflow DoS Controllers. 4.Trace: Peakflow DoS Controllers then quickly trace the attack to its source. 5.Filter: Peakflow DoS Controller recommends filters, which the network engineer can implement to stop the attack before it brings down key routers, firewalls and/or the entire network.

Arbor Networks

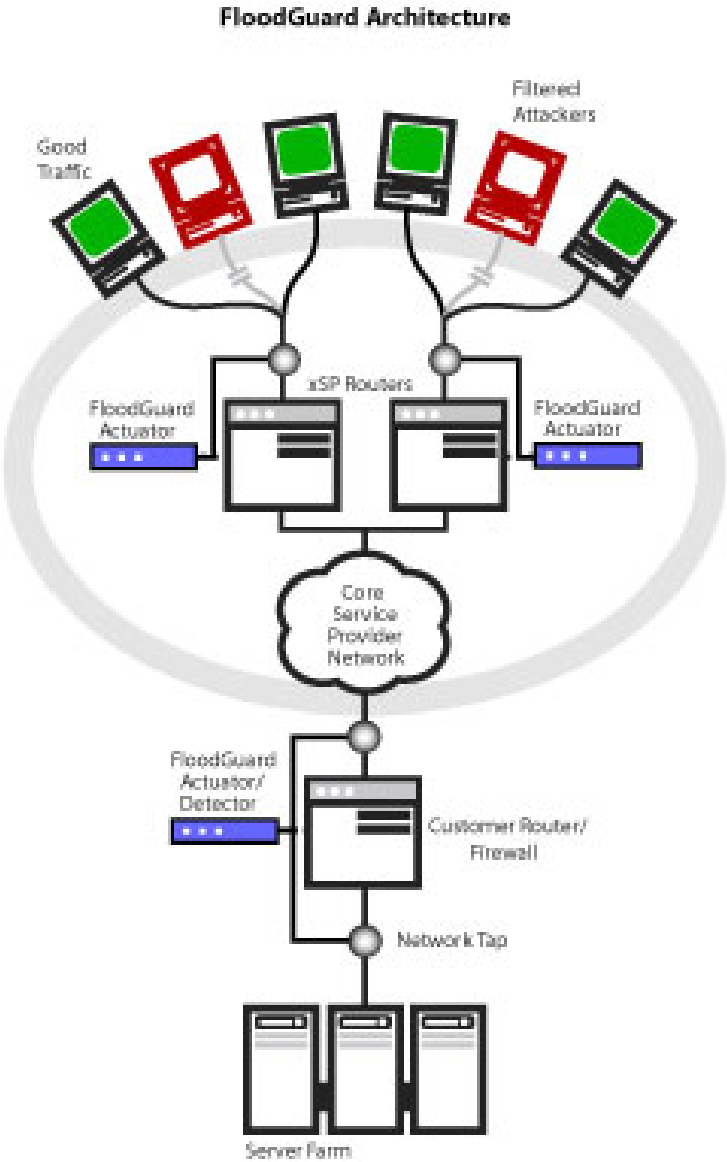
- **Peakflow**
  - Hardened OpenBSD system
- Netflow
- Builds suggested ACLs and filters for placement on customer router
  - Requires customer to view filter before applying

Reactive Networks

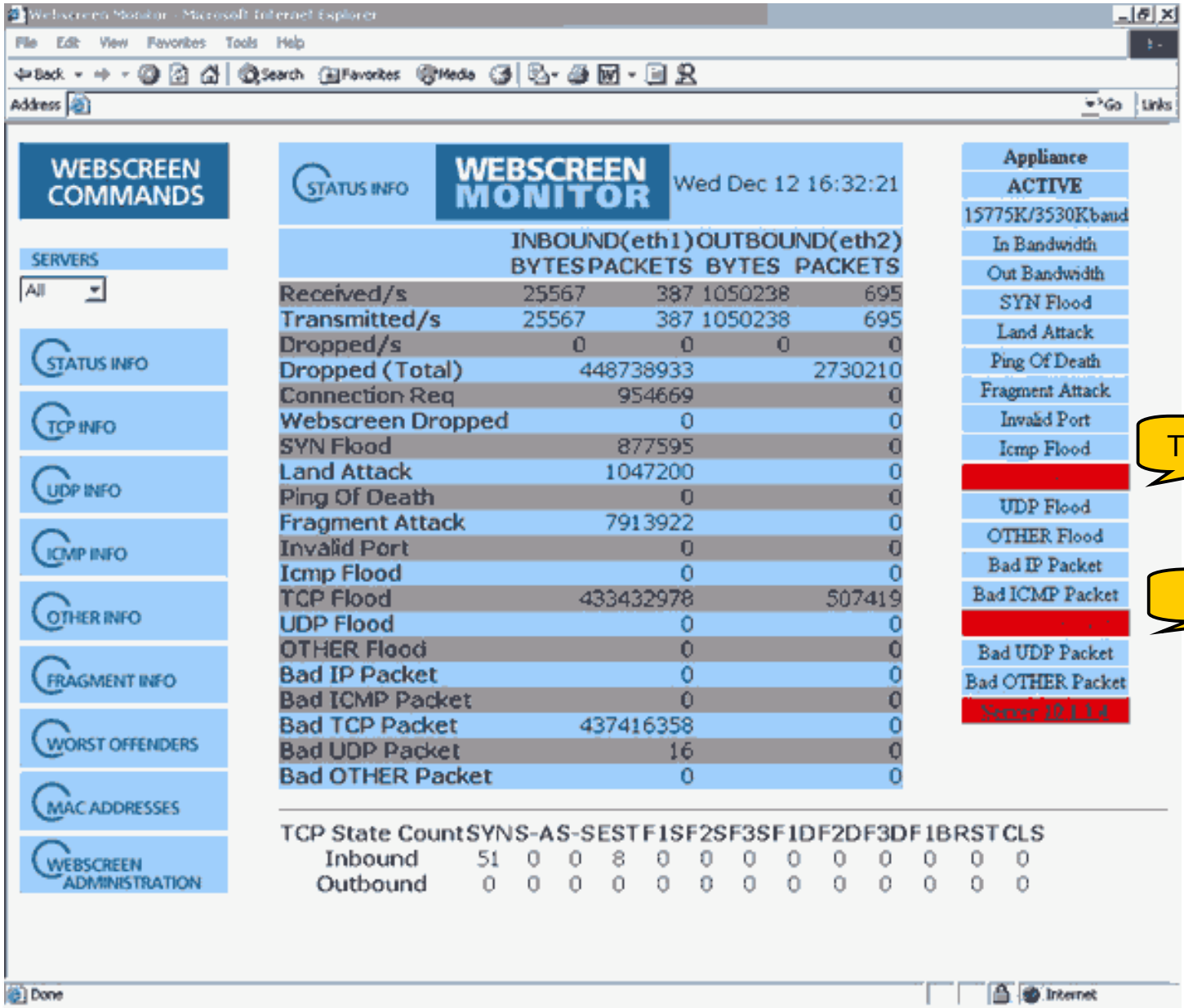
- **Floodguard**
  - 1U box
  - Linux based
- Modifies upstream Cisco ACLs
  - Doesn't support Juniper routers
- Spoofs RSTs to close incoming connections
  - Mitigates valid and attack traffic on an equal basis

[FIREPROOFING- NetworkComputing 01]

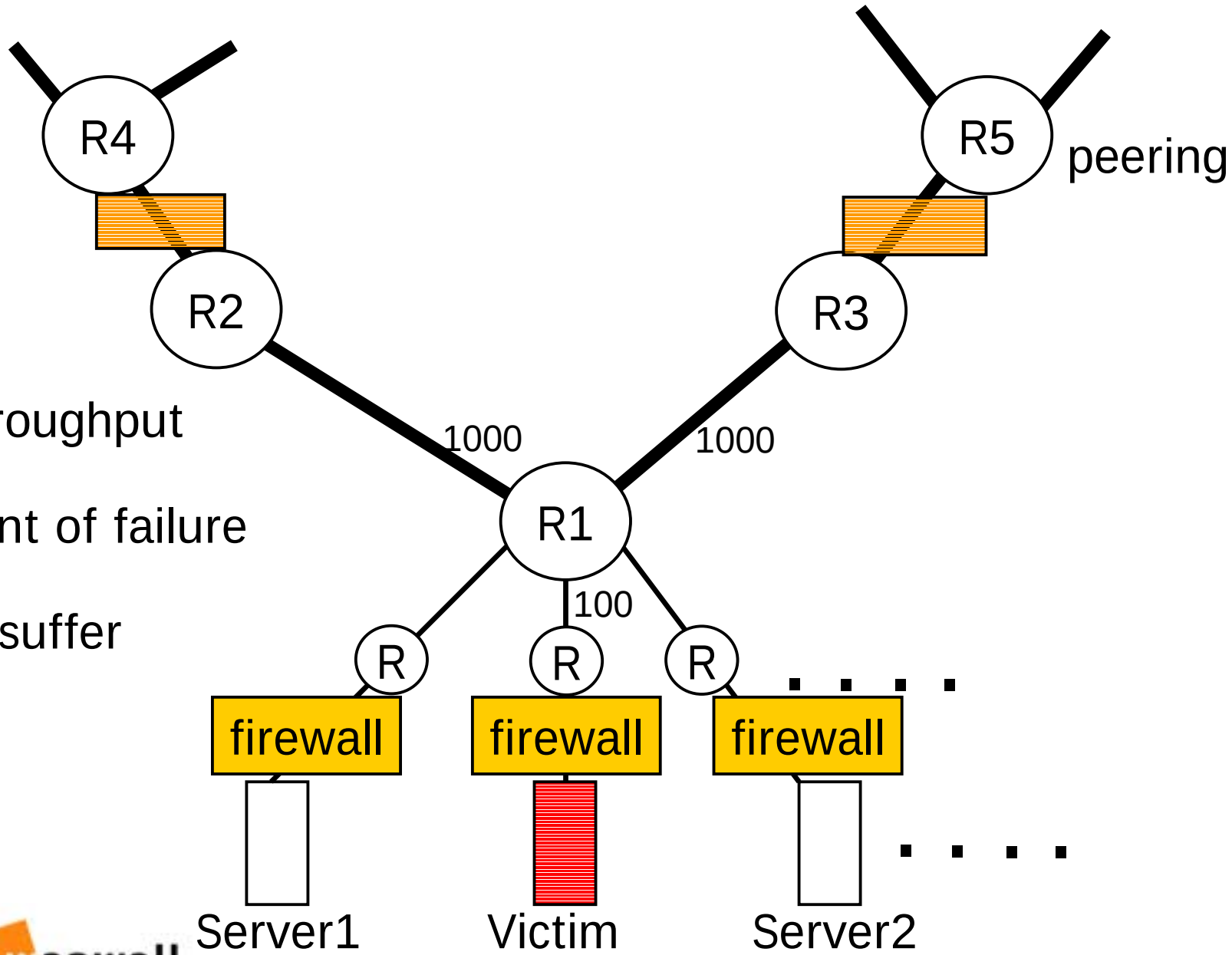
Reactive Networks



Webscreen

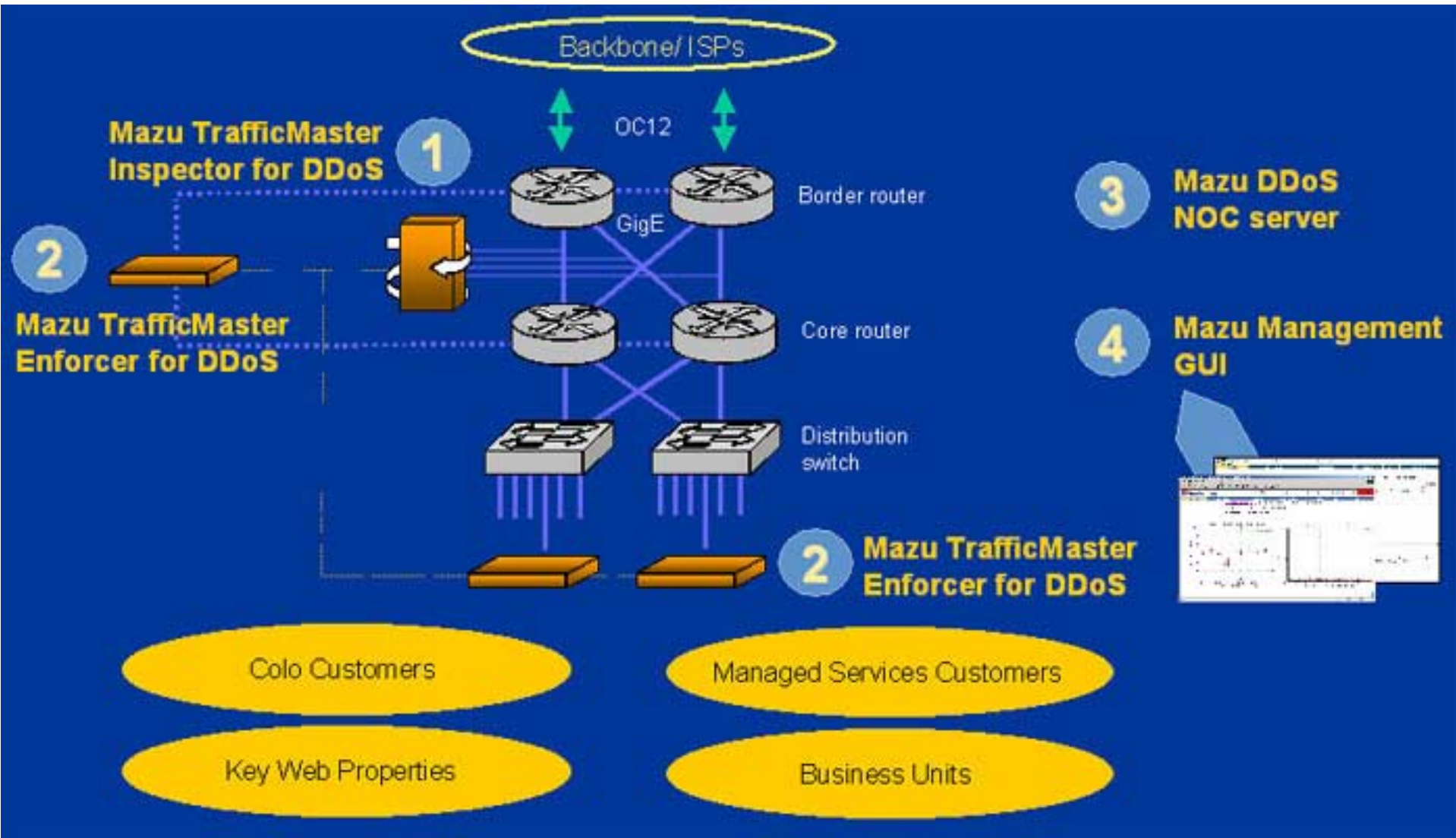

WEBSCREEN COMMANDS
SERVERS
All
STATUS INFO
TCP INFO
UDP INFO
ICMP INFO
OTHER INFO
FRAGMENT INFO
WORST OFFENDERS
MAC ADDRESSES
WEBSCREEN ADMINISTRATION
STATUS INFO
WEBSCREEN MONITOR
Wed Dec 12 16:32:21
INBOUND(eth1) OUTBOUND(eth2)
BYTES PACKETS BYTES PACKETS
Received/s 25567 387 1050238 695
Transmitted/s 25567 387 1050238 695
Dropped/s 0 0 0 0
Dropped (Total) 448738933 2730210
Connection Req 954669 0
Webscreen Dropped 0 0
SYN Flood 877595 0
Land Attack 1047200 0
Ping Of Death 0 0
Fragment Attack 7913922 0
Invalid Port 0 0
Icmp Flood 0 0
TCP Flood 433432978 507419
UDP Flood 0 0
OTHER Flood 0 0
Bad IP Packet 0 0
Bad ICMP Packet 0 0
Bad TCP Packet 437416358 0
Bad UDP Packet 16 0
Bad OTHER Packet 0 0
TCP State Count SYNS A S ESTF1SF2SF3SF1DF2DF3DF1BRST CLS
Inbound 51 0 0 8 0 0 0 0 0 0 0 0 0
Outbound 0 0 0 0 0 0 0 0 0 0 0 0 0
Appliance
ACTIVE
15775K/3530Kbaud
In Bandwidth
Out Bandwidth
SYN Flood
Land Attack
Ping Of Death
Fragment Attack
Invalid Port
Icmp Flood
UDP Flood
OTHER Flood
Bad IP Packet
Bad ICMP Packet
Bad UDP Packet
Bad OTHER Packet
TCP Flood
Bad TCP Pkt

Inline



- Throughput
- Point of failure
- All suffer

Mazu Networks



Mazu Networks

- **Traffic Master Inspector & Enforcer**
  - Runs on hardened Linux on IBM Netfinity box
    - 3U device
- Real time graphs
- Works by detecting anomalies
  - Suggests filters
  - Needs to be ok'ed by NOC to turn on filter
  - Some filters too complex
    - Filters cannot be edited before applying
- Has additional SYN-Queue technology
  - Sends RST to the server
  - Makes no distinction between good and bad SYNs

[FIREPROOFING- NetworkComputing 01]

Radware

- **Fireproof + Application switch**
  - **1U device**
- Requires additional SynApp technology
- FastEthernet
  - Requires separate Application Switch for GigaE
- Fireproof – Firewall load balancer
- Lacks detailed reporting tools
- Problematic attacks: NAPHTA, Targa

Source: [FIREPROOFING- NetworkComputing 01]

Foundry Networks

- **ServerIron 400**
  - 5U product
- Very Cisco-like CLI
- GUI with many menus & submenus
- SYN flood protection via a virtual Web server
- Smurf protection doesn't stop ICMP echo floods
  - Rate-limiting on sending out ICMP echo
- SYN protection – blocks **ALL** SYNs once a threshhold is hit for a specified period of time

Source: [FIREPROOFING- NetworkComputing 01]

Captus Networks

- **CaptIO G2**
  - Internet appliance
- TLIDS (Traffic Limiting Intrusion Detection System)
- Lacks reporting
  - No graphs or traffic breakouts
- Doesn't handle spoofed SYN attacks
- Doesn't handle NAPHTA attacks
- Does handle some Targa attacks
  - UDP and ICMP

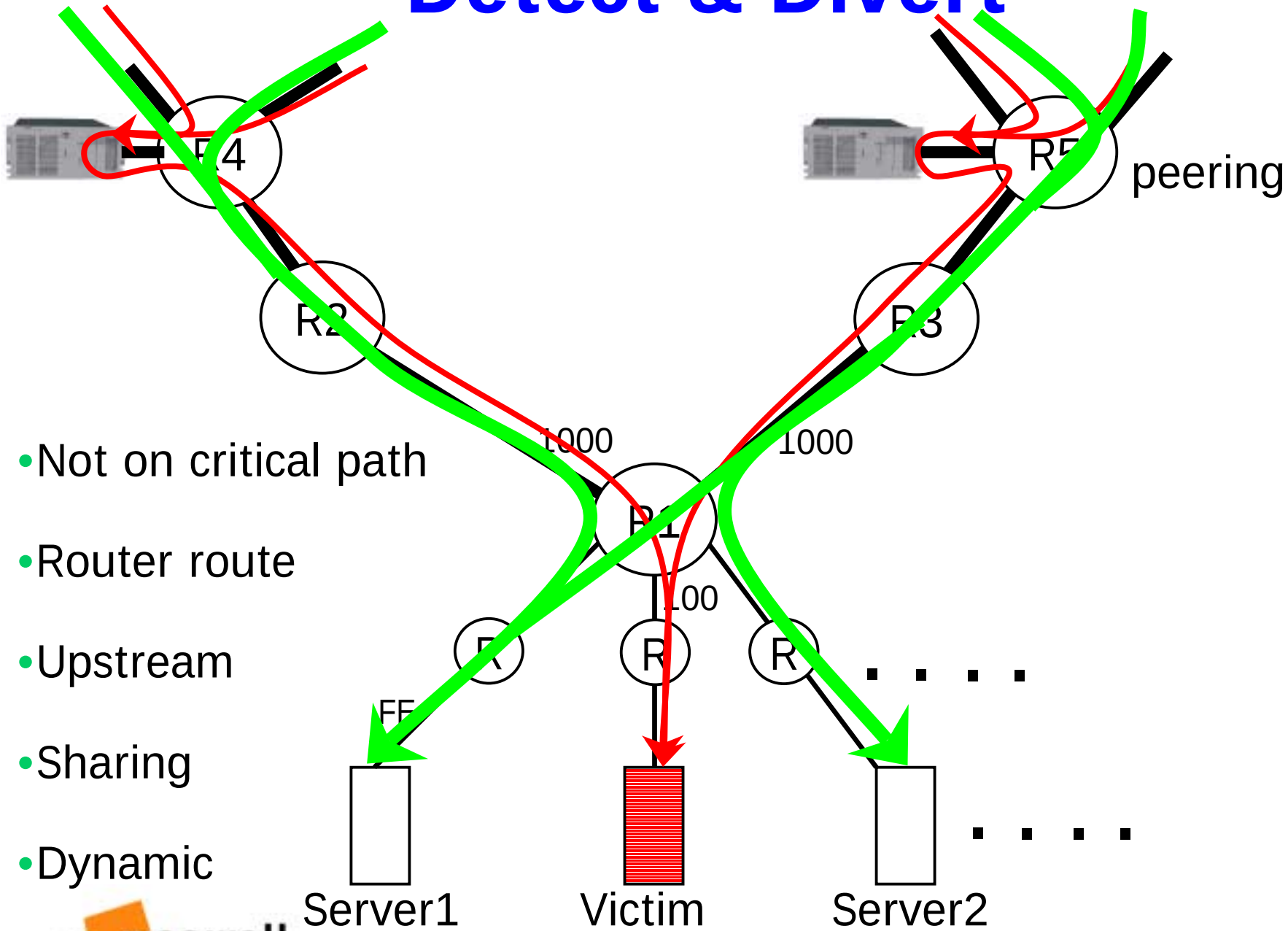
Source: [FIREPROOFING- NetworkComputing 01]

TopLayer Networks

- **Appswitch 3500**
  - GigaE support
  - 2U device
- Handles: land, Smurf, fraggle, UDP, SYN, fragments, source routing
- Doesn't stop floods of small ICMP packets
- Handles 256,000 simultaneous flows
- IP layer proxy

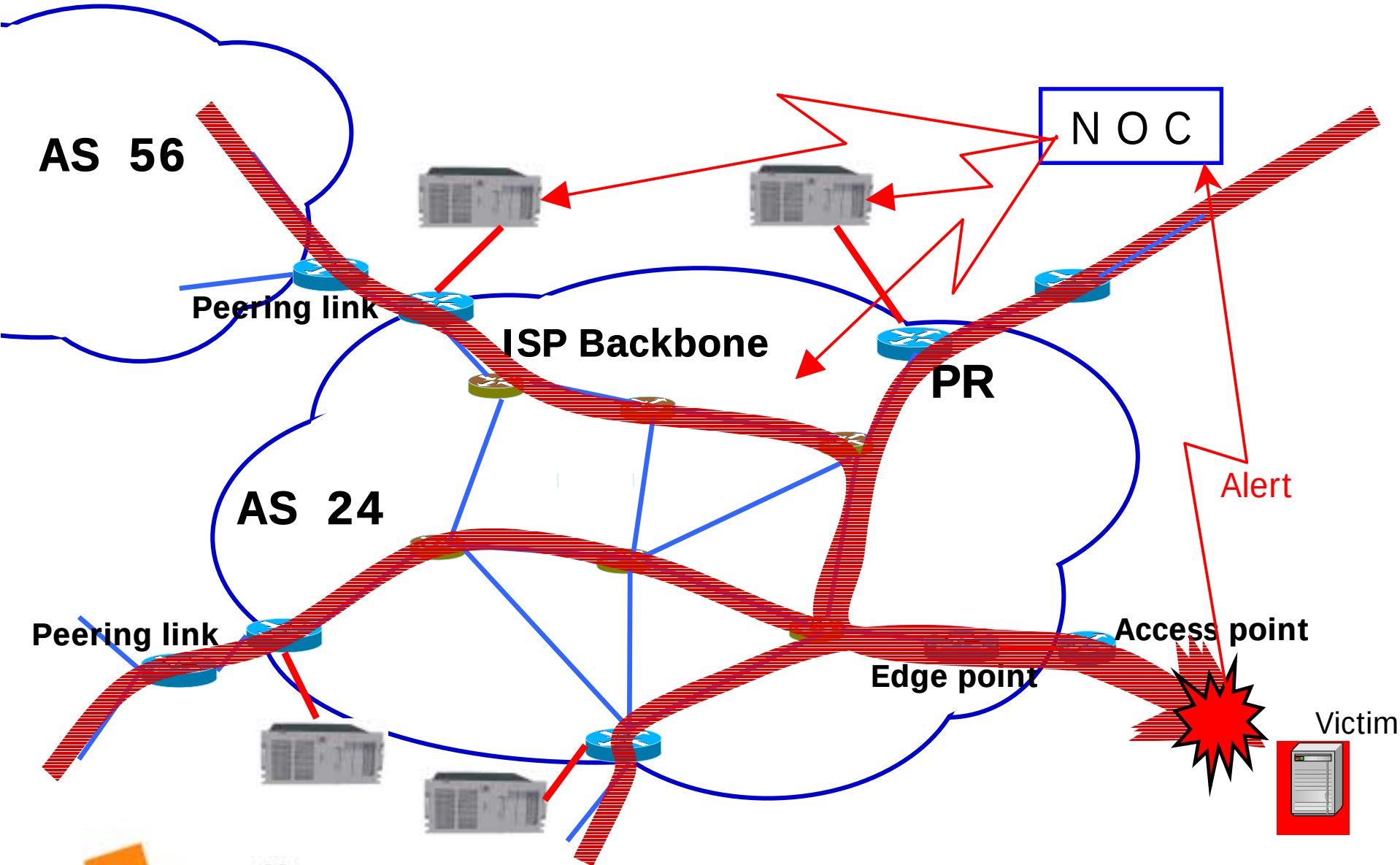
Source: [FIREPROOFING- NetworkComputing 01]

Detect & Divert

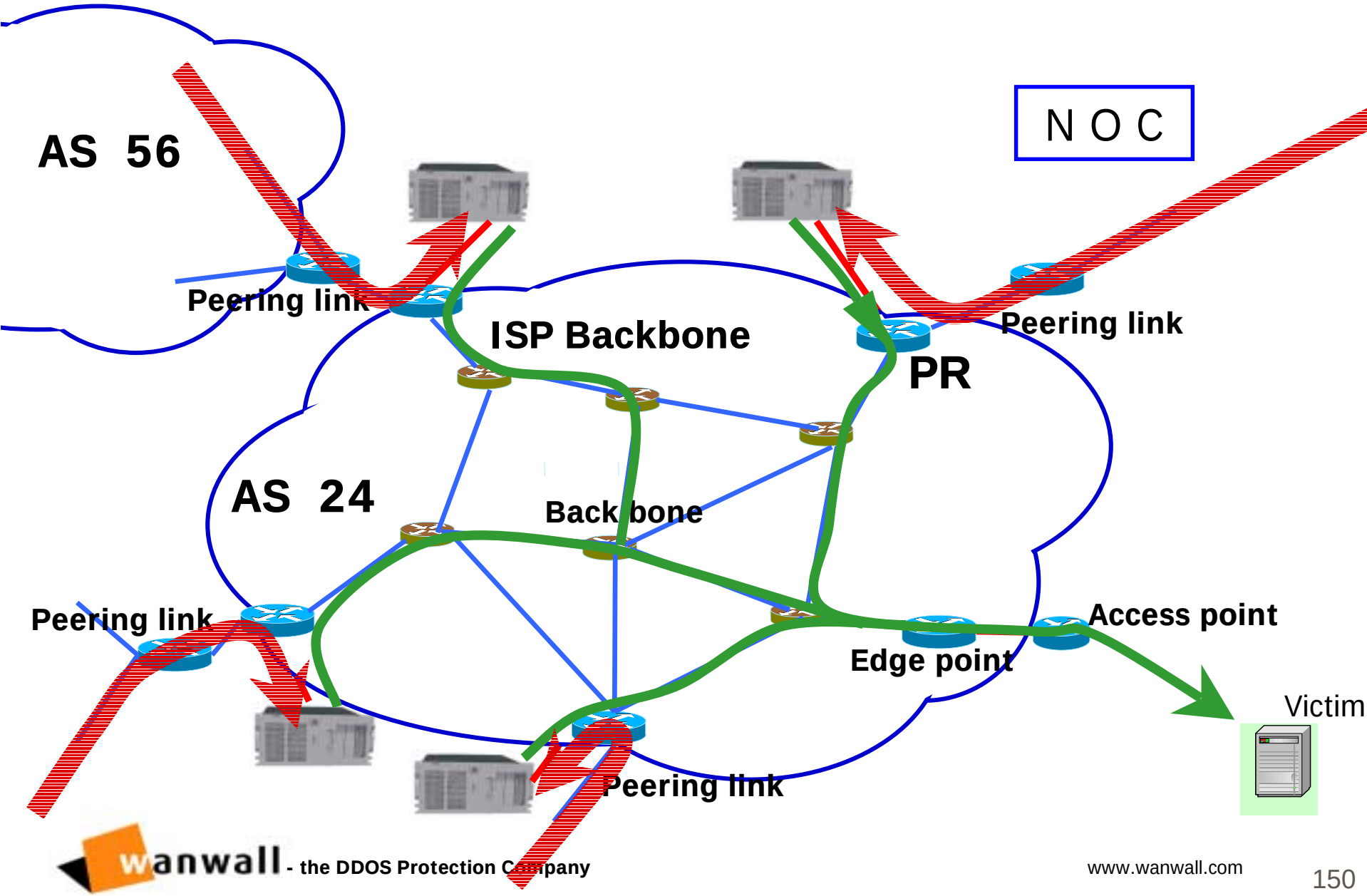


- Not on critical path
- Router route
- Upstream
- Sharing
- Dynamic

Basic Scheme

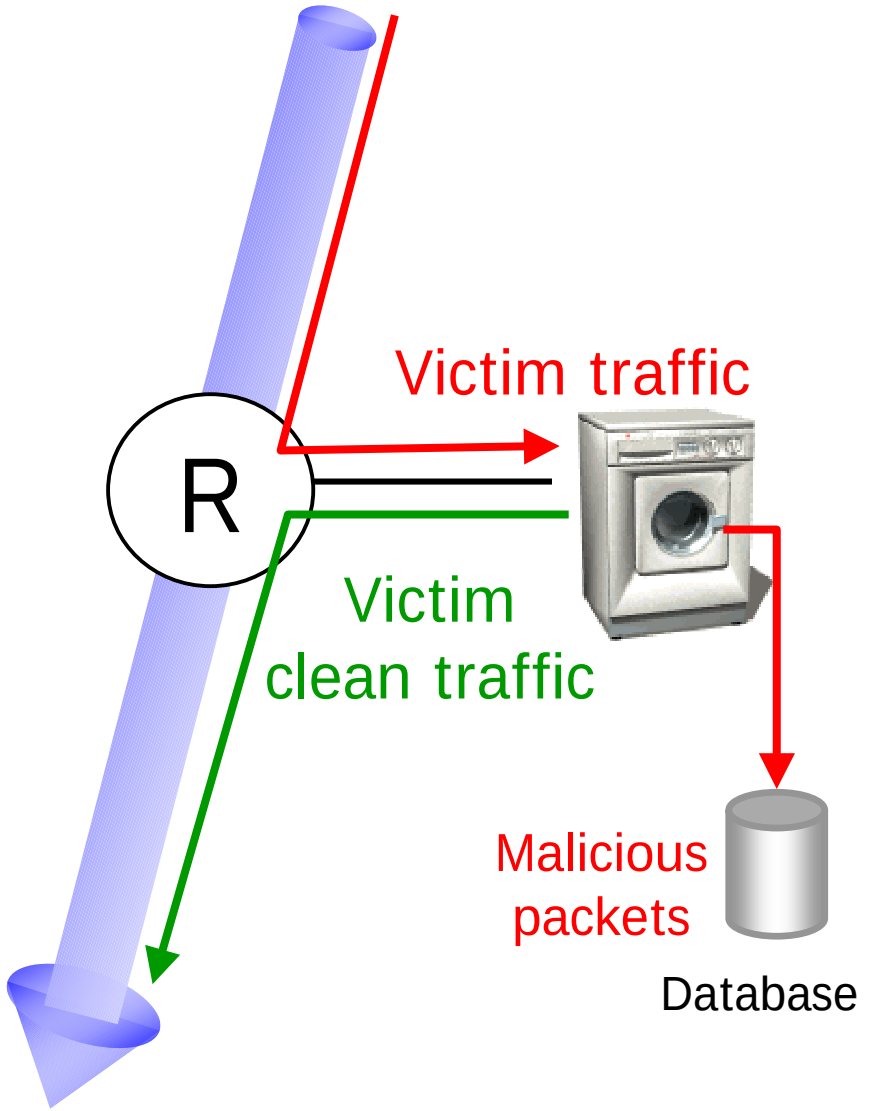


Basic Scheme

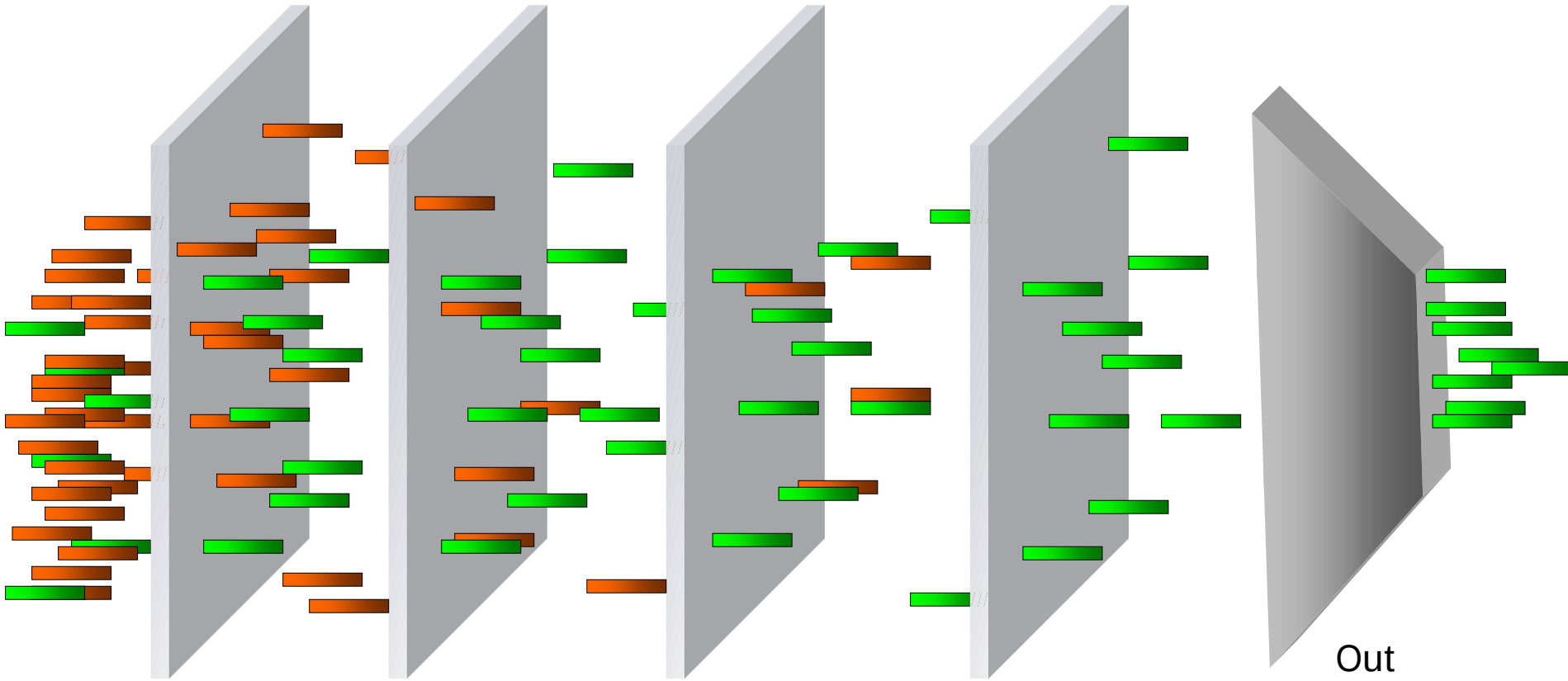


Wanwall

1. Detect
2. Divert victim's traffic
3. Sieve
4. Legitimate traffic continues on its route



Wanwall


Packet filtering
Anti spoofing
Learning & Statistical analysis
HTTP Analysis & Authentication
Out

Others

- CS3
- Entercept
- Tripwire
- Oneka
- Recourse
- Shai
- Netscreen

Summary Links

- http://staff.washington.edu/dittrich/misc/ddos/
- http://www.linuxsecurity.com/resource_files/intrusion_detection/ddos-faq.html
- http://www.networkcomputing.com/1201/1201f1c1.html
- http://www.sans.org/dosstep/index.htm
- http://downloads.securityfocus.com/library/sn_ddos.doc

Dank u wel, voor uw aandacht

Comments: {afek,hank}@wanwall.com