

CSE 123b

Communications Software

Spring 2003

Lecture 16: Network Security II

Stefan Savage

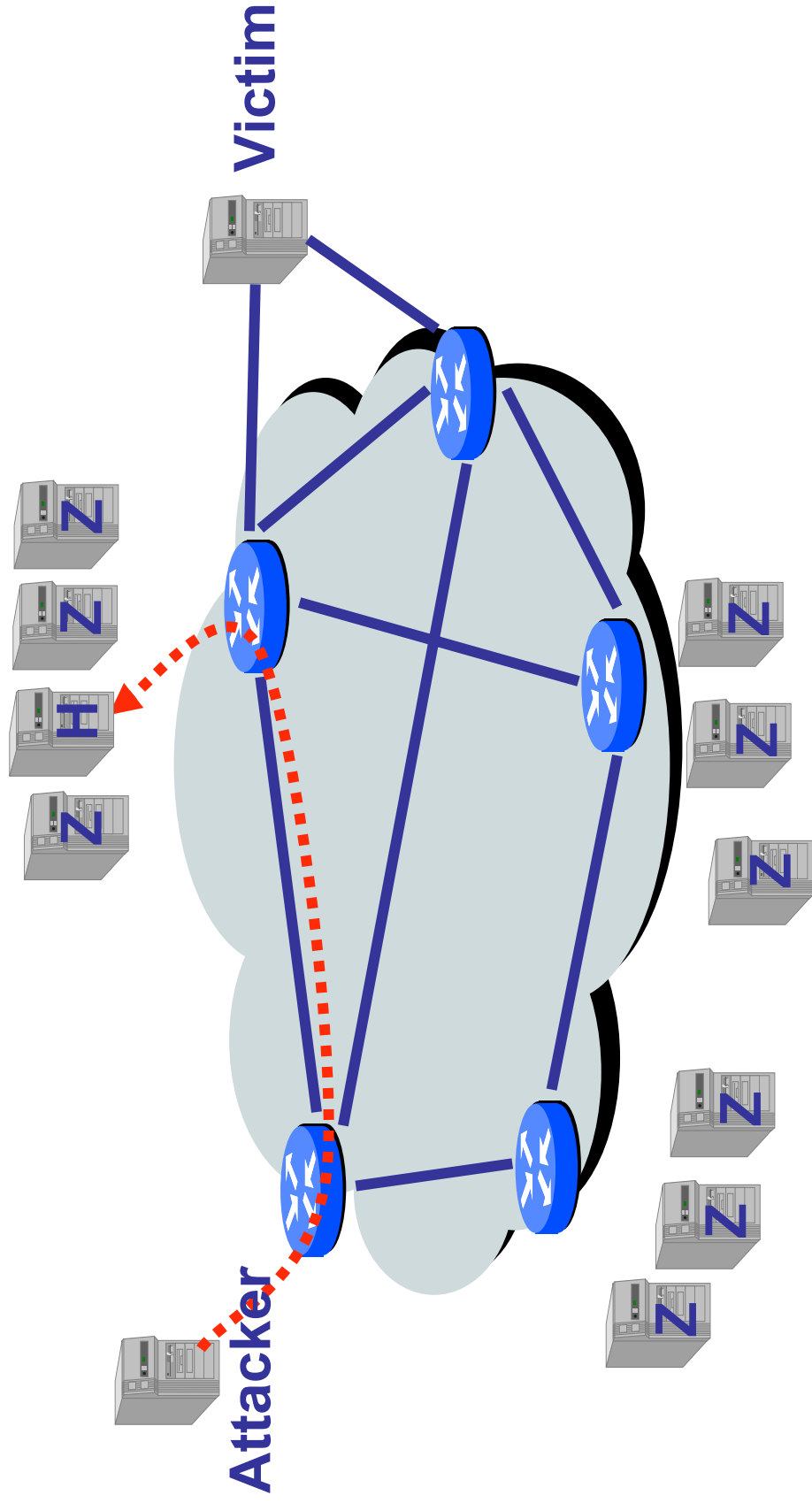
How do DoS attacks work?

- Denial-of-service attacks
 - ◆ **Logic:** exploit bugs to cause crash
 - » e.g. Ping-of-Death, Land
 - ◆ **Flooding:** overwhelm with spurious requests
 - » e.g. SYN flood, Smurf
- **Distributed denial-of-service attacks**
 - ◆ Flooding attack from multiple machines
 - ◆ More potent & harder to defend against

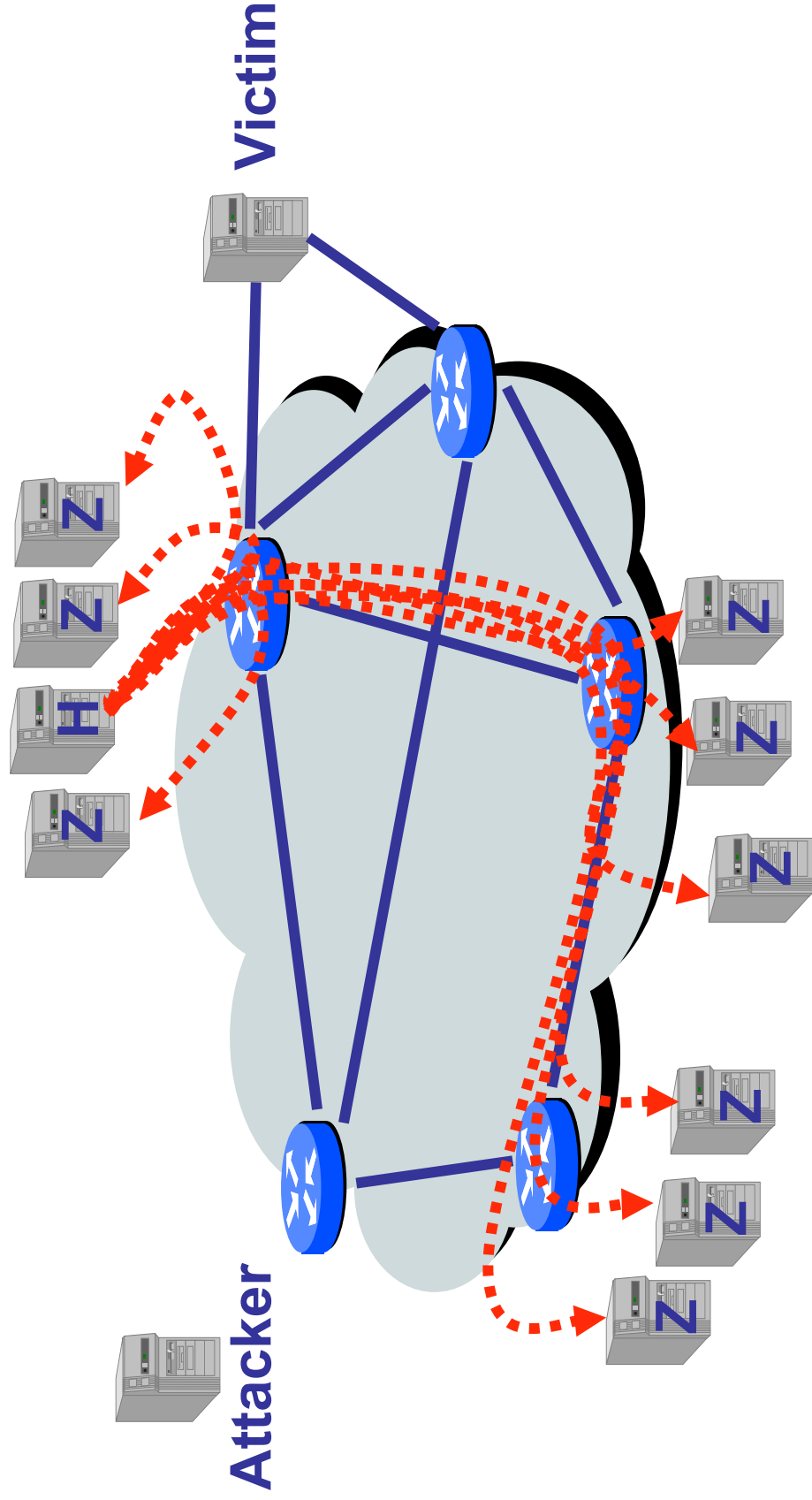
Step 1: Attacker infiltrates machines

- Scan machines via Internet
- Exploit known bugs & vulnerabilities
- Install backdoor software
 - ◆ *Zombie* software (for attacking target)
 - ◆ Handler software (for controlling zombies)
- Cover tracks (e.g. **rootkit**)
- Repeat... (**highly automated**)

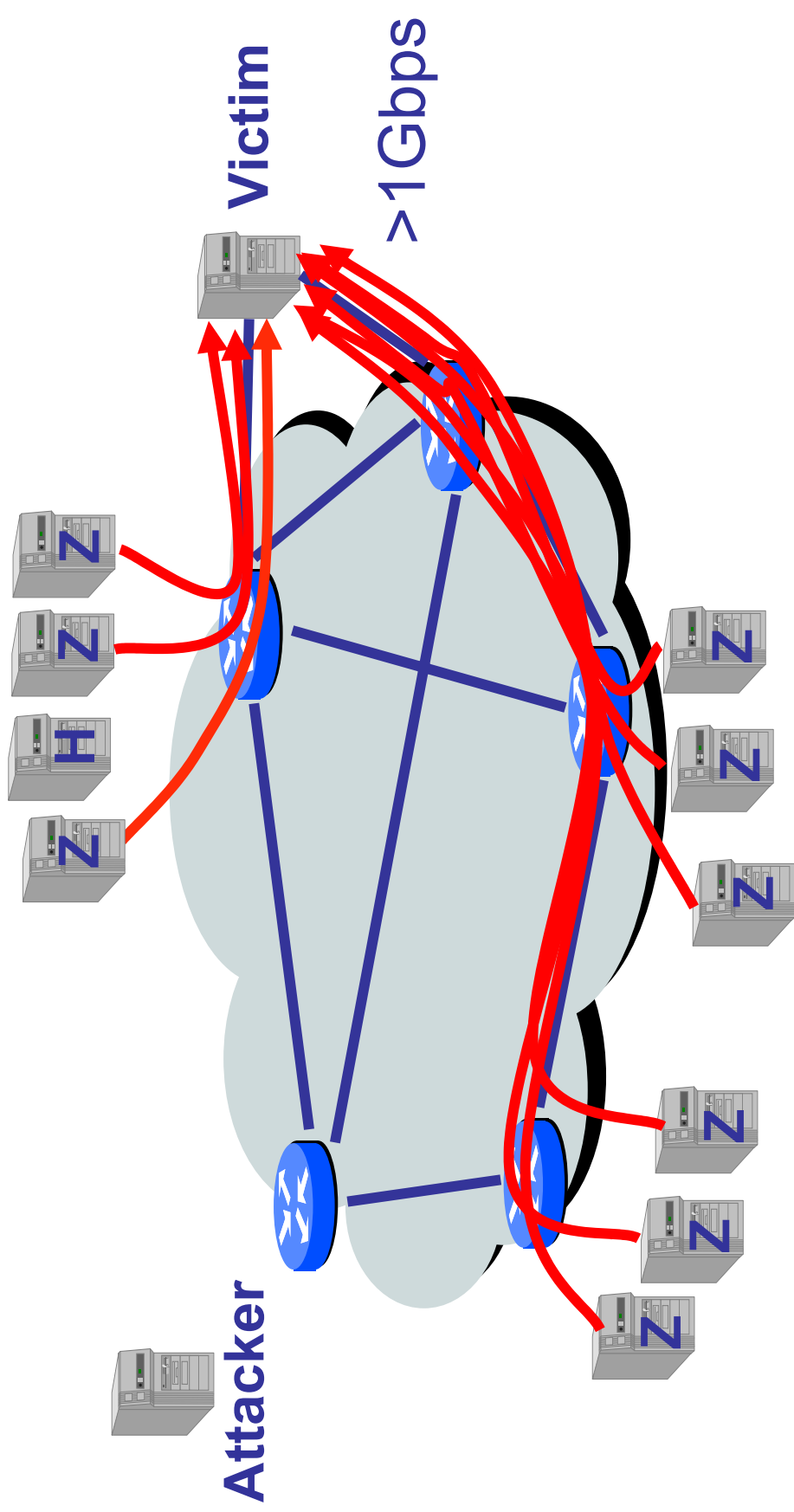
Step 2: Attacker sends commands to handler



Step 3: Handler sends commands to zombies



Step 4: Zombies attack target



Step 5: Victim suffers

- Server CPU/Memory resources
 - ♦ Consumes connection state (e.g. SYN flood)
 - ♦ Time to evaluate messages (interrupt livelock)
 - » Some messages take “slow path” (e.g. invalid ACK)
 - ♦ Can cause new connections to be dropped and existing connections to time-out
- Network resources
 - ♦ Routers PPS limited, FIFO queuing
 - ♦ If attack is greater than forwarding capacity, good data will be dropped

Simple question

**How prevalent are
denial-of-service attacks?**

Most data is anecdotal

Press reports:



Analysts:

“Losses ... could total more than \$1.2 billion”
- *Yankee Group* report

Surveys:

“38% of security professionals surveyed reported denial of service activity in 2000”

Quantitative data?

- **Isn't available** (i.e. no one knows)
- **Inherently hard to acquire**
 - ♦ Few content or service providers collect such data
 - ♦ If they do, its usually considered sensitive
- **Infeasible to collect** at Internet scale
 - ♦ How to monitor enough to the Internet to obtain a representative sample?

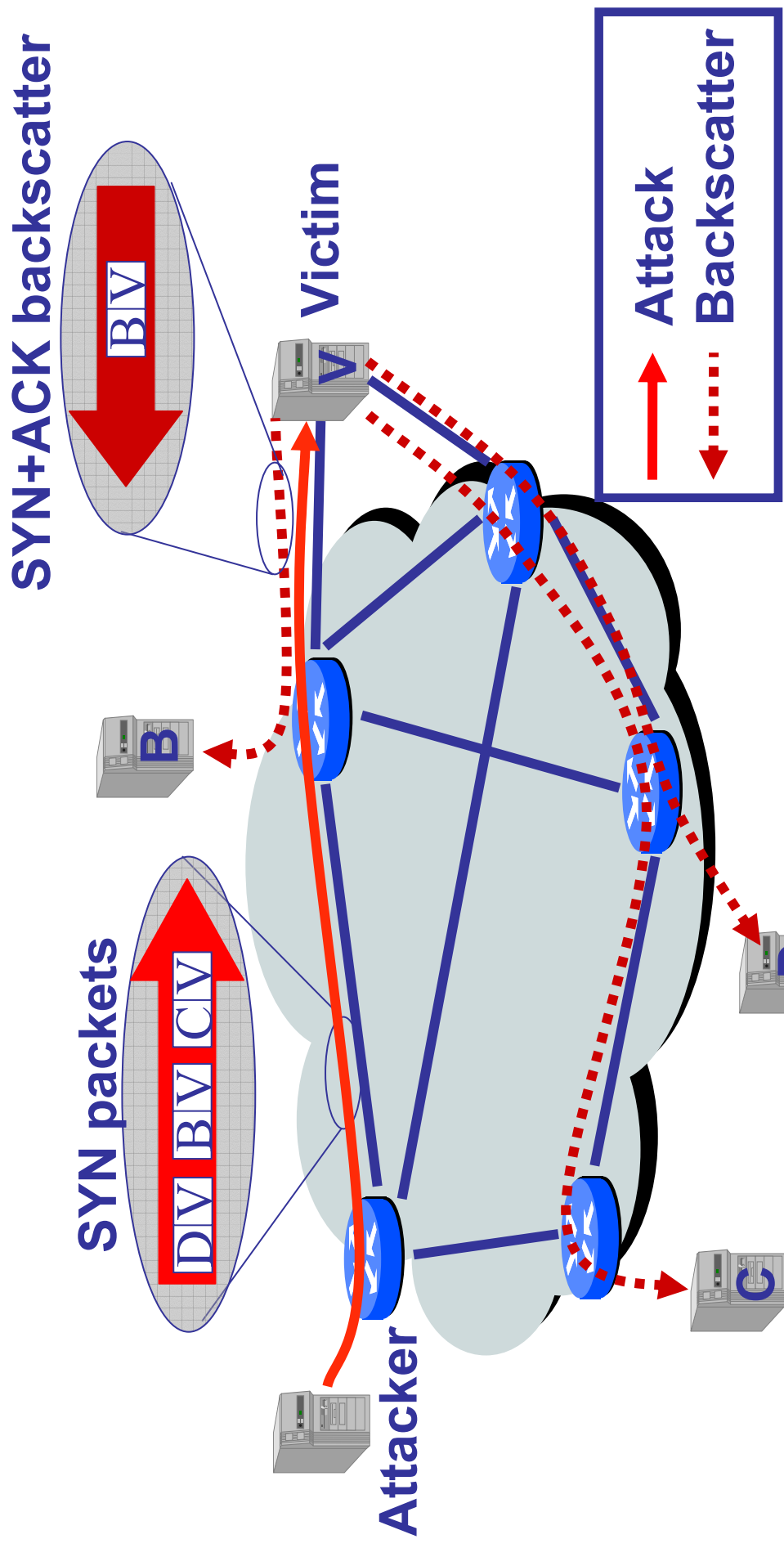
A good estimate: [Moore, Voelker, Savage01]

- Backscatter analysis
 - ◆ New technique for estimating **global** denial-of-service activity
- First data describing Internet-wide DoS activity
 - ◆ **~4,000** attacks per week (> **12,000** over 3 weeks)
 - ◆ Instantaneous loads above **600k pps**
 - ◆ Characterization of attacks and victims

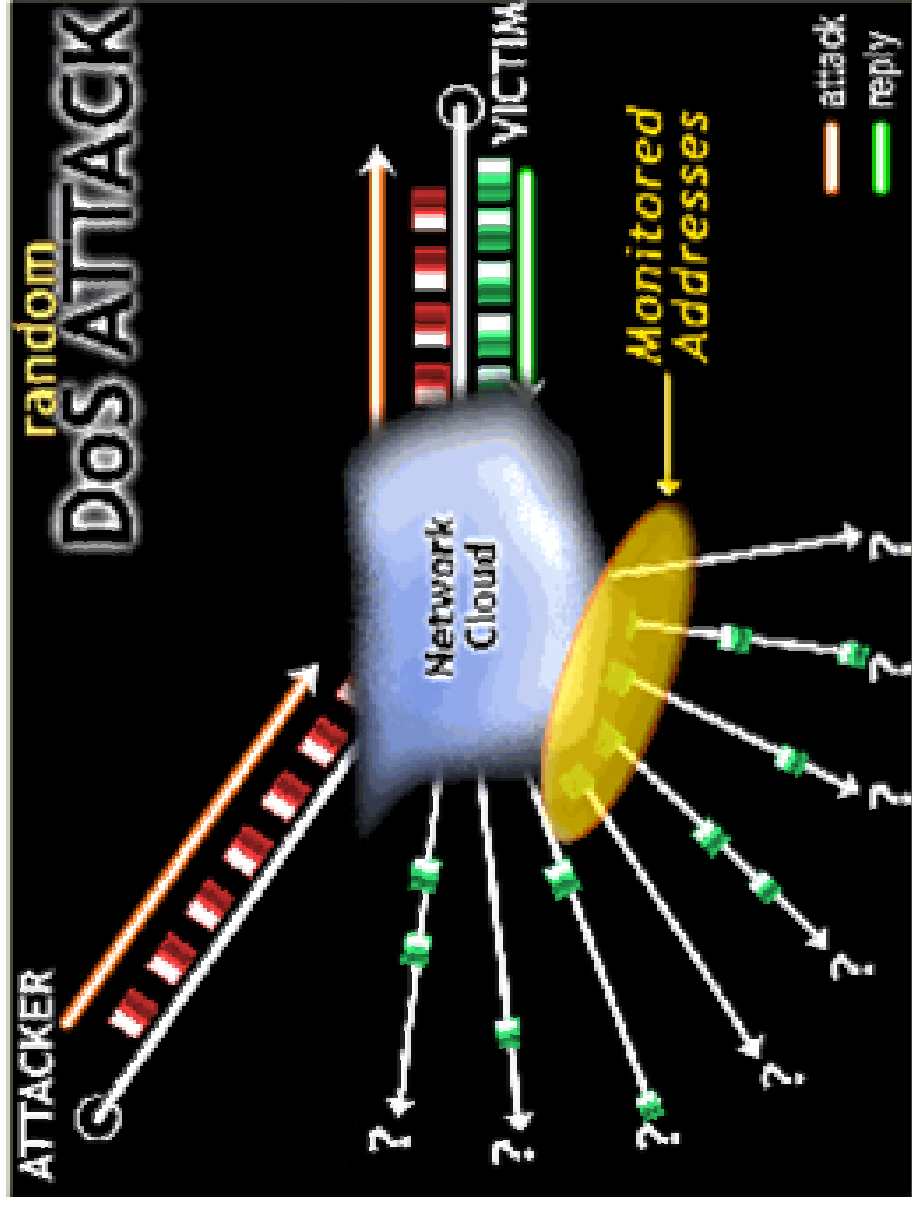
Key idea

- Flooding-style DoS attacks
 - ◆ e.g. SYN flood, ICMP flood
- Attackers spoof source address **randomly**
 - ◆ True of all major attack tools
- Victims, in turn, respond to attack packets
- Unsolicited responses (*backscatter*) equally distributed across IP address space
- Received backscatter is **evidence** of an attacker elsewhere

Random IP spoofing produces random backscatter



Example



Backscatter analysis

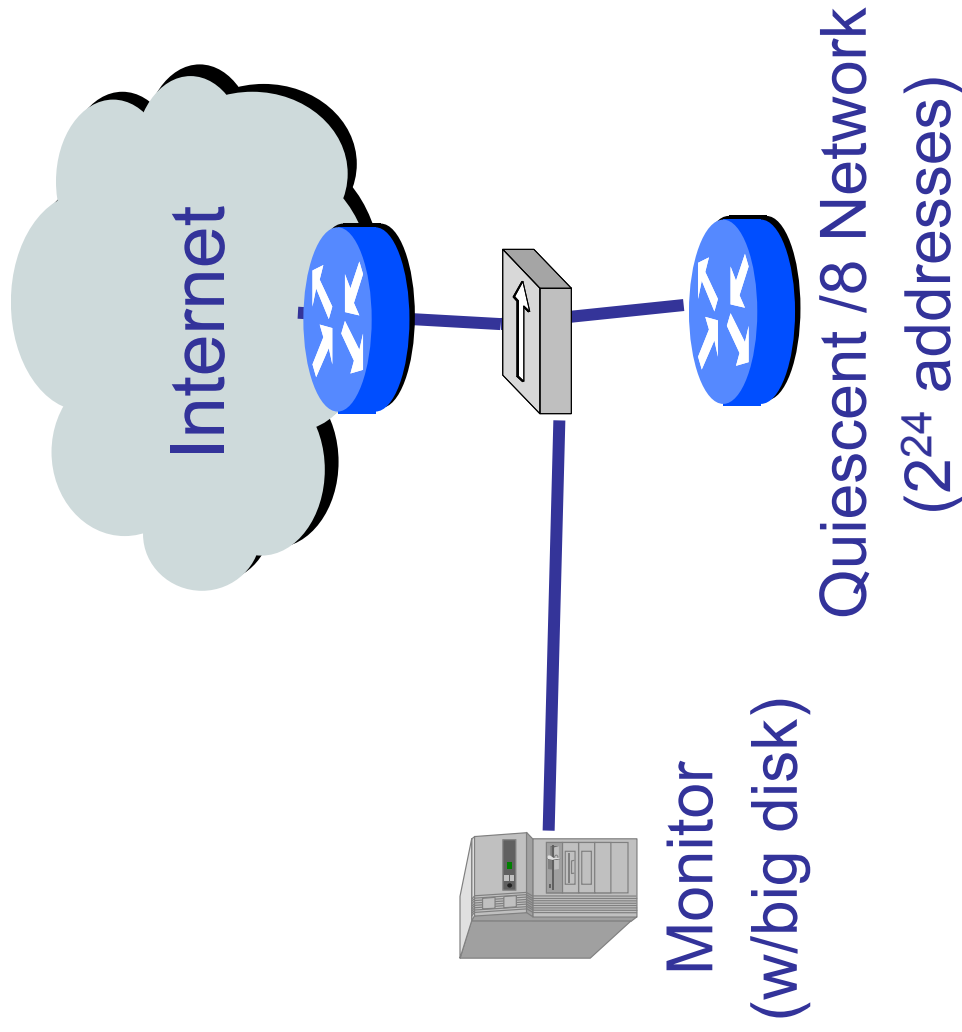
- Monitor block of n IP addresses
- Expected # of backscatter packets given an attack of m packets:

$$E(X) = \frac{nm}{2^{32}}$$

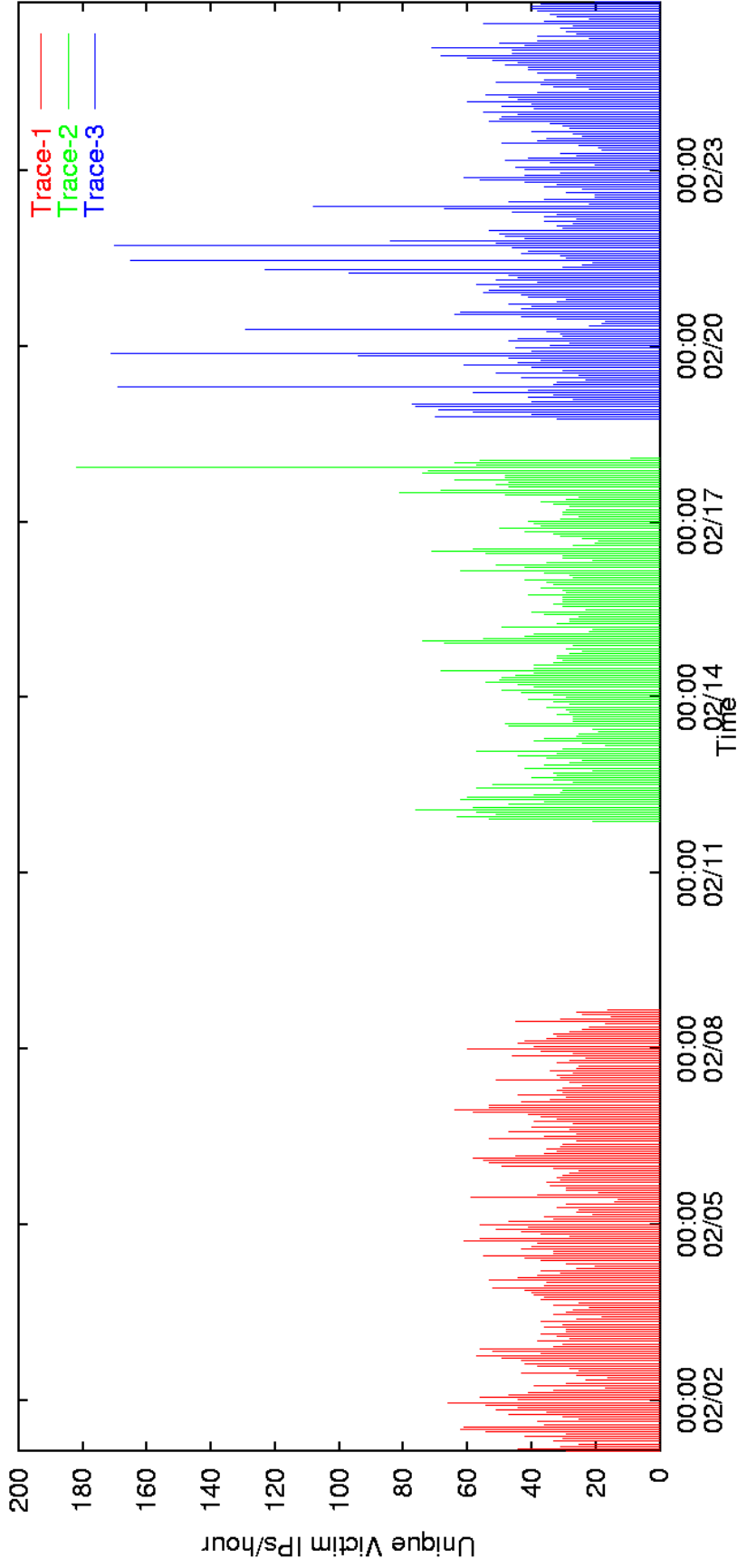
- Extrapolated attack rate R' is a function of measured backscatter rate R :

$$R \geq R' \frac{2^{32}}{n}$$

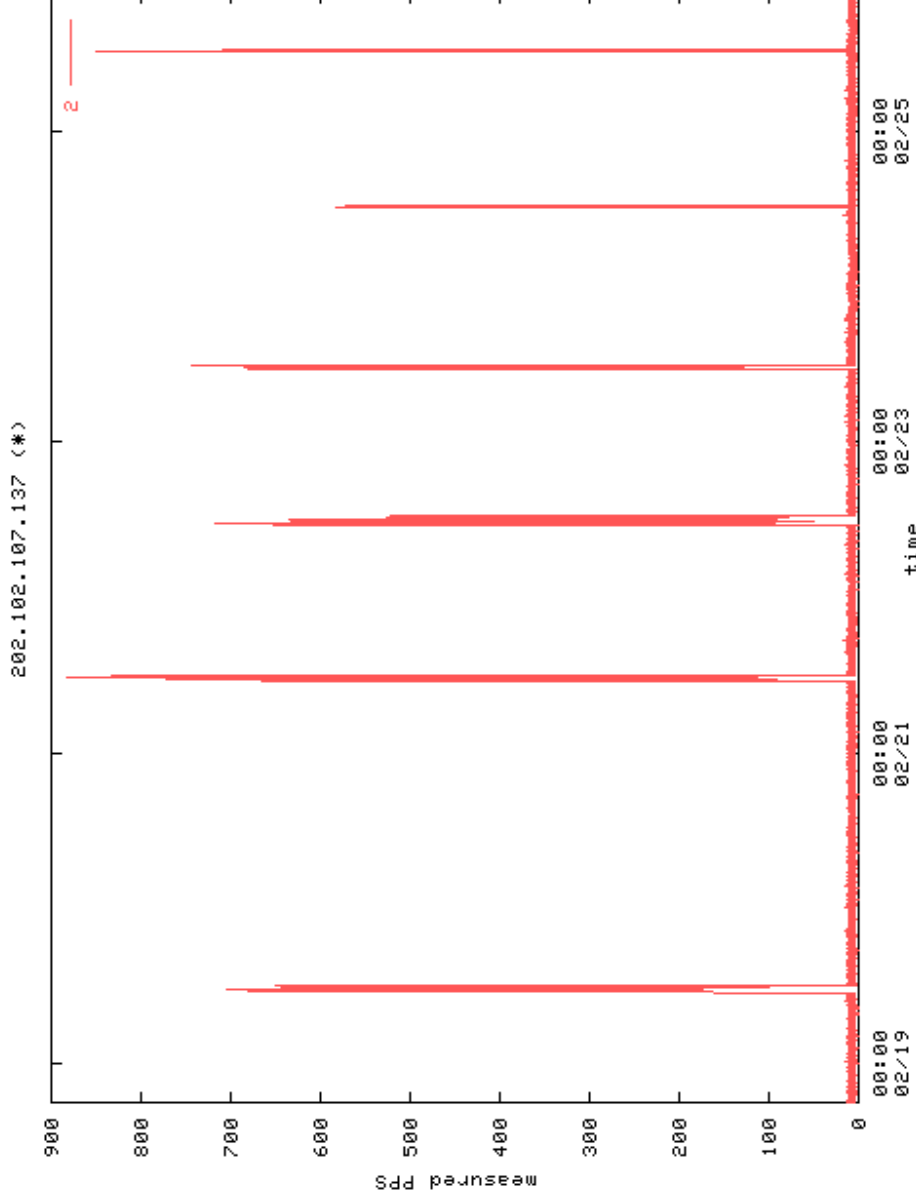
Experimental apparatus...



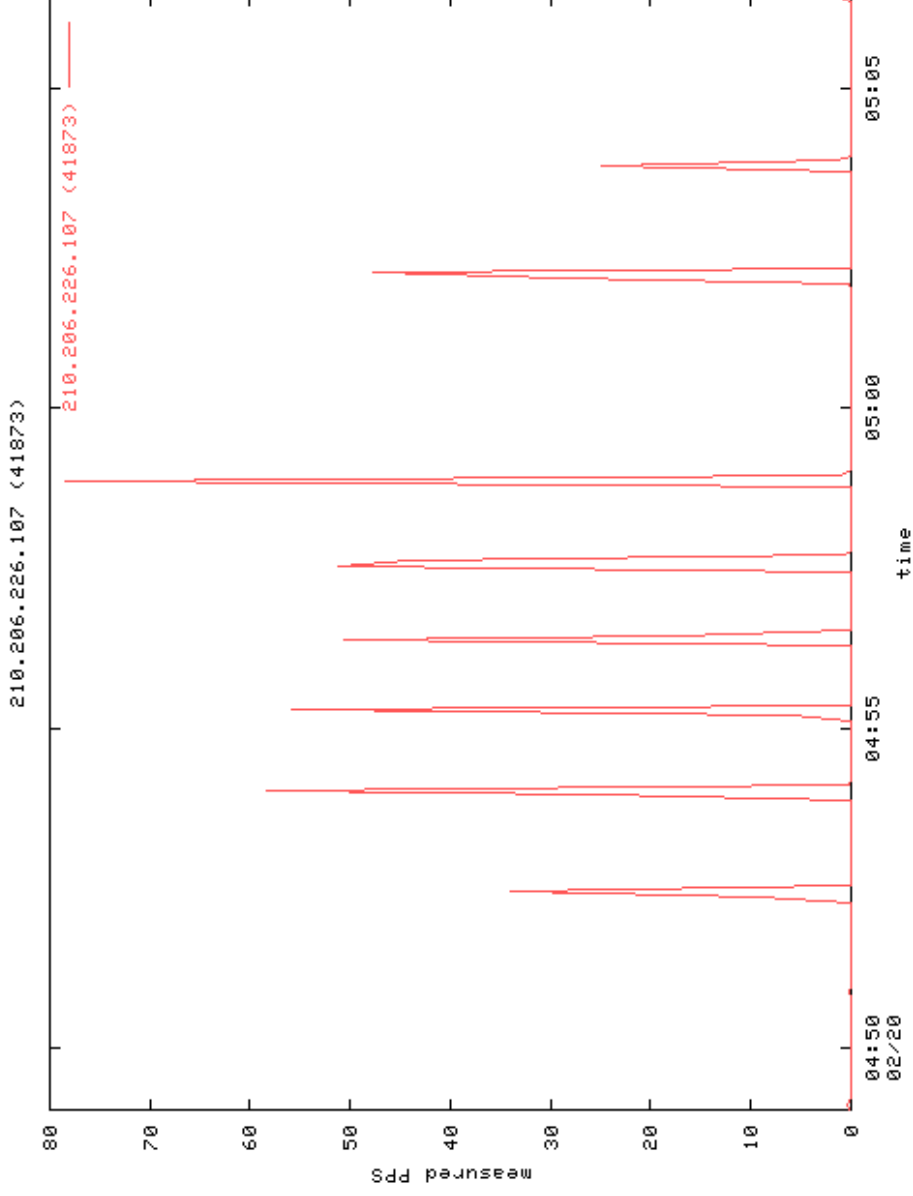
Attacks over time



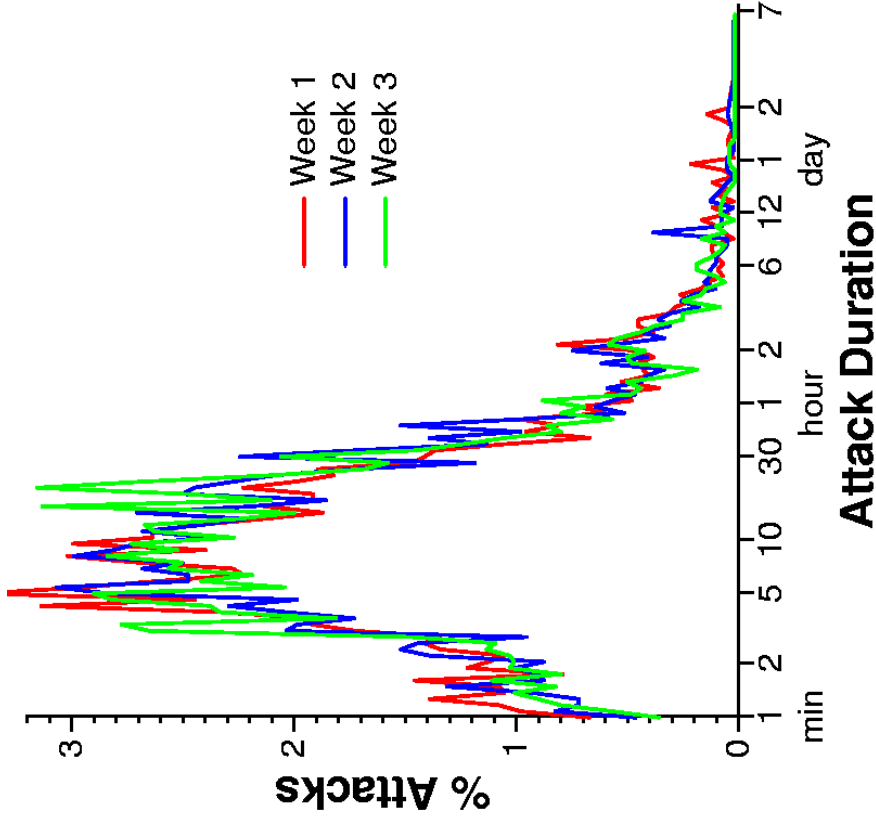
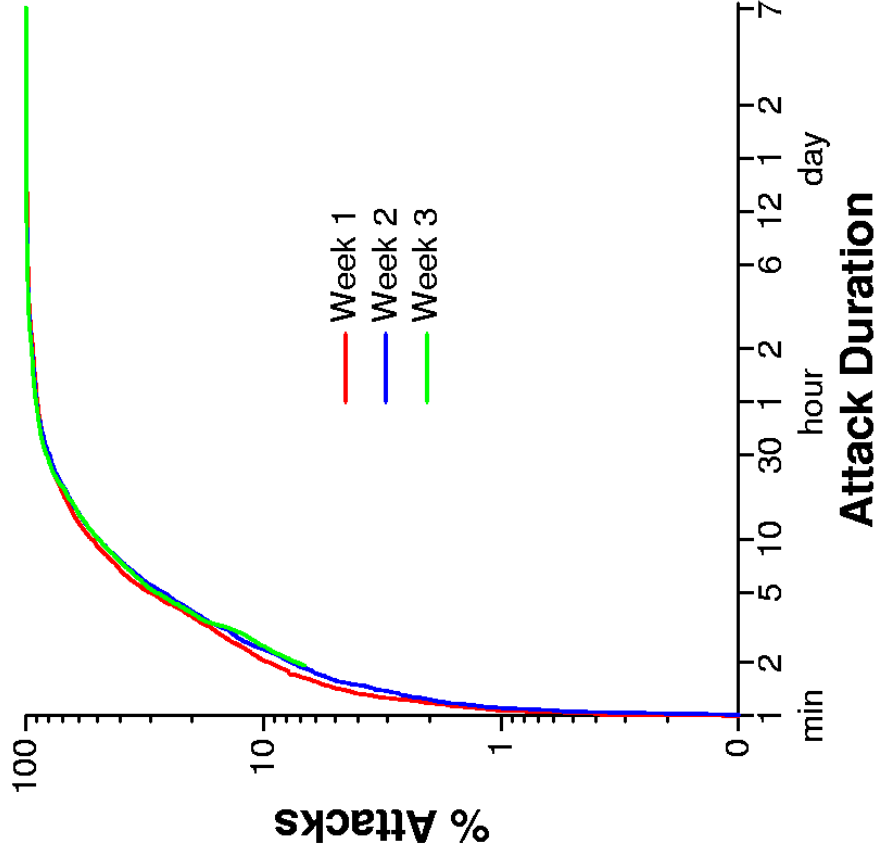
Example 1: Periodic attack (1hr per 24hrs)



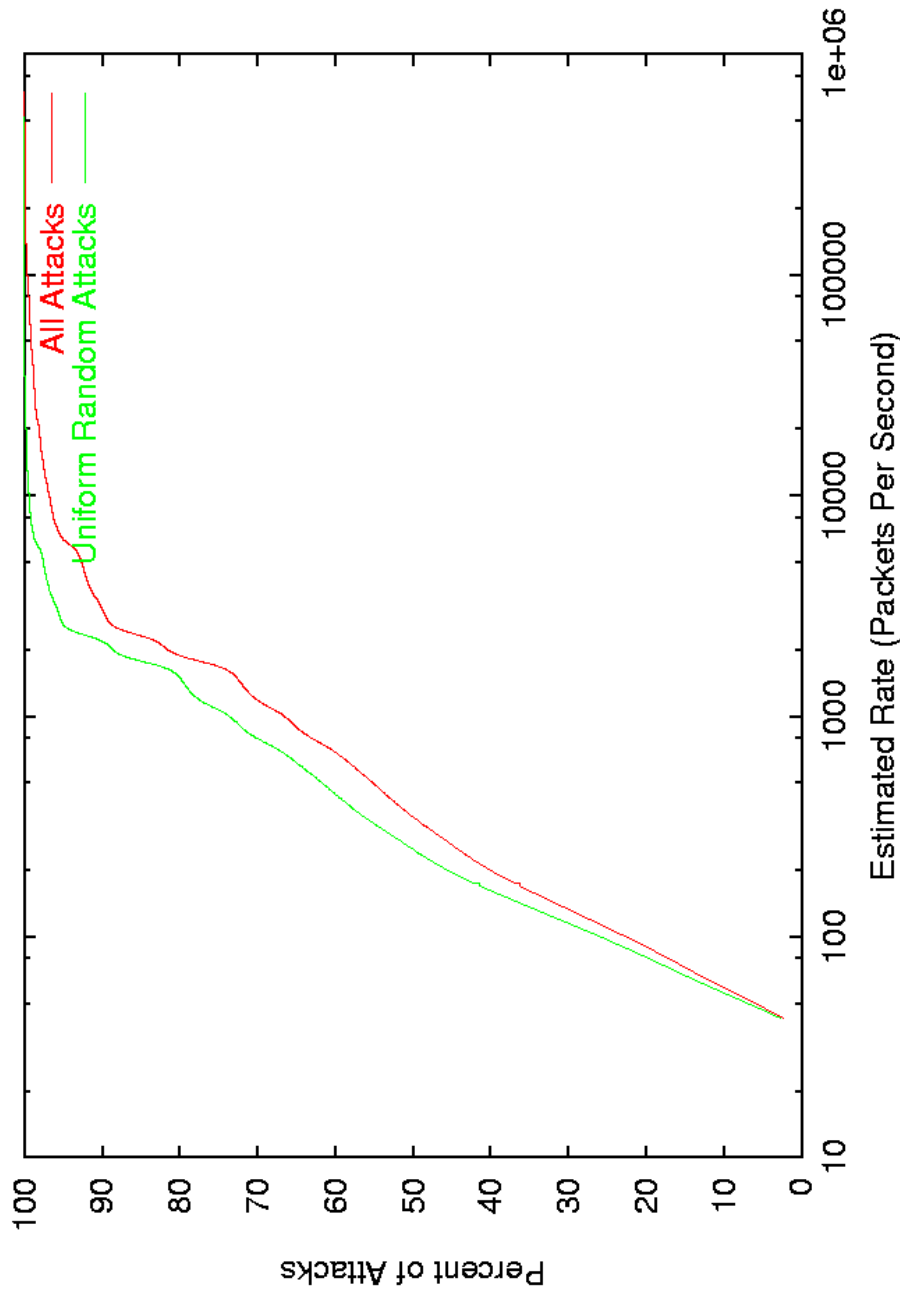
Example 2: Punctuated attack (1min interval)



Attack duration distribution



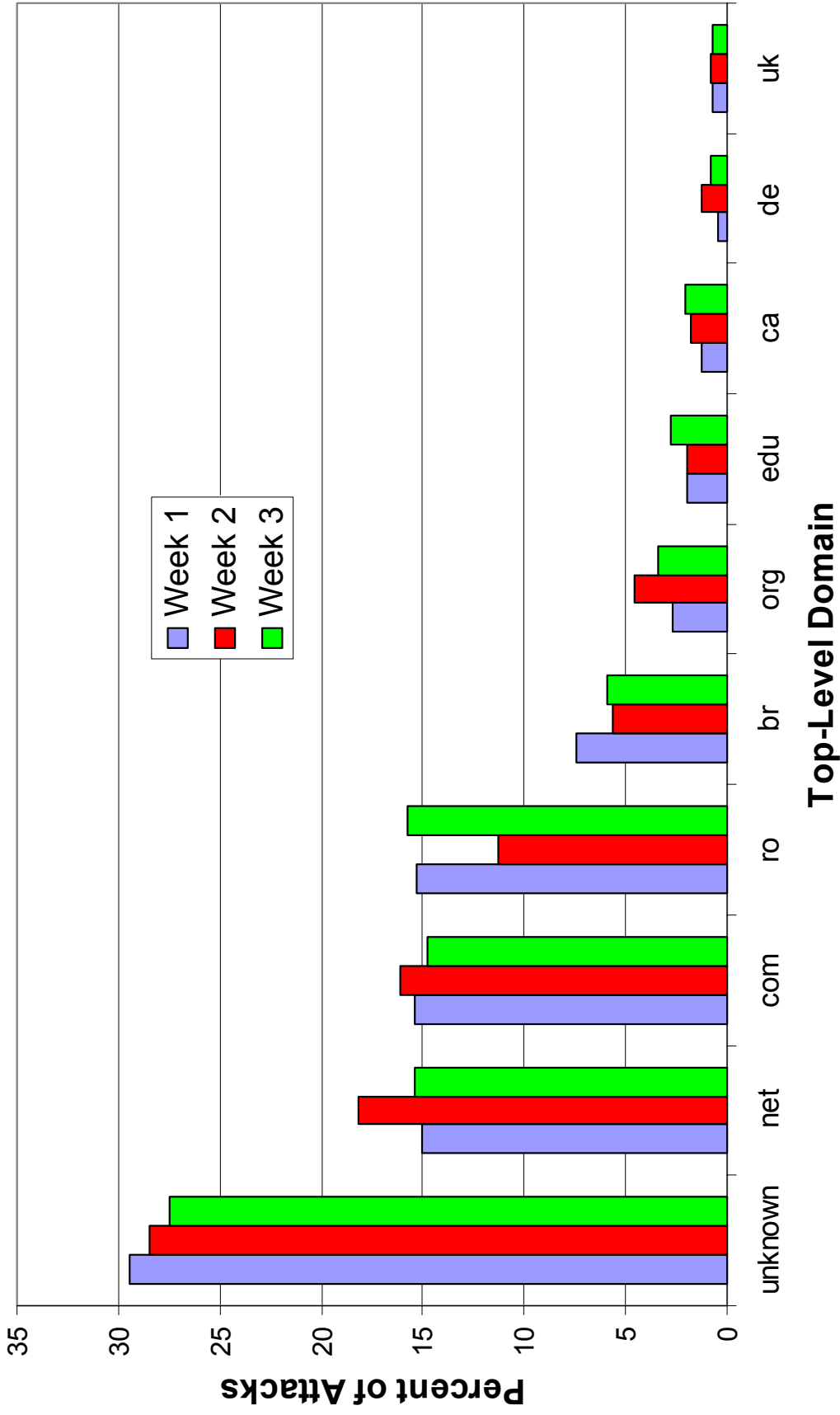
Attack rate distribution



Victim characterization by DNS name

- Entire spectrum of commercial businesses
 - ◆ Yahoo, CNN, Amazon, etc and many smaller biz
- Evidence that minor DoS attacks used for personal vendettas
 - ◆ 10-20% of attacks to home machines
 - ◆ A few very large attacks against broadband
 - ◆ Many reverse mappings clearly compromised (e.g. is.on.the.net.illegal.ly and the.feds.cant.secure.their.shellz.ca)
- 5% of attack target infrastructure
 - ◆ Routers (e.g. core2-core1-oc48.paol.above.net)
 - ◆ Name servers (e.g. ns4.reliablehosting.com)

Victim breakdown by TLD

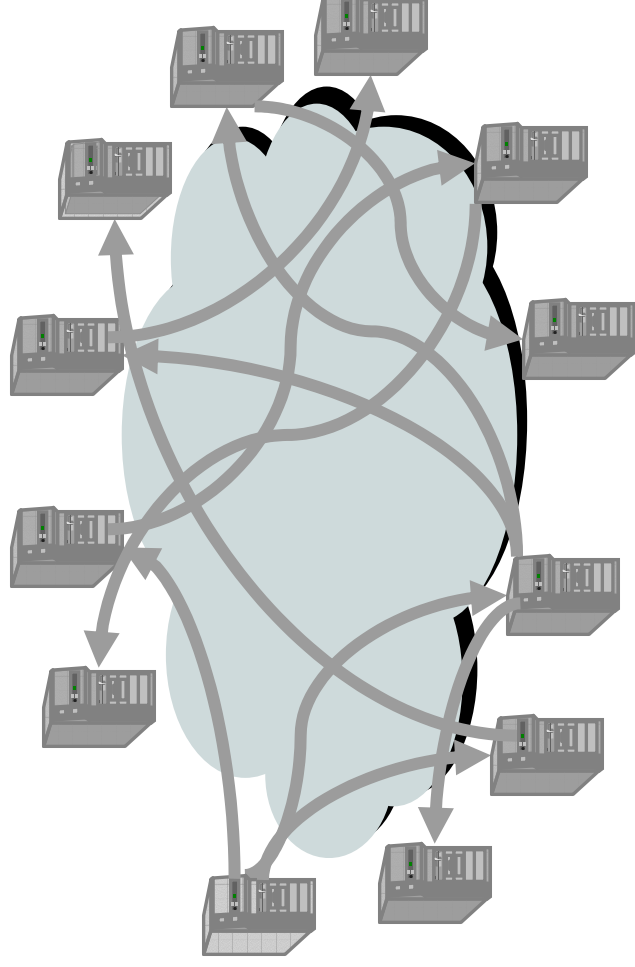


Denial-of-Service summary

- Lots of attacks – some very large
 - ◆ >**12,000** attacks against >**5,000** targets in a week
 - ◆ Most < **1,000** pps, but some over **600,000** pps
- Everyone is a potential target
 - ◆ Targets not dominated by any TLD, 2LD or AS
 - » Targets include large e-commerce sites, mid-sized business, ISPs, government, universities and end-users
 - ◆ Something weird is happening in Romania
- New attack “styles”
 - ◆ Punctuated/periodic attacks
 - ◆ Attacks against infrastructure targets & broadband

What is a Network Worm?

- Self-propagating self-replicating network program
 - ♦ Exploits some vulnerability to infect remote machines
 - » No human intervention necessary
 - ♦ Infected machines continue propagating infection



A Brief History...

- Brunner describes “tapeworm” program in novel “Shockwave Rider” (1972)
- Shoch&Hupp co-opt idea; coin term “worm” (1982)
 - ◆ Key idea: programs that self-propagate through network to accomplish some task
 - ◆ Benign; didn’t replicate
- Fred Cohen demonstrates power and threat of self-replicating viruses (1984)
- Morris worm exploits buffer overflow vulnerabilities & infects a few thousand hosts (1988)

Hiatus for 13 years...

Recent Events

- **CodeRed** worm released in Summer 2001
- Exploited buffer overflow in IIS
- Uniform random target selection
 - ♦ Pick IP address at random from 2^{32}
 - ♦ Can measure using same apparatus as DoS measurement
 - ♦ If unsolicited request arrives then a worm or a port scan
 - ♦ [Moore et al, 2002]
- Infects 360,000 hosts in less than 10 hours (CRv2)

Modeling network worms

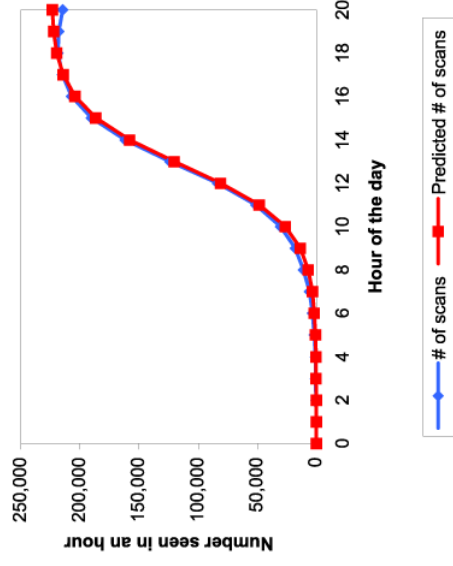
- Network worms are well modeled as infectious epidemics
 - ♦ Homogeneous random contacts
- Classic SI model

$$\frac{dI}{dt} = \beta \frac{IS}{N} \quad \rightarrow \quad \frac{di}{dt} = \beta i(1-i)$$
$$\frac{dS}{dt} = -\beta \frac{IS}{N}$$

- » N: population size
- » S(t): susceptible hosts at time t
- » I(t): infected hosts at time t
- » β : contact rate
- » i(t): I(t)/N, s(t): S(t)/N

$$i(t) = \frac{e^{\beta(t-T)}}{1 + e^{\beta(t-T)}}$$

courtesy Paxson, Staniford, Weaver



Since Code Red...

- **Renaissance** in worm development
- CodeRedII, Nimda, Scalper, Slapper, etc.. soon follow
 - ◆ Multiple vulnerabilities, backdoors on machine, biased target selection (more likely to try infecting machines in same network)
- Sapphire worm (Winter 2003)
 - ◆ Open loop scanning – bandwidth limited
 - ◆ **Scanned most of Internet in << 10mins**
 - ◆ Infected ~100,000 hosts

Gloom and Doom

- **Most potent computer security threat today**
 - ◆ Many *millions* of susceptible hosts
 - ◆ Easy to write worms
 - » Worm payload separate from vulnerability exploit
 - » Significant code reuse in practice
 - ◆ Possible to cause major damage
 - » Lucky so far; existing worms have benign payload
 - » Wipe disk; flash bios; modify data; reveal data; Internet DoS
- **We have no operational defense**
 - ◆ Evidence suggests that humans don't react fast enough
 - ◆ Defensive technology is nascent at best

What can be done?

- Reduce the number of susceptible hosts
 - ◆ Prevention. Very hard. All software has bugs; software homogeneity makes impact of single vulnerability large
- Reduce the number of infected hosts
 - ◆ Treatment. Very hard. Takes time to understand how to disinfect machines.
- Reduce the contact rate
 - ◆ Containment. Bottom line – how quickly can you detect and react.

Design Issues

[Moore, Shannon, Voelker, Savage03]

- Any reactive defense is defined by:
 - ♦ **Reaction time** – **how long** to detect, propagate information, and activate response
 - ♦ **Containment strategy** – **how** malicious behavior is identified
 - ♦ **Deployment scenario** - **who** participates in the system
- We evaluated the requirements for these parameters to build **any** effective system.

Methodology

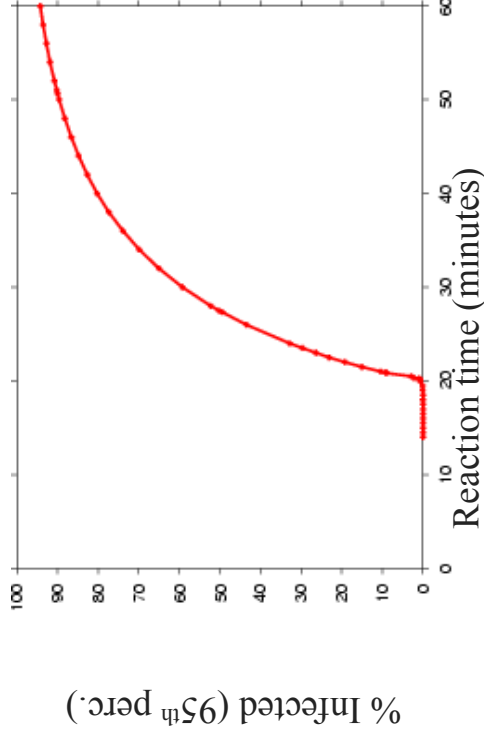
- **Simulate spread of worm across Internet topology:**
 - ♦ infected hosts *attempt* to spread at a fixed rate (probes/sec)
 - ♦ target selection is uniformly random over IPv4 space
- **Simulation of defense:**
 - ♦ system detects infection within reaction time
 - ♦ subset of network nodes employ a containment strategy
- **Evaluation metric:**
 - ♦ % of vulnerable hosts infected in 24 hours
 - ♦ 100 runs of each set of parameters (95th percentile taken)
 - » Systems must plan for reasonable situations, **not** the average case
- **Source data:**
 - ♦ vulnerable hosts: 359,000 IP addresses of CodeRed v2 victims
 - ♦ Internet topology: AS routing topology derived from RouteViews

Initial Approach: Universal Deployment

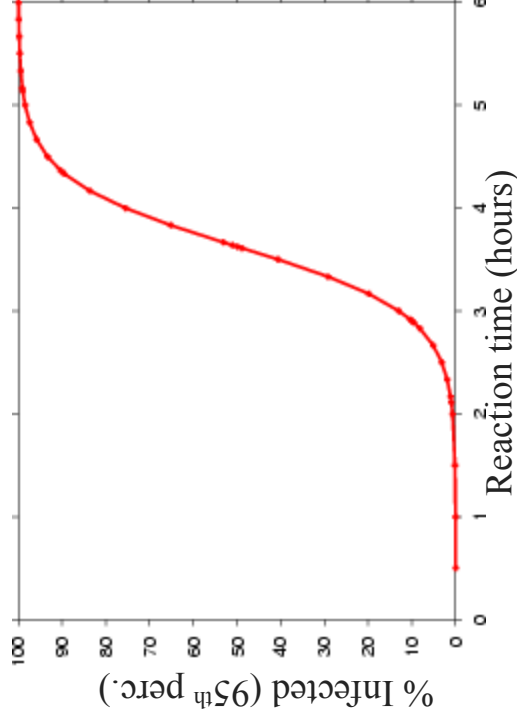
- Assume **every host** employs the containment strategy
- Two natural containment strategies:
 - ♦ **Address blacklisting:**
 - » block traffic from malicious source IP addresses
 - » reaction time is relative to each infected host
 - ♦ **Content filtering:**
 - » block traffic based on signature of content
 - » reaction time is from first infection
- How quickly does each strategy need to react?
- How sensitive is reaction time to worm probe rate?

How quickly does each strategy need to react?

Address Blacklisting:



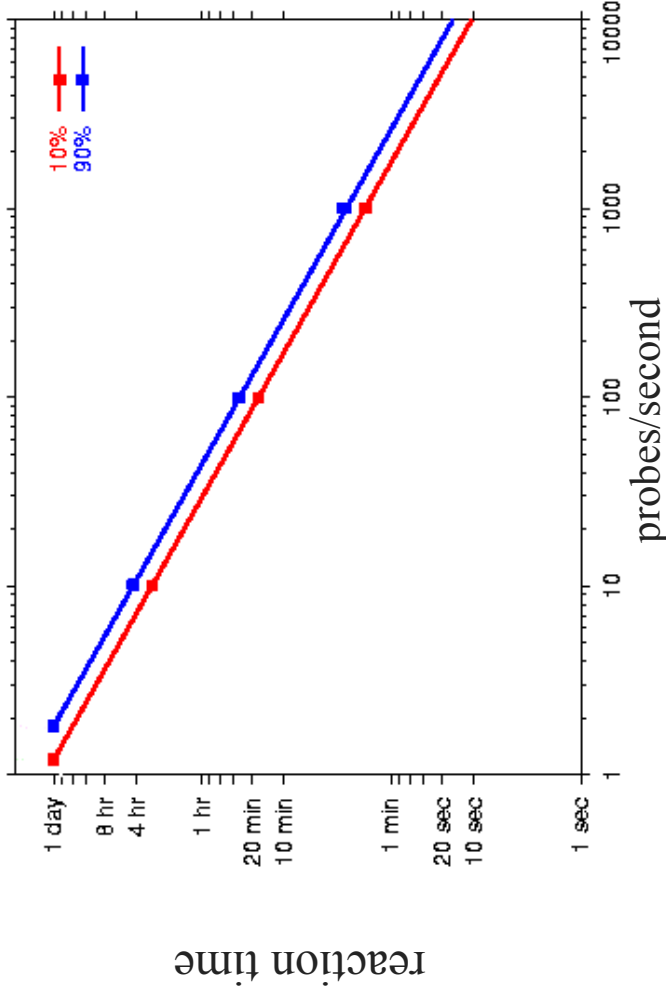
Content Filtering:



- To contain worms to 10% of vulnerable hosts after 24 hours of spreading at 10 probes/sec (CodeRed):
 - ◆ Address blacklisting: reaction time must be < 25 minutes.
 - ◆ Content filtering: reaction time must be < 3 hours

How sensitive is reaction time to worm probe rate?

Content Filtering:



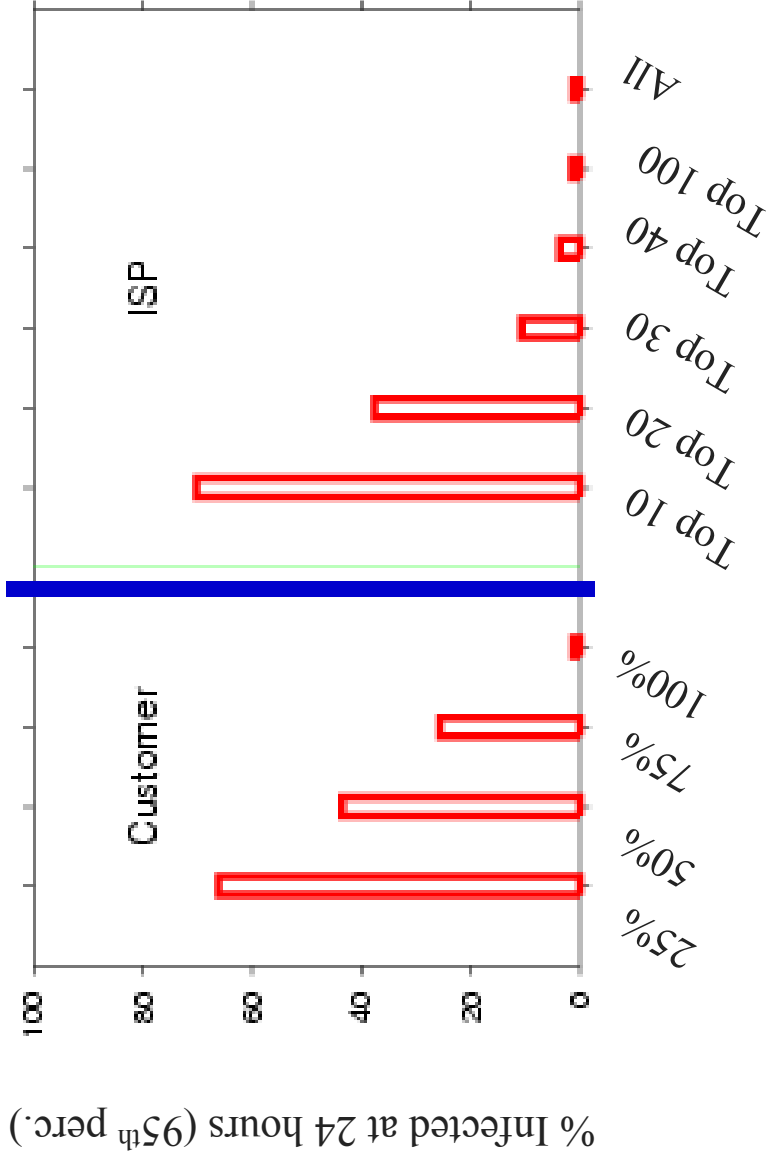
- Reaction times must be fast when probe rates get high:
 - ◆ 10 probes/sec: reaction time must be < 3 hours
 - ◆ 1000 probes/sec: reaction time must be < 2 minutes

Limited Network Deployment

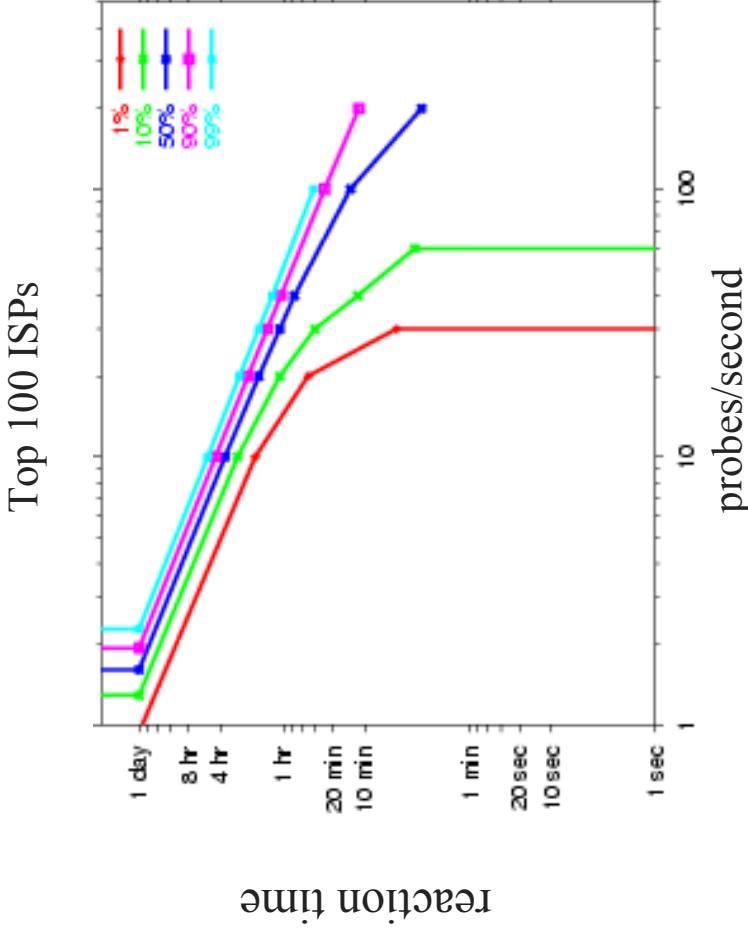
- Depending on every host to implement containment is not feasible:
 - ♦ installation and administration costs
 - ♦ system communication overhead
- A more realistic scenario is limited deployment in the **network**:
 - ♦ Customer Network: firewall-like inbound filtering of traffic
 - ♦ ISP Network: traffic through border routers of large transit ISPs
- How effective are the deployment scenarios?
- How sensitive is reaction time to worm probe rate under limited network deployment?

How effective are the deployment scenarios?

CodeRed-like Worm:



How sensitive is reaction time to worm probe rate?



- Above 60 probes/sec, containment to 10% hosts within 24 hours is impossible even with *instantaneous* reaction.

Summary

- Reaction time:
 - ◆ required reaction times are a couple minutes or less
- Containment strategy:
 - ◆ content filtering is more effective than address blacklisting
- Deployment scenarios:
 - ◆ need nearly all customer networks to provide containment
 - ◆ need at least top 40 ISPs provide containment

Worm summary

- Network worms are increasing both in frequency and virulence
- Incident time-scales **requires** automated defense
- Reactive systems can be built to contain some worms, but the engineering challenges are huge

Other security issues...

- Detecting/tracking denial-of-service attacks
- Statically/dynamically detecting likely program vulnerabilities (e.g. buffer overflows, race conditions)
- Steganography
- Digital watermarking, copy protection
- Revocable credentials
- Secure and/or anonymous storage
- Micropayments
- Side-channel attacks (timing, power, etc)
- Tamper resistant environments
- Worms, viruses, etc...

Summary

- Security is a huge field, poorly fleshed out
- Mostly based on trust
 - ◆ Authenticity, confidentiality, integrity to establish trust with outsider
 - ◆ Firewalls/IDS define trusted vs untrusted infrastructure
 - ◆ If you don't have trust, these measures don't help
- **Every** protocol in use today likely has security holes
 - ◆ We don't design for the adversary
- How many of the flaws we discussed today still exist?

Next Time

- Potpourri... QoS, IPv6, etc...