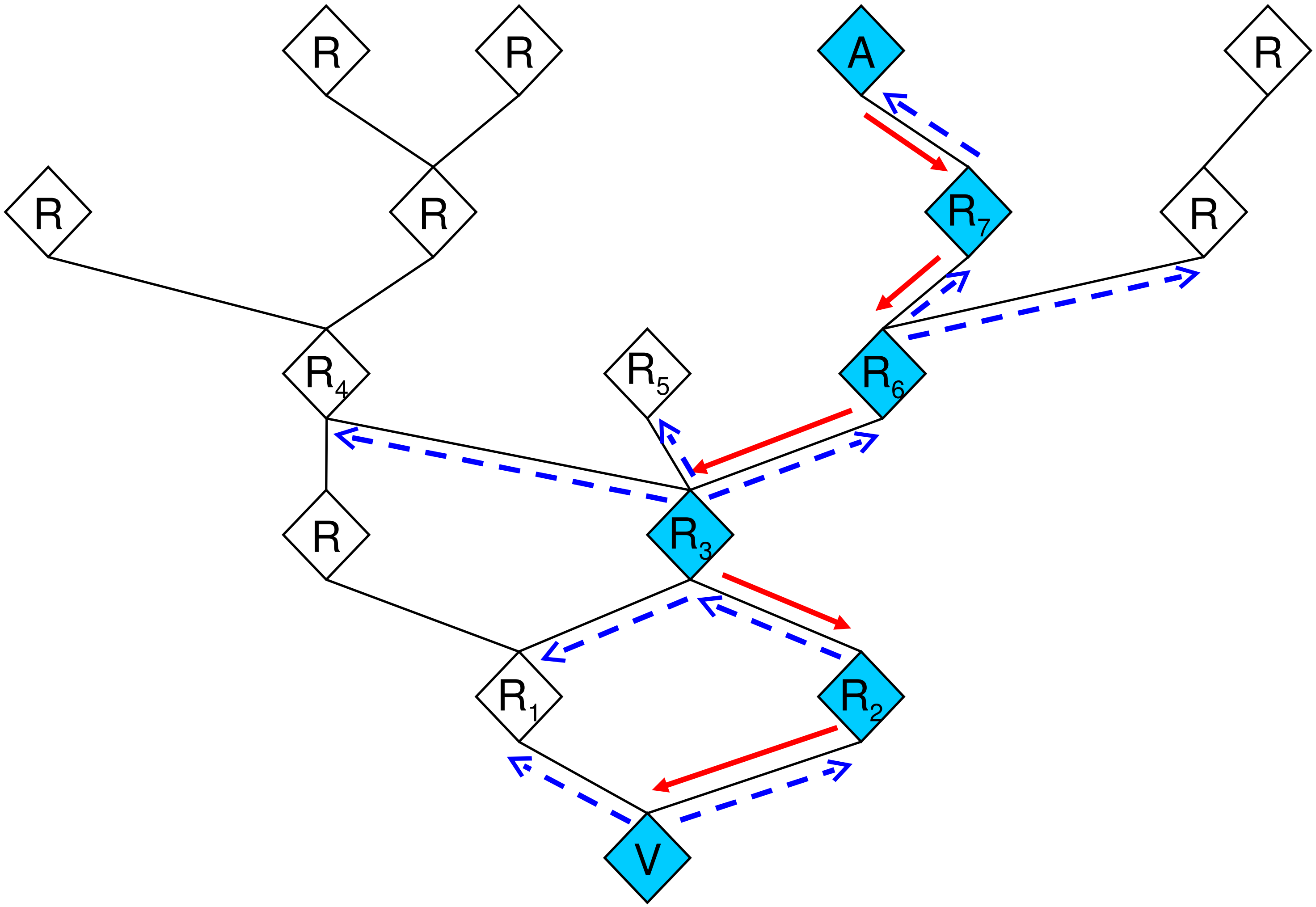
Defenses, Application-Level Attacks, etc.

Nick Feamster

CS 7260
April 4, 2007

IP Traceback



Logging Challenges

- Attack path reconstruction is difficult
  - Packet may be transformed as it moves through the network
- Full packet storage is problematic
  - Memory requirements are prohibitive at high line speeds (OC-192 is ~10Mpkt/sec)
- Extensive packet logs are a privacy risk
  - Traffic repositories may aid eavesdroppers

Single-Packet Traceback: Goals

- Trace a *single* IP packet back to source
  - Asymmetric attacks (*e.g.,* Fraggle, Teardrop, ping-of-death)

- Minimal cost (resource usage)

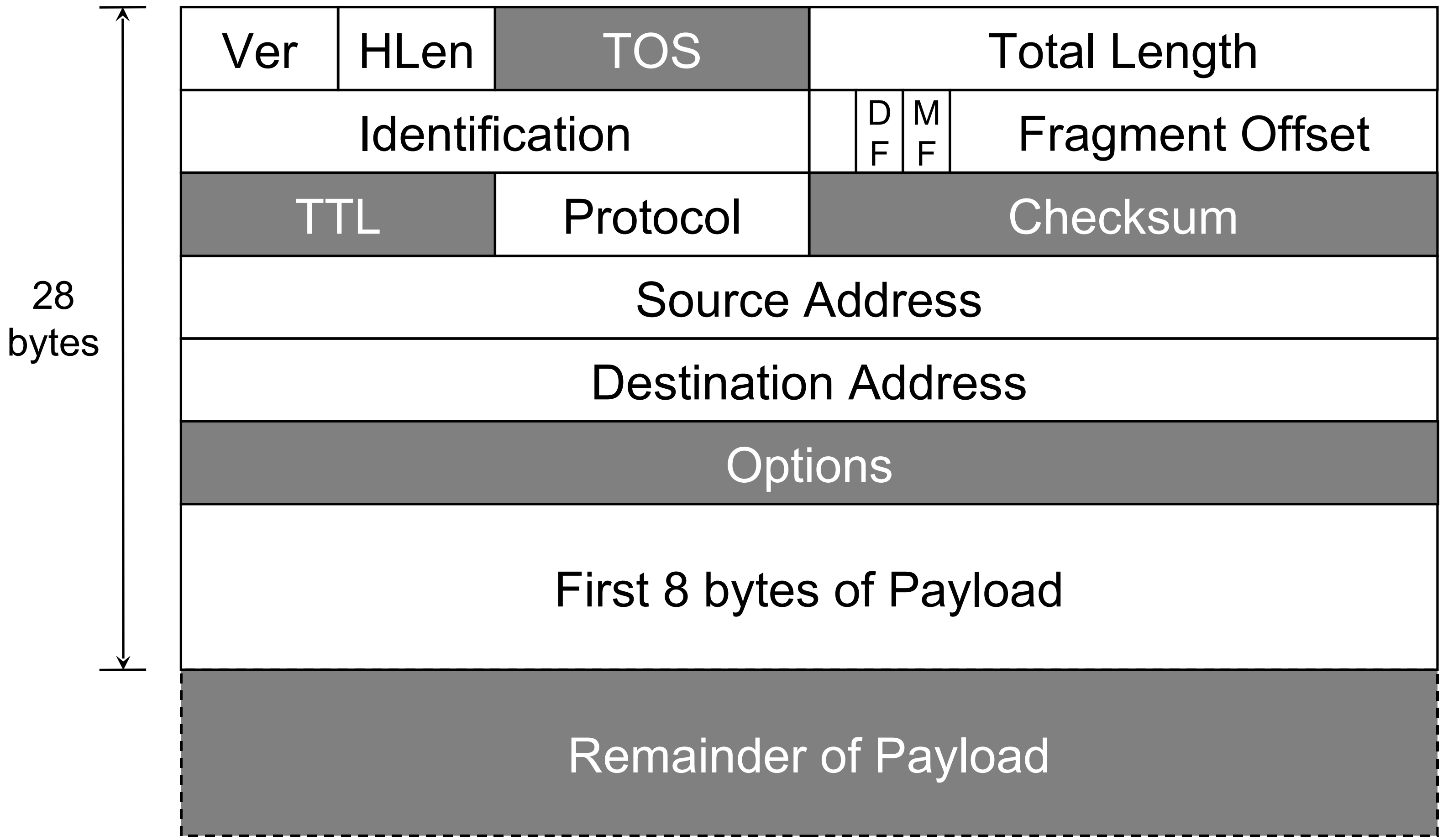
One solution: Source Path Isolation Engine (SPIE)

Packet Digests

- Compute hash(p)
  - Invariant fields of p only
  - 28 bytes hash input, 0.00092% WAN collision rate
  - Fixed sized hash output, *n*-bits

- Compute *k* independent digests
  - Increased robustness
  - Reduced collisions, reduced false positive rate

Hash input: Invariant Content



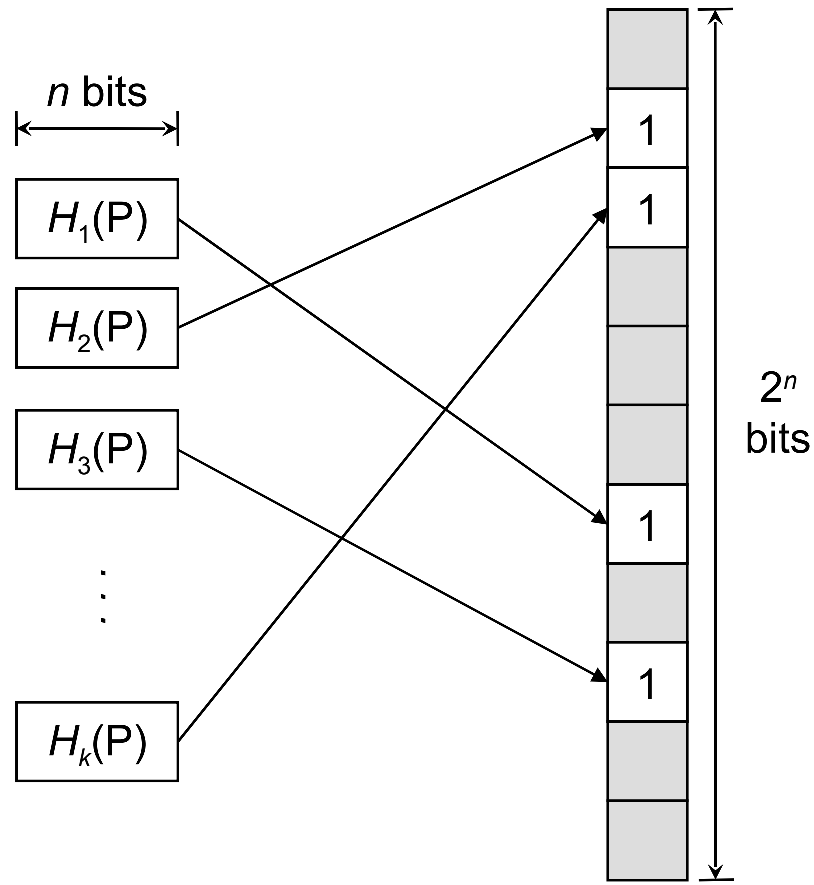
Hashing Properties

- Each hash function
  - Uniform distribution of input -> output

    H1(x) = H1(y) for some x,y -> unlikely
- Use k independent hash functions
  - Collisions among k functions independent
  - H1(x) = H2(y) for some x,y -> unlikely
- Cycle k functions every time interval, t

Digest Storage: Bloom Filters

- **Fixed structure size**
  - Uses 2^n bit array
  - Initialized to zeros
- **Insertion**
  - Use *n*-bit digest as indices into bit array
  - Set to ‘1’
- **Membership**
  - Compute *k* digests, d_1, d_2, etc…
  - If ($filter[d_i]=1$) for all i, router forwarded packet

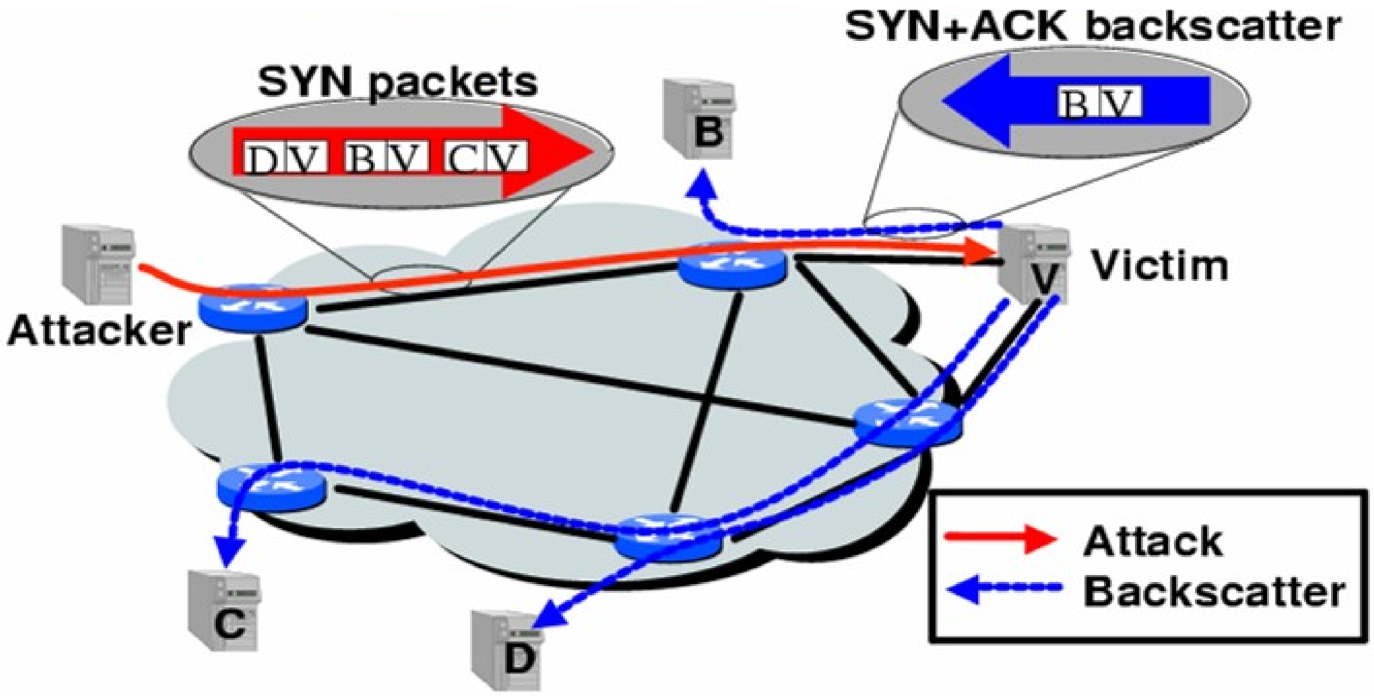


Other In-Network Defenses

- Automatic injection of blackhole routes
- Rerouting through traffic “scrubbers”

Inferring DoS Activity

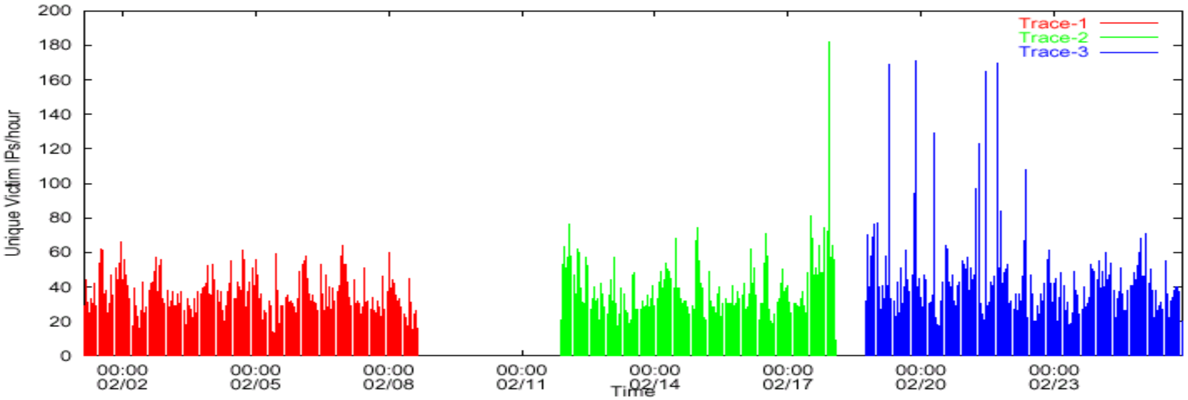
IP address spoofing creates random *backscatter.*

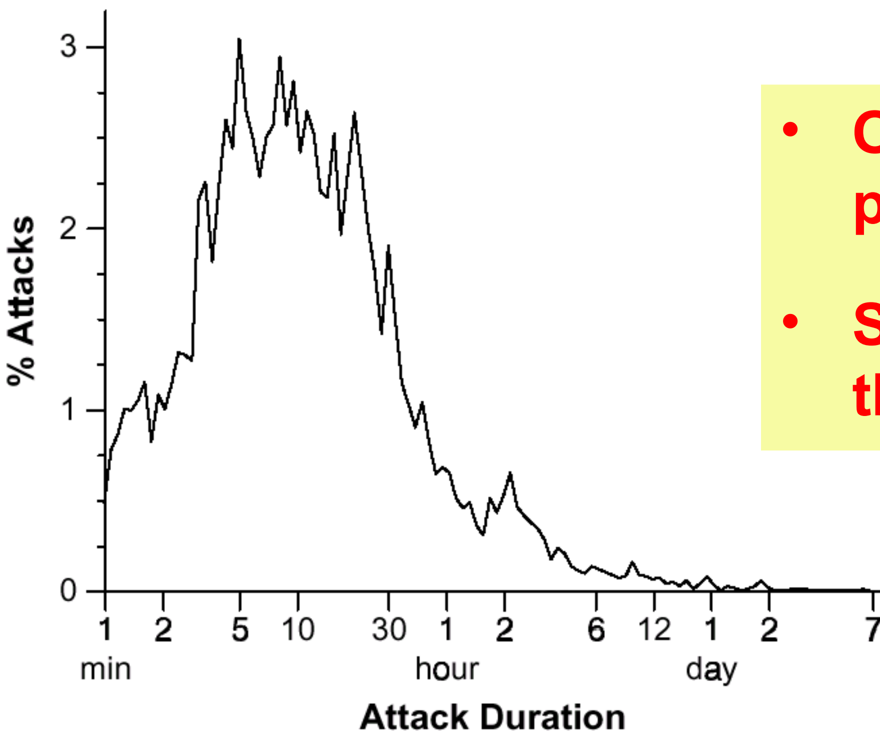


Backscatter Analysis

- Monitor block of *n* IP addresses
- Expected # of backscatter packets given an attack of *m* packets:
  - $E(X) = nm / 2^{32}$
  - Hence, $m = x * (2^{32} / n)$
- Attack Rate $R >= m/T = x/T * (2^{32} / n)$

Inferred DoS Activity





- **Over 4000 DoS/DDoS attacks per week**
- **Short duration: 80% last less than 30 minutes**

Moore *et al.* Inferring Internet Denial of Service Activity

DDoS: Setting up the Infrastructure

- Zombies
  - Slow-spreading installations can be difficult to detect
  - Can be spread quickly with **worms**

- Indirection makes attacker harder to locate
  - No need to spoof IP addresses

What is a Worm?

- Code that replicates and propagates across the network
  - Often carries a “payload”
- Usually spread via exploiting flaws in open services
  - “Viruses” require user action to spread
- **First worm:** Robert Morris, November 1988
  - 6-10% of all Internet hosts infected (!)
- Many more since, but none on that scale until July 2001

Example Worm: Code Red

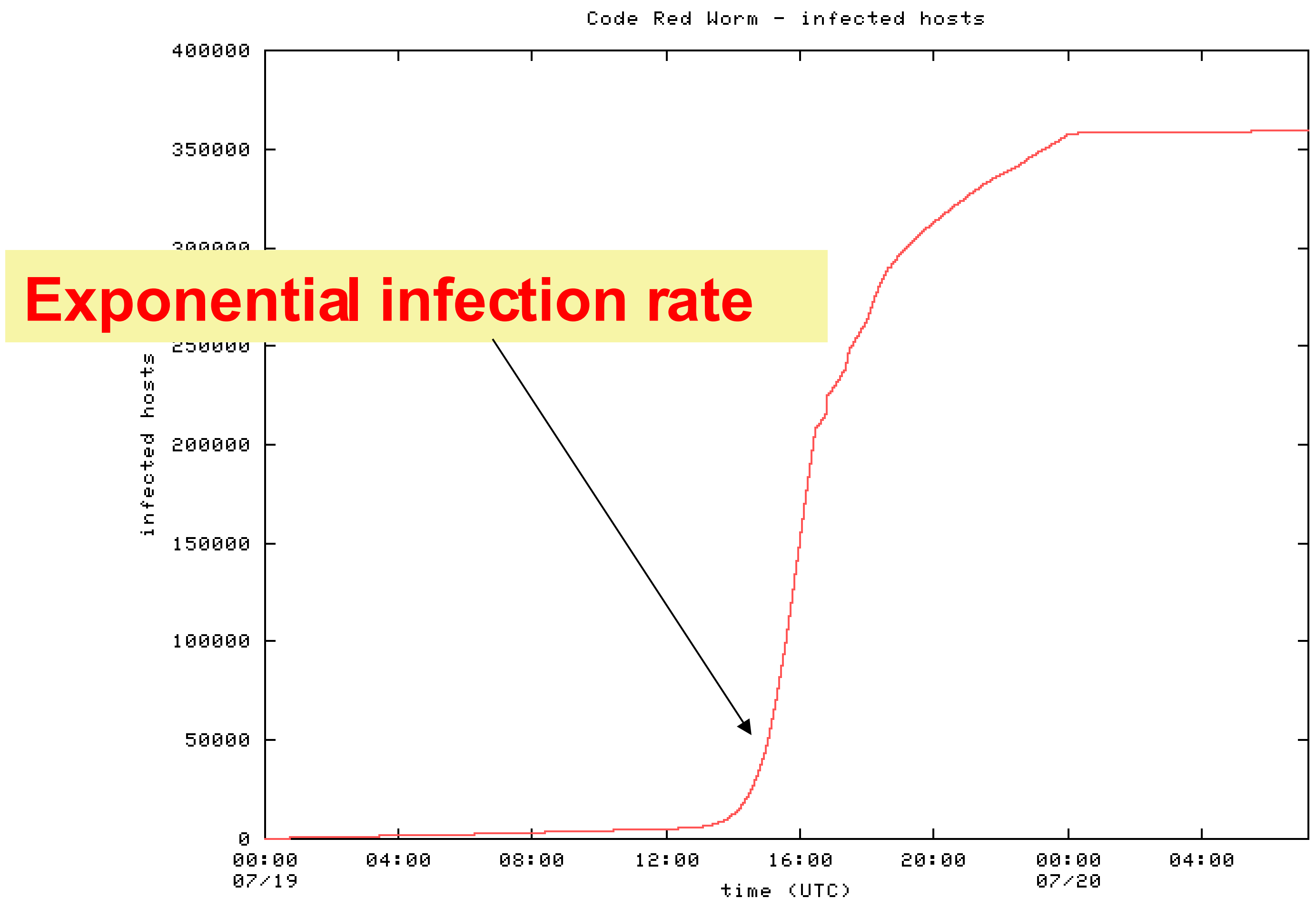
- Initial version: July 13, 2001
- Exploited known ISAPI vulnerability in Microsoft IIS Web servers
- 1st through 20th of each month: spread
  20th through end of each month: attack
- **Payload:** Web site defacement
- **Scanning:** Random IP addresses
- **Bug:** failure to seed random number generator

Code Red: Revisions

- Released July 19, 2001
- **Payload:** flooding attack on www.whitehouse.gov
  - Attack was mounted at the *IP address of the Web site*
- **Bug:** died after 20th of each month
- Random number generator for IP scanning fixed

Code Red: Host Infection Rate

Measured using backscatter technique



Modeling the Spread of Code Red

- Random Constant Spread model
  - *K: initial* compromise rate
  - *N:* number of vulnerable hosts
  - *a*: fraction of vulnerable machines already compromised

$$N da = (N a) K (1 - a) dt$$

Newly infected machines in *dt*

Machines already infected

Rate at which uninfected machines are compromised

Bristling Pace of Innovation

Various improvements to increase the infection rate

- **Code Red 2:** August 2001
  - **Localized scanning**
  - Same exploit, different codebase
  - Payload: root backdoor

- **Nimda:** September 2001
  - Spread via **multiple exploits** (IIS vulnerability, email, CR2 root backdoor, copying itself over network shares, etc.)
  - Firewalls were not sufficient protection

Designing Fast-Spreading Worms

- **Hit-list scanning**
  - Time to infect first 10k hosts dominates infection time
  - **Solution:** Reconnaissance (stealthy scans, etc.)
- **Permutation scanning**
  - **Observation:** Most scanning is redundant
  - **Idea:** Shared permutation of address space. Start scanning from own IP address. Re-randomize when another infected machine is found.
- **Internet-scale hit lists**
  - *Flash worm:* complete infection within 30 seconds

Recent Advances: Slammer

- February 2003
- Exploited vulnerability in MS SQL server
- Exploit fit into a single UDP packet
  - ***Send and forget!***
- Lots of damage
  - BofA, Wash. Mutual ATMs unavailable
  - Continental Airlines ticketing offline
  - Seattle E911 offline

Scary recent advances: Witty

- March 19, 2004
- Single UDP packet exploits flaw in the *passive analysis* of Internet Security Systems products.
- "Bandwidth-limited" UDP worm ala' Slammer.
- Initial spread seeded via a *hit-list*.
- All 12,000 vulnerable hosts infected within 45 mins
- **Payload:** slowly corrupt random disk blocks

Why does DDoS work?

- Simplicity
- “On by default” design
- Readily available zombie machines
- Attacks look like normal traffic
- Internet’s federated operation obstructs cooperation for diagnosis/mitigation

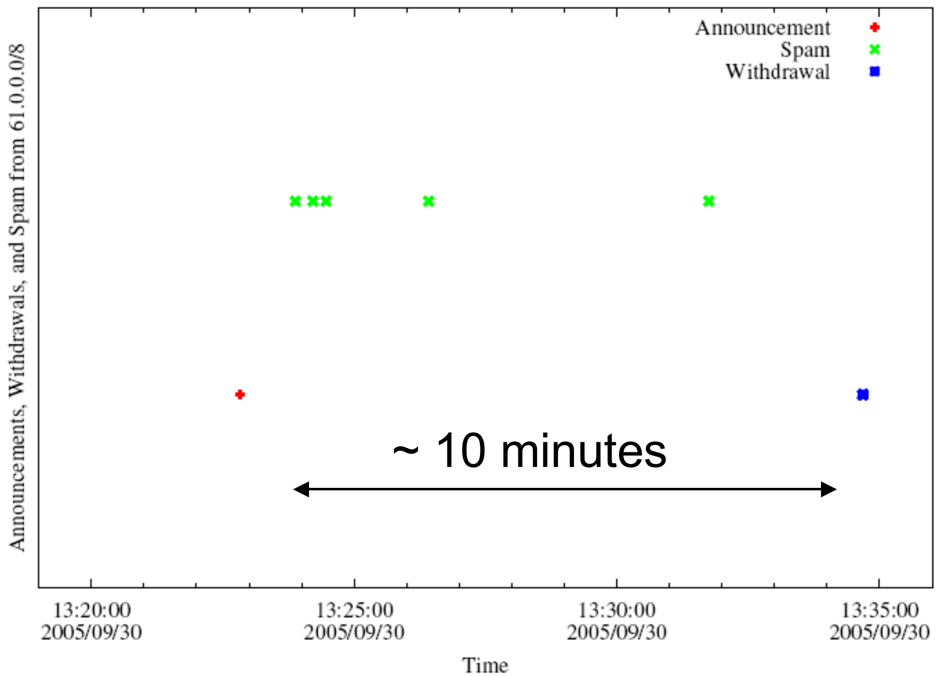
Resource Exhaustion: Spam

- Unsolicited commercial email
- As of about February 2005, estimates indicate that about 90% of all email is spam
- Common spam filtering techniques
  - Content-based filters
  - DNS Blacklist (DNSBL) lookups: Significant fraction of today's DNS traffic!

Can IP addresses from which spam is received be spoofed?

BGP Spectrum Agility

- Log IP addresses of SMTP relays
- Join with BGP route advertisements seen at network where spam trap is co-located.



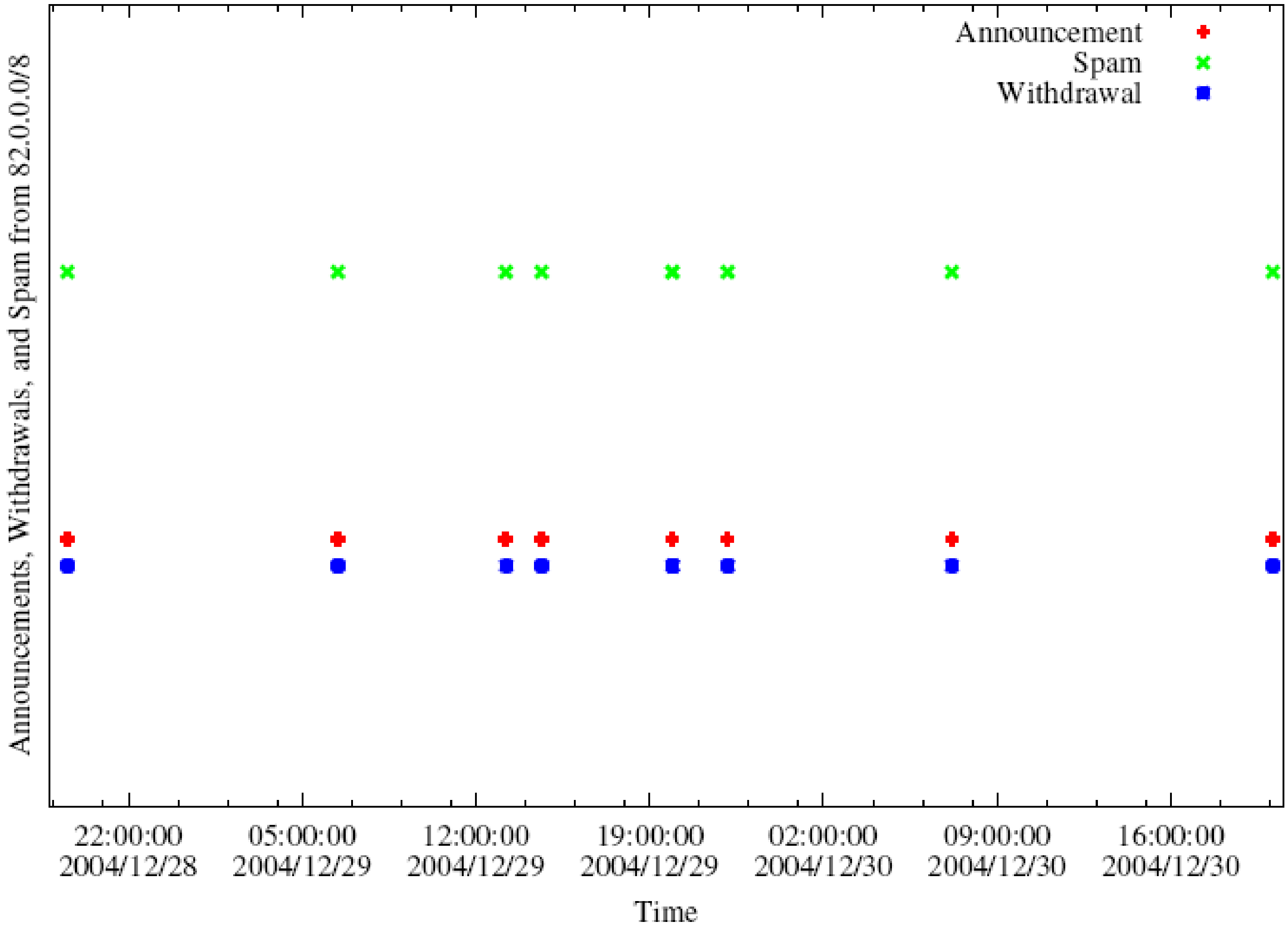
A small club of persistent players appears to be using this technique.

Common short-lived prefixes and ASes

61.0.0.0/8 4678
66.0.0.0/8 21562
82.0.0.0/8 8717

Somewhere between 1-10% of all spam (some clearly intentional, others might be flapping)

A Slightly Different Pattern



Why Such Big Prefixes?

- **Flexibility:** Client IPs can be scattered throughout dark space within a large /8
  - Same sender usually returns with different IP addresses

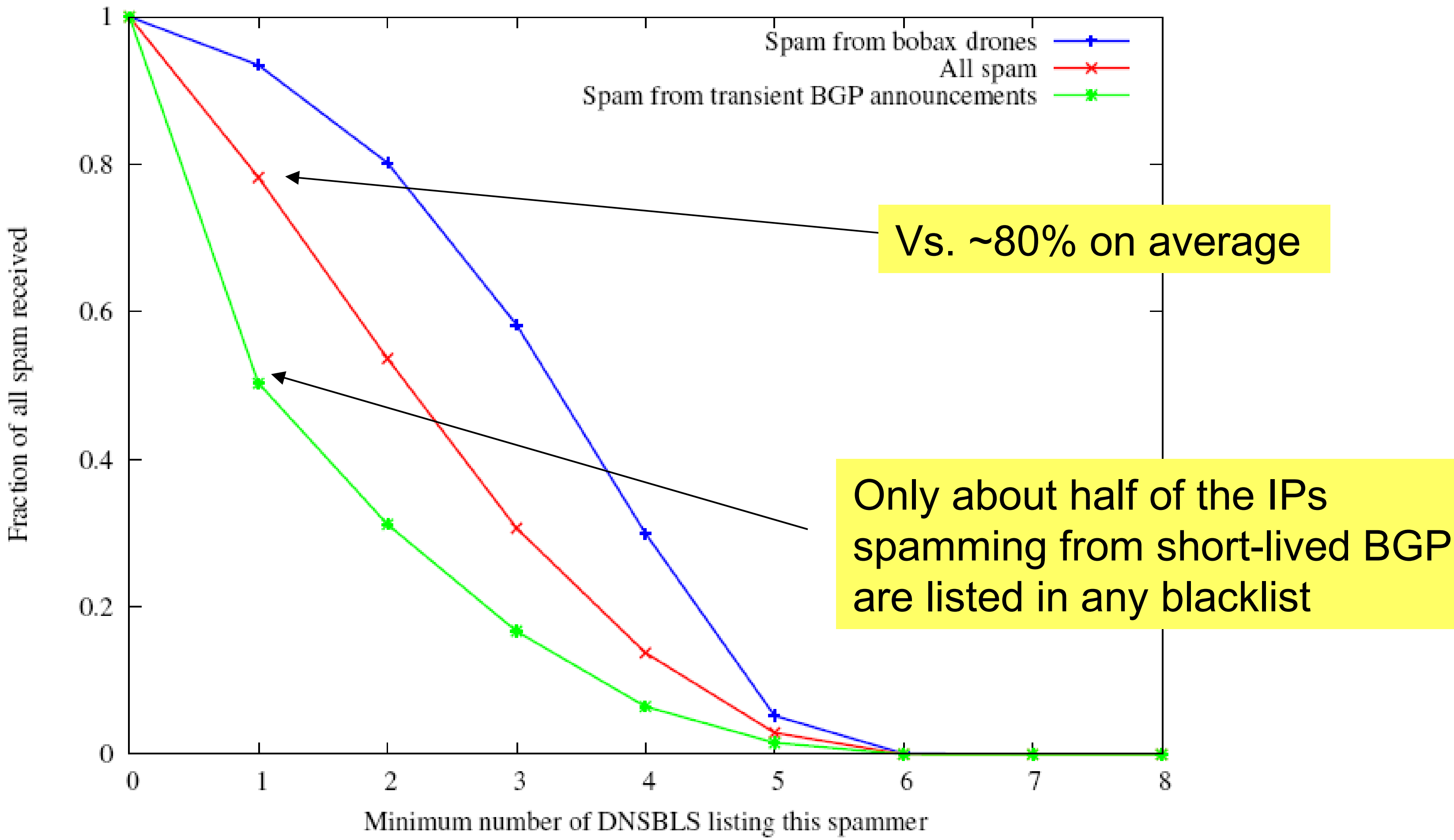
- **Visibility:** Route typically won’t be filtered (nice and short)

Characteristics of IP-Agile Senders

- IP addresses are widely distributed across the /8 space
- IP addresses typically appear only once at our sinkhole
- Depending on which /8, 60-80% of these IP addresses were not reachable by traceroute when we spot-checked
- Some IP addresses were in *allocated*, albeing unannounced space
- Some AS paths associated with the routes contained reserved AS numbers

Some evidence that it's working

Spam from IP-agile senders tend to be listed in fewer blacklists

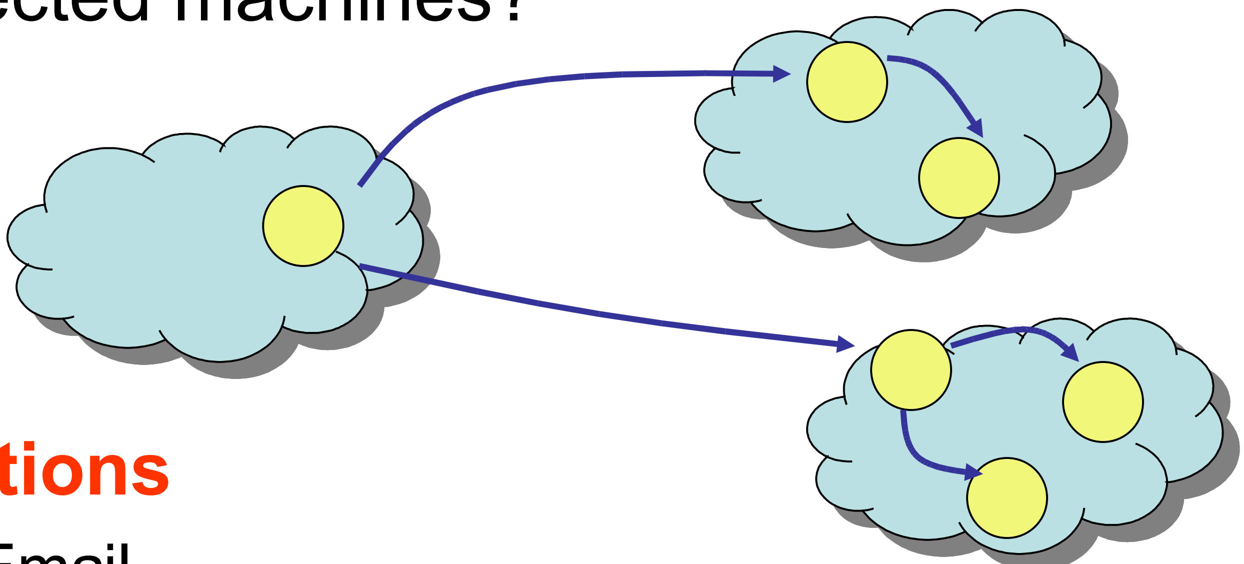


Botnets

- **Bots:** Autonomous programs performing tasks
- Plenty of “benign” bots
  - *e.g.*, weatherbug
- **Botnets:** group of bots
  - Typically carries malicious connotation
  - Large numbers of infected machines
  - Machines “enlisted” with infection vectors like worms (last lecture)
- Available for **simultaneous control** by a master
- *Size:* up to 350,000 nodes (from today’s paper)

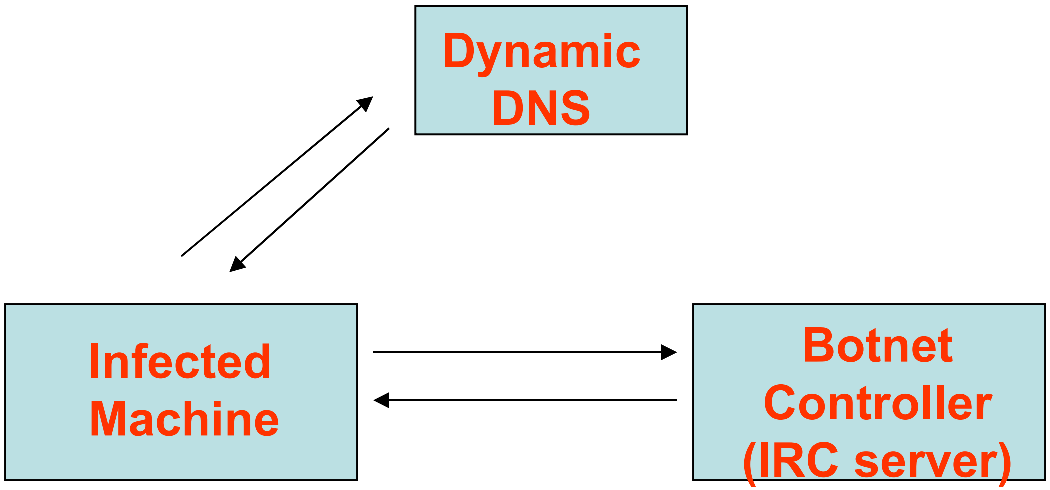
“Rallying” the Botnet

- Easy to combine worm, backdoor functionality
- **Problem:** how to learn about successfully infected machines?



- **Options**
  - Email
  - Hard-coded email address

Botnet Control



- Botnet master typically runs some IRC server on a well-known port (*e.g.*, 6667)
- Infected machine contacts botnet with pre-programmed DNS name (*e.g.*, big-bot.de)
- **Dynamic DNS:** allows controller to move about freely

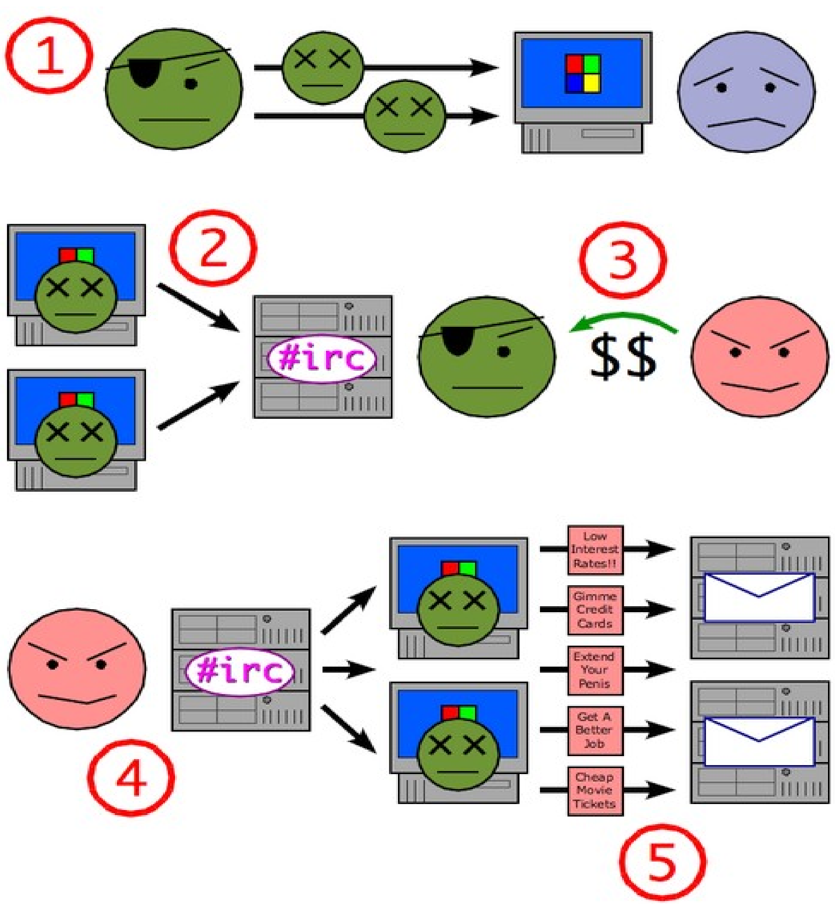
Botnet History: How we got here

- **Early 1990s:** IRC bots
  - eggdrop: automated management of IRC channels
- **1999-2000:** DDoS tools
  - Trinoo, TFN2k, Stacheldraht
- **1998-2000:** Trojans
  - BackOrifice, BackOrifice2k, SubSeven
- **2001- :** Worms
  - Code Red, Blaster, Sasser

Fast spreading capabilities pose big threat

Put these pieces together and add a controller…

Putting it together



1. Miscreant (botherd) launches worm, virus, or other mechanism to infect Windows machine.

3. Infected machines contact botnet controller via IRC.

5. Spammer (sponsor) pays miscreant for use of botnet.

7. Spammer uses botnet to send spam emails.

Botnet Detection and Tracking

- Network Intrusion Detection Systems (*e.g.,* Snort)
  - **Signature:** alert tcp any any -> any any (msg:"Agobot/Phatbot Infection Successful"; flow:established; content:"221
- **Honeynets:** gather information
  - Run unpatched version of Windows
  - Usually infected within 10 minutes
  - **Capture binary**
    - determine scanning patterns, etc.
  - **Capture network traffic**
    - Locate identity of command and control, other bots, etc.

Detection: In-Protocol

- Snooping on IRC Servers
- Email (*e.g.,* CipherTrust ZombieMeter)
  - > 170k new zombies per day
  - 15% from China
- Managed network sensing and anti-virus detection
  - Sinkholes detect scans, infected machines, etc.

- **Drawback:** Cannot detect botnet structure

Using DNS Traffic to Find Controllers

- Different types of queries may reveal info
  - Repetitive A queries may indicate bot/controller
  - MX queries may indicate spam bot
  - PTR queries may indicate a server
- Usually 3 level: hostname.subdomain.TLD
- Names and subdomains that just look rogue
  - (*e.g.*, irc.big-bot.de)

DNS Monitoring

- Command-and-control hijack
  - **Advantages:** accurate estimation of bot population
  - **Disadvantages:** bot is rendered useless; can’t monitor activity from command and control
- Complete TCP three-way handshakes
  - Can distinguish distinct infections
  - Can distinguish infected bots from port scans, etc.

Traffic Monitoring

- Goal: Recover communication structure
  - “Who’s talking to whom”

- Tradeoff: Complete packet traces with partial view, or partial statistics with a more expansive view

New Trend: Social Engineering

- Bots frequently spread through AOL IM
  - A bot-infected computer is told to spread through AOL IM
  - It contacts all of the logged in buddies and sends them a link to a malicious web site
  - People get a link from a friend, click on it, and say “sure, open it” when asked

Early Botnets: AgoBot (2003)

- Drops a copy of itself as svchost.exe or syschk.exe
- Propagates via Grokster, Kazaa, etc.
- Also via Windows file shares

Botnet Operation

- General
  - Assign a new random nickname to the bot
  - Cause the bot to display its status
  - Cause the bot to display system information
  - Cause the bot to quit IRC and terminate itself
  - Change the nickname of the bot
  - Completely remove the bot from the system
  - Display the bot version or ID
  - Display the information about the bot
  - Make the bot execute a .EXE file
- IRC Commands
  - Cause the bot to display network information
  - Disconnect the bot from IRC
  - Make the bot change IRC modes
  - Make the bot change the server Cvars
  - Make the bot join an IRC channel
  - Make the bot part an IRC channel
  - Make the bot quit from IRC
  - Make the bot reconnect to IRC
- Redirection
  - Redirect a TCP port to another host
  - Redirect GRE traffic that results to proxy PPTP VPN connections
- DDoS Attacks
  - Redirect a TCP port to another host
  - Redirect GRE traffic that results to proxy PPTP VPN connections
- Information theft
  - Steal CD keys of popular games
- Program termination

PhatBot (2004)

- Direct descendent of AgoBot
- More features
  - Harvesting of email addresses via Web and local machine
  - Steal AOL logins/passwords
  - Sniff network traffic for passwords
- Control vector is peer-to-peer (not IRC)

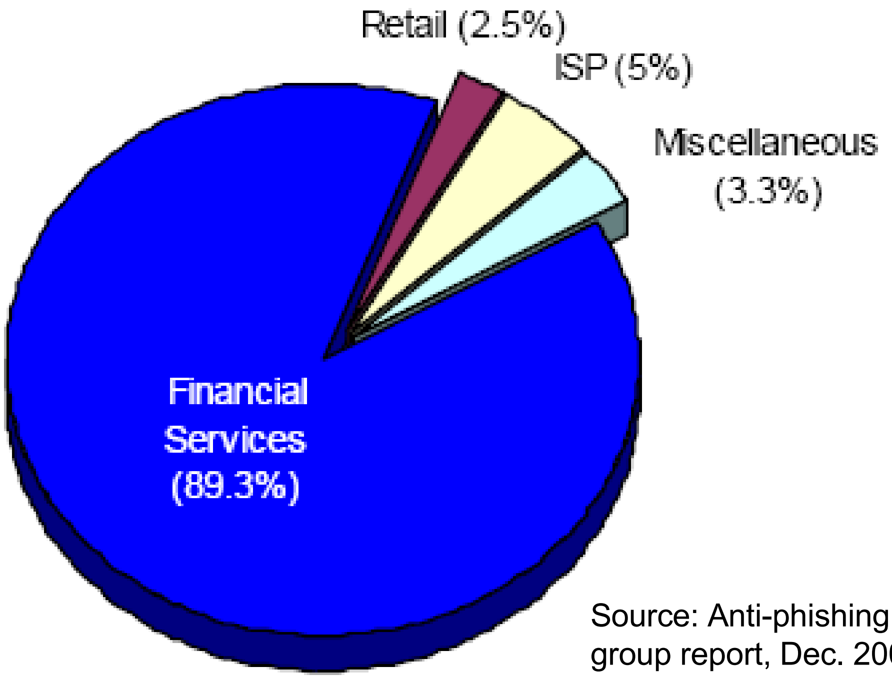
Botnet Application: Phishing

"Phishing attacks use both **social engineering** and **technical subterfuge** to steal consumers' personal identity data and financial account credentials." -- Anti-spam working group

- Social-engineering schemes
  - Spoofed emails direct users to counterfeit web sites
  - Trick recipients into divulging financial, personal data
- Anti-Phishing Working Group Report (Oct. 2005)
  - 15,820 phishing e-mail messages 4367 unique phishing sites identified.
  - 96 brand names were hijacked.
  - Average time a site stayed on-line was 5.5 days.

Question: What does phishing have to do with botnets?

Which web sites are being phished?



Source: Anti-phishing working group report, Dec. 2005

- Financial services by far the most targeted sites

***New trend:* Keystroke logging…**

Botnet Application: Click Fraud

- Pay-per-click advertising
  - **Publishers** display links from **advertisers**
  - **Advertising networks** act as middlemen
    - Sometimes the same as publishers (*e.g.*, Google)

- **Click fraud:** botnets used to click on pay-per-click ads

- **Motivation**
  - Competition between advertisers
  - Revenue generation by bogus content provider

Open Research Questions

- Botnet membership detection
  - Existing techniques
    - Require special privileges
    - Disable the botnet operation
  - Under various datasets (packet traces, various numbers of vantage points, etc.)
- Click fraud detection
- Phishing detection