CS 3700
Networks and Distributed Systems

Lecture 20: Malware/Botnets

Slides stolen from Vern Paxson (ICSI) and Stefan Savage (UCSD)

Motivation

- Internet currently used for important services
  - Financial transactions, medical records
- Increasingly used for critical services
  - 911, surgical operations, water/electrical system control, remote controlled drones, etc.
- Networks more open than ever before
  - Global, ubiquitous Internet, wireless

Malicious Users

- Miscreants, e.g. LulzSec
  - In it for thrills, street cred, or just to learn
  - Defacing web pages, spreading viruses, etc.
- Hacktivists, e.g. Anonymous
  - Online political protests
  - Stealing and revealing classified information
- Organized Crime
  - Profit driven, online criminals
  - Well organized, divisions of labor, highly motivated

Network Security Problems

- Host Compromise
  - Attacker gains control of a host
  - Can then be used to try and compromise others
- Denial-of-Service
  - Attacker prevents legitimate users from gaining service
- Attack can be both
  - E.g., host compromise that provides resources for denial-of-service

Definitions

- Virus
  - Program that attaches itself to another program
- Worm
  - Replicates itself over the network
  - Usually relies on remote exploit (e.g. buffer overflow)
- Rootkit
  - Program that infects the operating system (or even lower)
  - Used for privilege elevation, and to hide files/processes
- Trojan horse
  - Program that opens “back doors” on an infected host
  - Gives the attacker remote access to machines
- Botnet
  - A large group of Trojaned machines, controlled en-mass
  - Used for sending spam, DDoS, click-fraud, etc.

Host Compromise

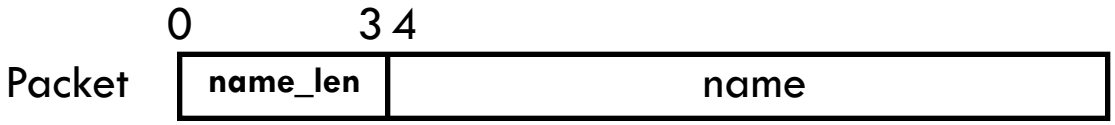
- One of earliest major Internet security incidents
  - Internet Worm (1988): compromised almost every BSD-derived machine on Internet
- Today: estimated that a single worm could compromise 10M hosts in < 5 min
- Attacker gains control of a host
  - Read data
  - Erase data
  - Compromise another host
  - Launch denial-of-service attacks on another host

Host Compromise: Stack Overflow

- Typical code has many bugs because those bugs are not triggered by common input
- Network code is vulnerable because it accepts input from the network
- Network code that runs with high privileges (i.e., as root) is especially dangerous
  - E.g., web server

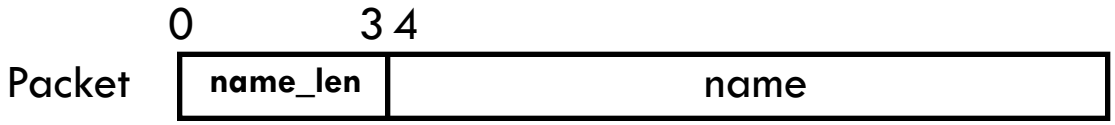
Example

- What is wrong with this code?

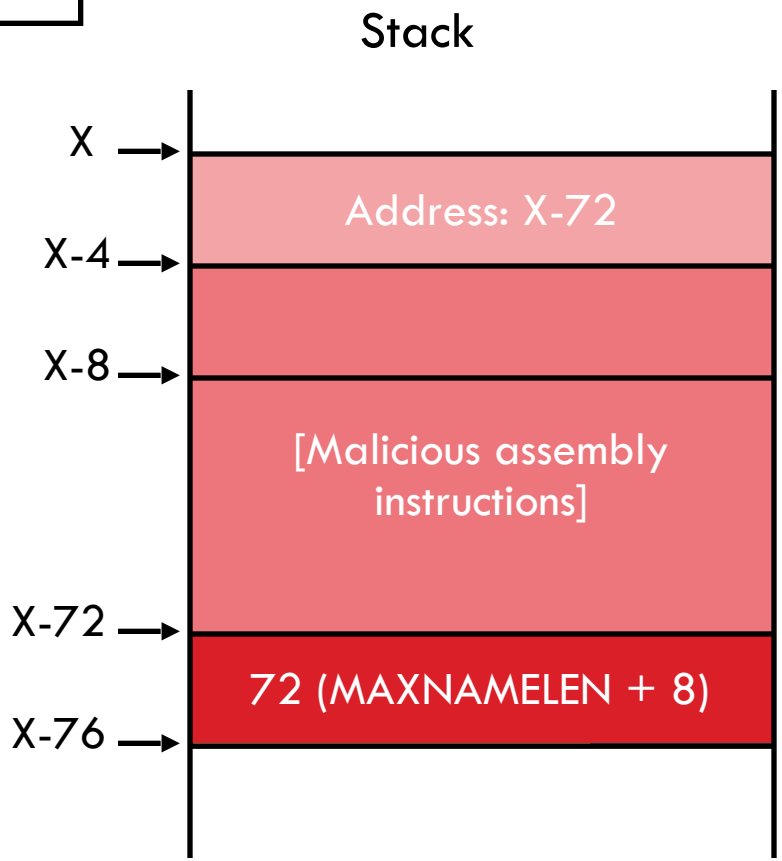



```
// Copy a variable length user name from a packet
#define MAXNAMELEN 64
int offset = OFFSET_USERNAME;
char username[MAXNAMELEN];
int name_len;

name_len = packet[offset];
memcpy(&username, packet[offset + 1], name_len);
```

Example



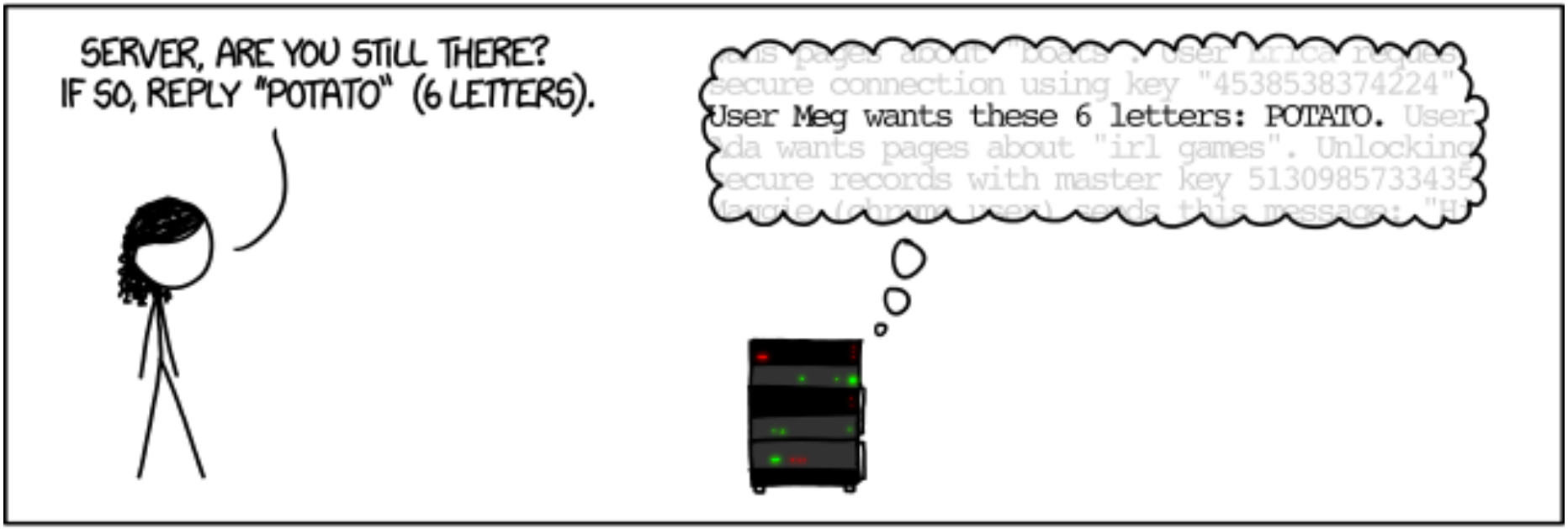

```
void foo(packet) {
  #define MAXNAMELEN 64
  int offset = OFFSET_USERNAME;
  char username[MAXNAMELEN];
  int name_len;

  name_len = packet[offset];
  memcpy(&username,
         packet[offset + 1],name_len);
  …
}
```



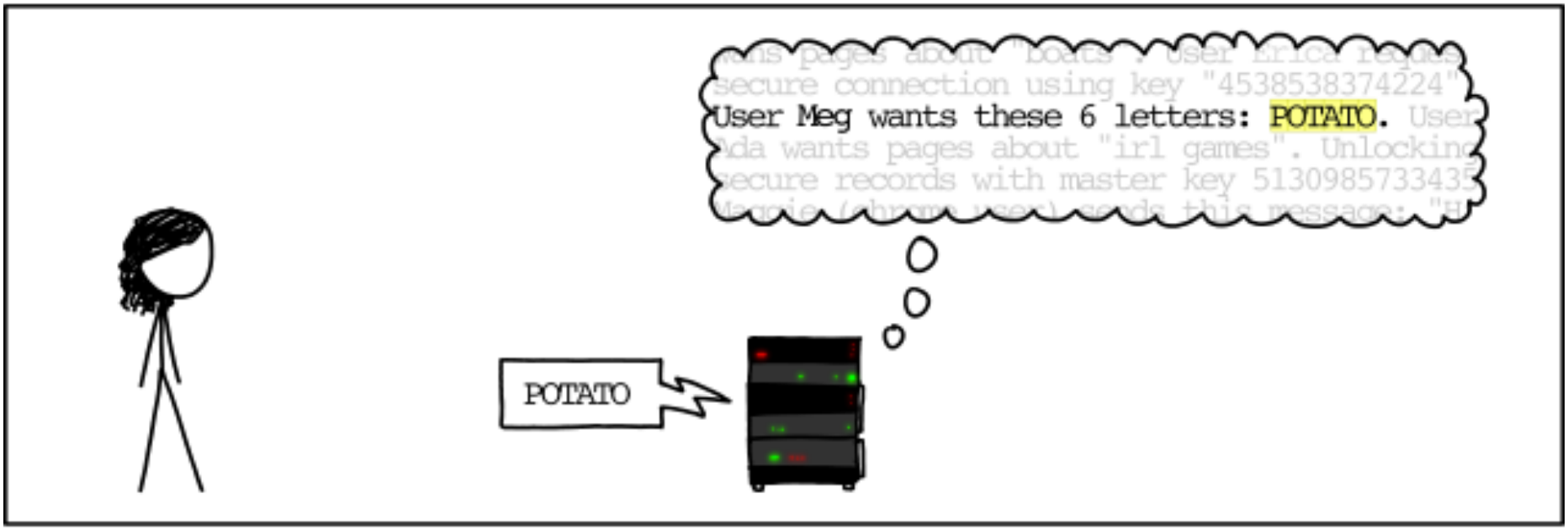
Heartbleed Attack

- Vulnerability in OpenSSL
  - Used by HTTPS, SSH, many others to encrypt communication
- Heartbeat attack
  - Message of form: “Here’s some data, echo it back to me”
  - Takes as input: Data and length (L), where L <= 64KB
  - Echoes back a block of data L
  - What’s the problem?
- Send one byte, get 64KB of RAM!
  - Private keys, passwords, etc have been leaked

As described by XKCD

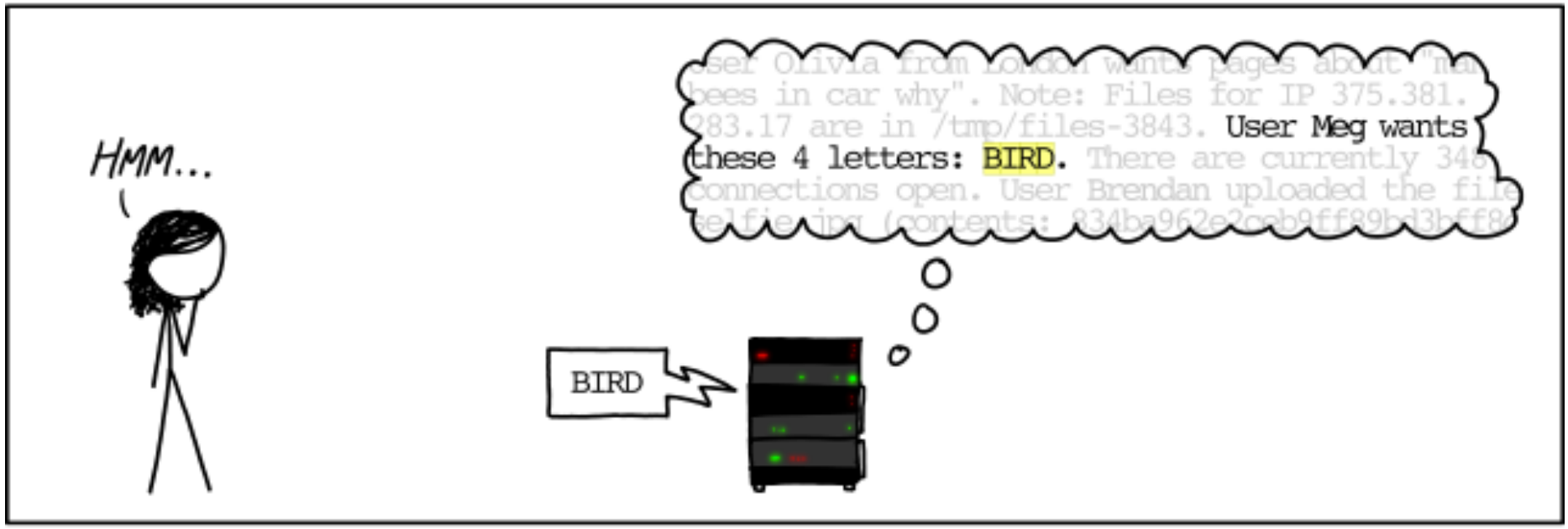


As described by XKCD

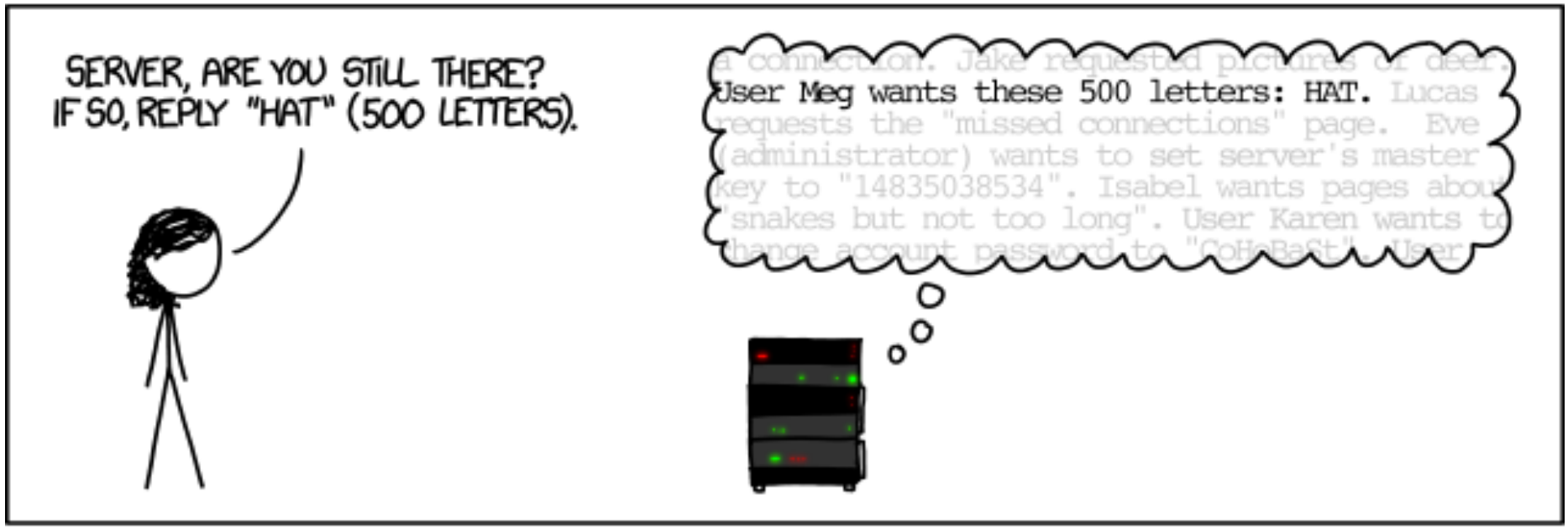


As described by XKCD

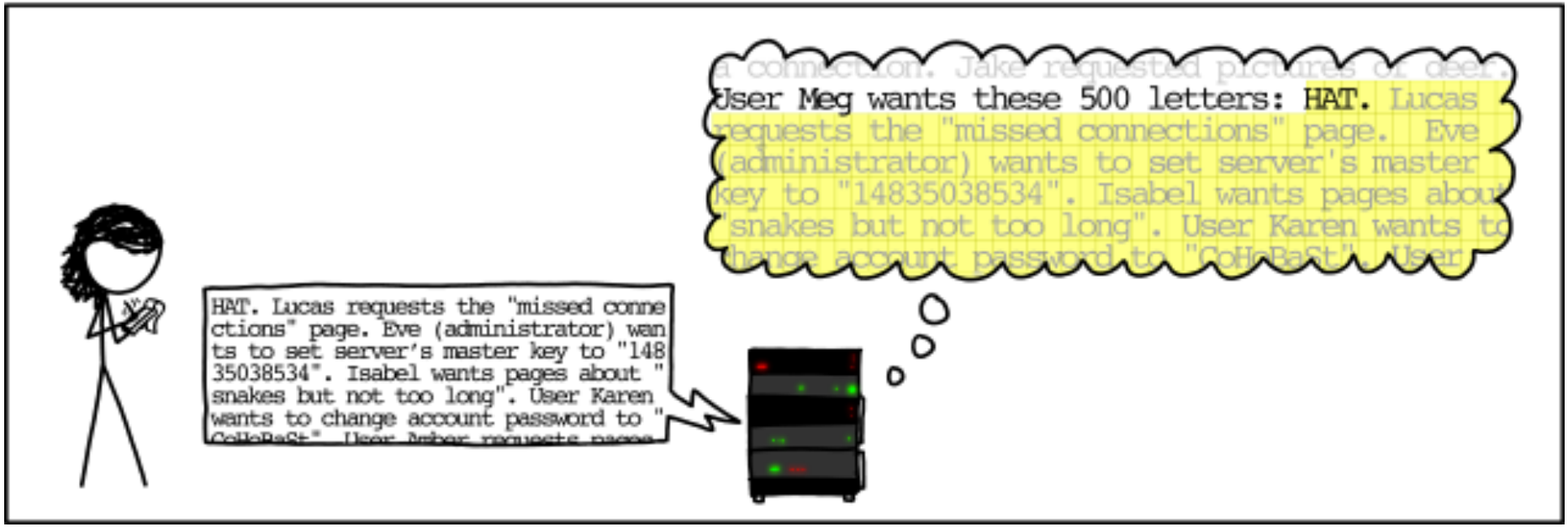
As described by XKCD



As described by XKCD



As described by XKCD



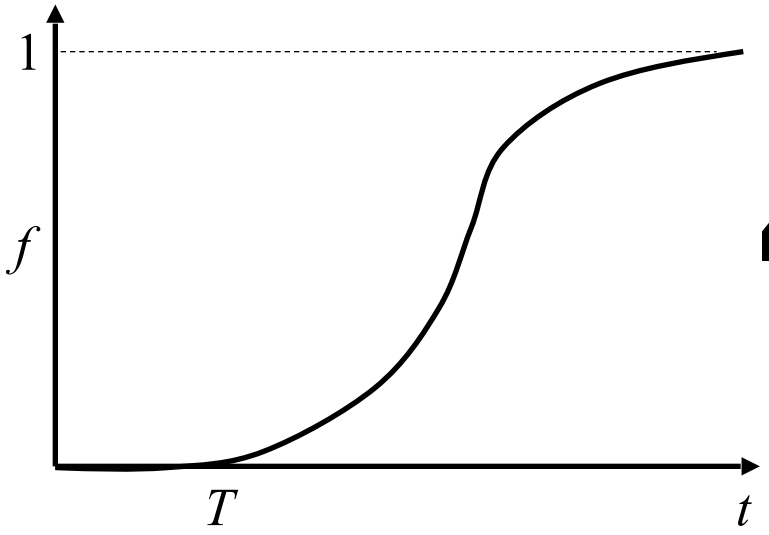
Effect of Stack Overflow

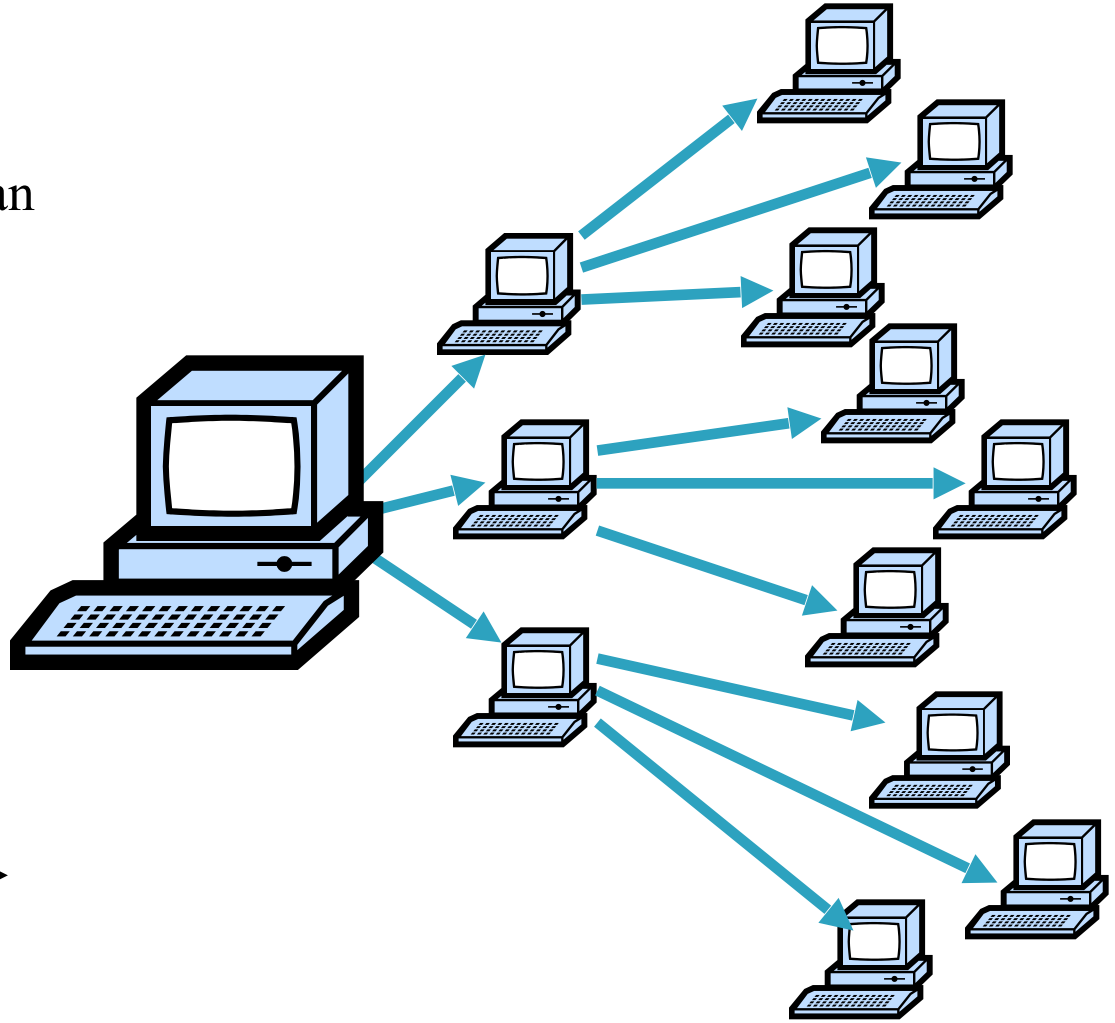
- Write into part of the stack or heap
  - Write arbitrary code to part of memory
  - Cause program execution to jump to arbitrary code
- Worm
  - Probes host for vulnerable software
  - Sends bogus input
  - Attacker can do anything that the privileges of the buggy program allows
    - Launches copy of itself on compromised host
  - Spread at exponential rate
  - 10M hosts in < 5 minutes

Worm Spreading

$f = (e^{K(t-T)} - 1) / (1 + e^{K(t-T)})$

- f – fraction of hosts infected
- K – rate at which one host can compromise others
- T – start time of the attack





Worm Examples

- Morris worm (1988)
- Code Red (2001)
- MS Slammer (January 2003)
- MS Blaster (August 2003)

Morris Worm (1988)

- Infect multiple types of machines (Sun 3 and VAX)
  - Spread using a Sendmail bug
- Attack multiple security holes including
  - Buffer overflow in fingerd
  - Debugging routines in Sendmail
  - Password cracking
- Intend to be benign but it had a bug
  - Fixed chance the worm wouldn't quit when reinfecting a machine → number of worm on a host built up rendering the machine unusable

Code Red Worm (2001)

- Attempts to connect to TCP port 80 on a randomly chosen host
- If successful, the attacking host sends a crafted HTTP GET request to the victim, attempting to exploit a buffer overflow
- Worm "bug": all copies of the worm use the same random seed to scanning new hosts
  - DoS attack on those hosts
  - Slow to infect new hosts
- 2nd generation of Code Red fixed the bug!
  - It spread much faster

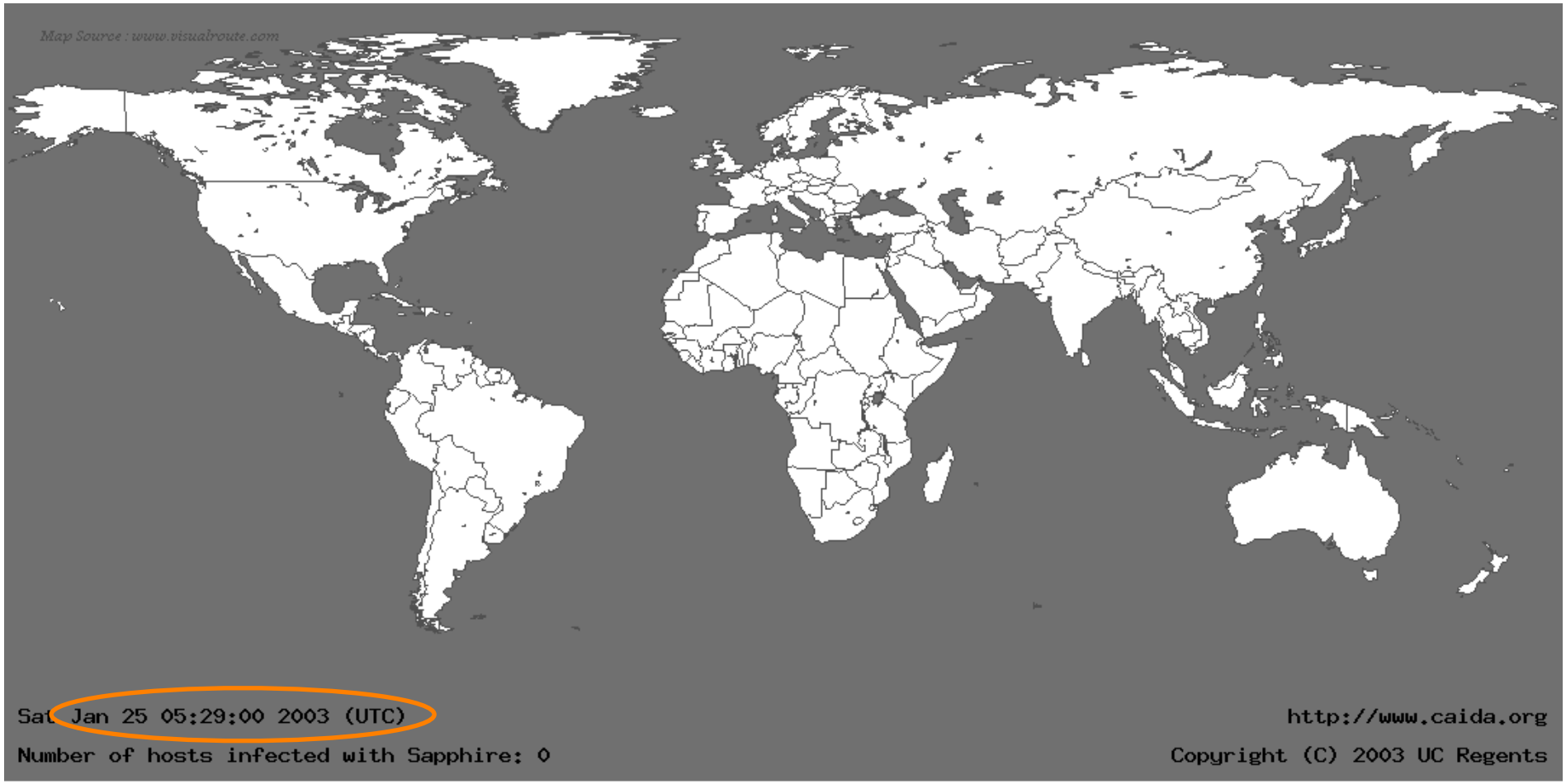
MS SQL Slammer (January 2003)

- Uses UDP port 1434 to exploit a buffer overflow in MS SQL server
  - Generate massive amounts of network packets
  - Brought down as many as 5 of the 13 internet root name servers
- Stealth Feature
  - The worm only spreads as an in-memory process: it never writes itself to the hard drive
    - Solution: close UDP port on firewall and reboot

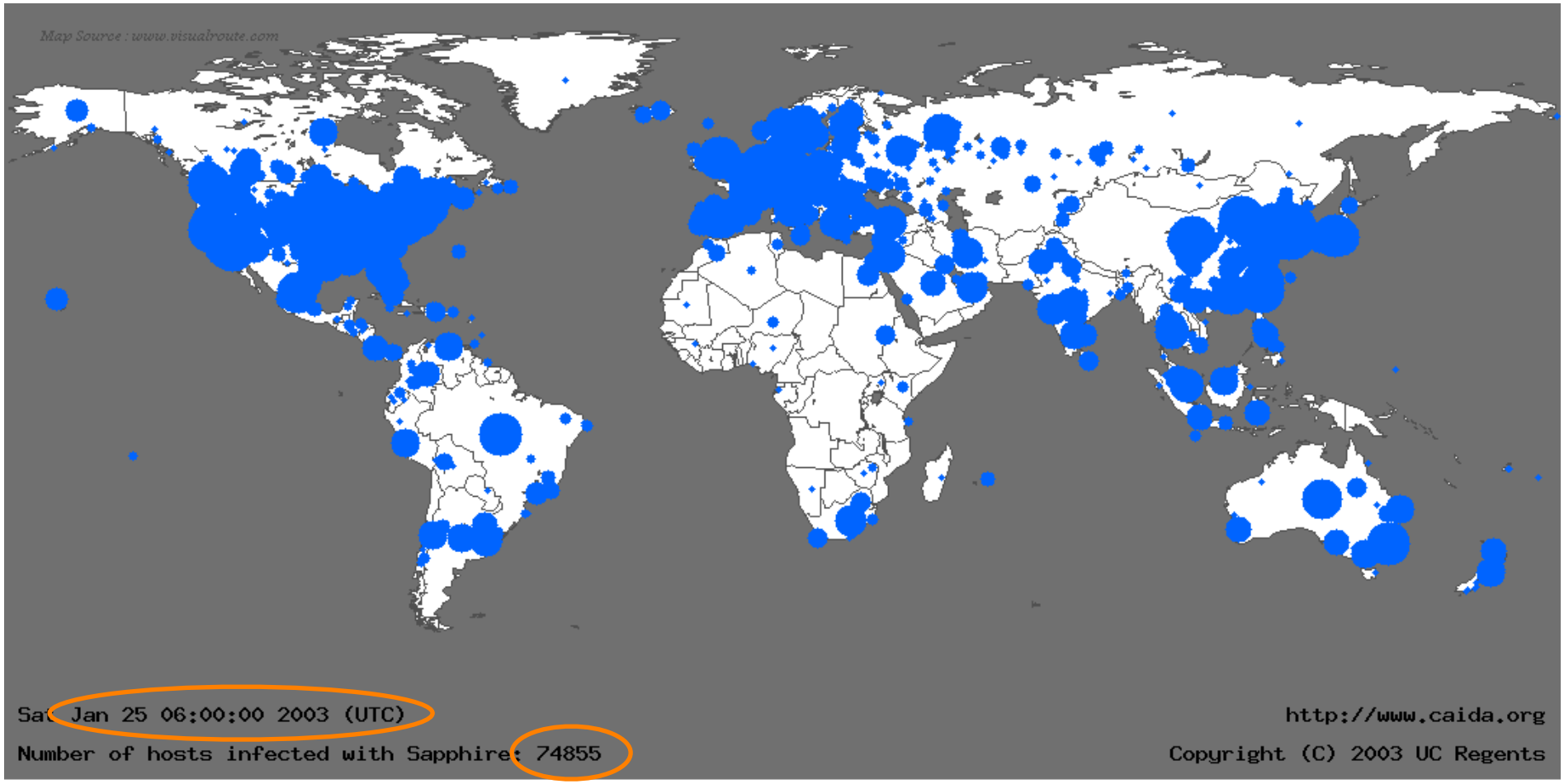
MS SQL Slammer (January 2003)

- Slammer exploited a connectionless UDP service, rather than connection-oriented TCP.
  - Entire worm fit in a single packet!
  - When scanning, worm could “fire and forget”.
- Worm infected 75,000+ hosts in 10 minutes (despite broken random number generator).
  - At its peak, doubled every 8.5 seconds
- Progress limited by the Internet’s carrying capacity!

Life Just Before Slammer



Life Just After Slammer



MS Blaster (August 2003)

- Exploits a buffer overflow vulnerability of the RPC (Remote Procedure Call) service in Win 200 and XP
- Scans a random IP range to look for vulnerable systems on TCP port 135
- Opens TCP port 4444, which could allow an attacker to execute commands on the system
- DDoS windowsupdate.com on certain versions of Windows

Spreading Faster

- Idea 1: *Reduce Redundant Scanning*
  - Construct permutation of address space.
  - Each new worm instance starts at random point
  - Worm instance that "encounters" another instance re-randomizes
- Idea 2: *Reduce Slow Startup Phase*
  - Construct a "hit-list" of vulnerable servers in advance
  - Assume 1M vulnerable hosts, 10K hit-list, 100 scans/worm/sec, 1 sec to infect
    - 99% infection rate in 5 minutes

Spreading Even Faster — Flash Worms

- Idea: use an *Internet-sized* hit list.
  - Initial copy of the worm has the entire hit list
  - Each generation…
    - Infect *n* hosts from the list
    - Give each new infection $1/n$ of the list
  - Need to engineer for locality, failure & redundancy
  - ~10 seconds to infect the whole Internet

Contagion worms

- Suppose you have two exploits: *Es* (Web server) and *Ec* (Web client)
- You infect a server (or client) with *Es* (*Ec*)
- Then you . . . wait (Perhaps you bait, e.g., host porn)
- When vulnerable client arrives, infect it
- You send over both *Es* and *Ec*
- As client happens to visit other vulnerable servers, infect

Incidental Damage … Today

- Today's worms have significant real-world impact:
  - Code Red disrupted routing
  - Slammer disrupted root DNS, elections, ATMs, airlines, operations at an off-line nuclear power plant …
  - Blaster possibly contributed to Great Blackout of Aug. 2003 … ?
  - Plus major clean-up costs
- But most worms are amateurish
  - Unimaginative payloads

Where are the Nastier Worms??

- Botched propagation the norm
- Doesn't anyone read the literature?
  - e.g. permutation scanning, flash worms, metaserver worms, topological, contagion
- Botched payloads the norm
  - e.g. Flooding-attack fizzles
- Some worm authors are in it for kicks …
  - No arms race.

Next-Generation Worm Authors

- Military (e.g. Stuxnet)
  - Worm spread in 2010 (courtesy of US/Israel)
  - Targets Siemens industrial (SCADA) systems
  - Target: Iranian uranium enrichment infrastructure
- Crooks:
  - Very worrisome onset of blended threats
    - Worms + viruses + spamming + phishing + DOS-for-hire + botnets + spyware
  - Money on the table → **arms race**
    - (market price for spam proxies: 3-10¢/host/week)

Witty

- Released March 19, 2004
- Single UDP packet exploits flaw in the passive analysis of Internet Security Systems products
- “Bandwidth-limited” UDP worm ala’ Slammer
- Vulnerable pop. (12K) attained in 75 minutes
- Payload: slowly corrupt random disk blocks

Witty, con't

- Flaw had been announced the *previous day*
- Telescope analysis reveals:
  - Initial spread seeded via a hit-list
  - In fact, targeted a U.S. military base
  - Analysis also reveals "Patient Zero", a European retail ISP
- Written by a Pro

Shamoon

- Found August 16, 2012
- Targeted computers from Saudi Aramco
  - Largest company/oil producer in the world
- Infected 30,000 desktop machines
  - Took one week to clean and restore
- Could have been much worse
  - Attack was not stealthy
    - Stolen data slowly over time
    - Slowly corrupt random disk blocks, spreadsheets, etc.
  - Did not target SCADA or production control systems

Some Cheery Thoughts

- Imagine the following species:
  - Poor genetic diversity; heavily inbred
  - Lives in "hot zone"; thriving ecosystem of infectious pathogens
  - Instantaneous transmission of disease
  - Immune response 10-1M times slower
  - Poor hygiene practices
- What if diseases were…
  - Trivial to create
  - Highly profitable to create and spread

What would its long-term prognosis be?

Worms to Botnets

- Ultimate goal of most Internet worms
  - Compromise machine, install rootkit, then trojan
  - One of many in army of remote controlled machines
- Used by online criminals to make money
  - Extortion
    - "Pay use $100K or we will DDoS your website"
  - Spam and click-fraud
  - Phishing and theft of personal information
    - Credit card numbers, bank login information, etc.

Botnet Attacks

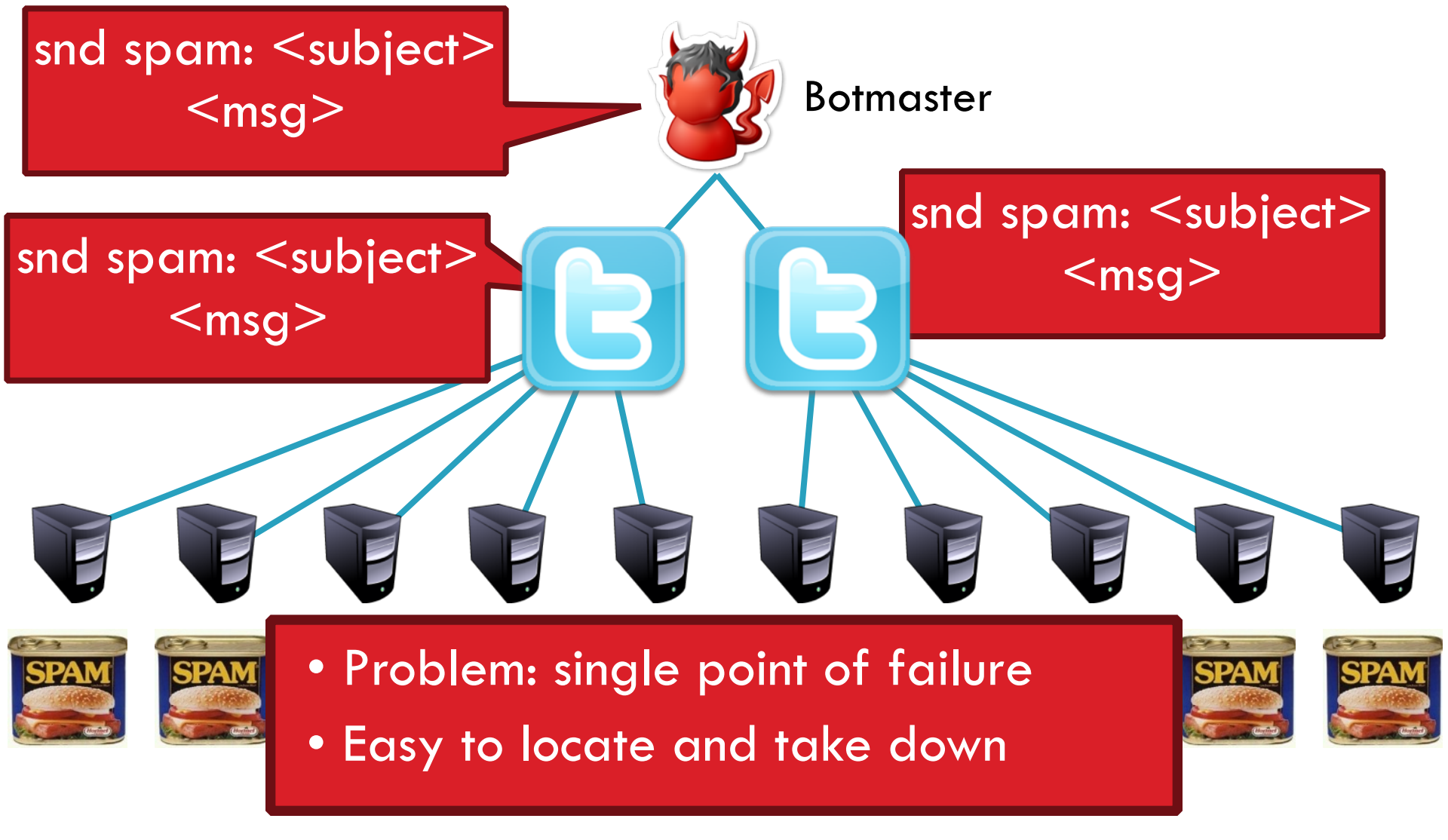
- Truly effective as an online weapon for terrorism
  - i.e. perform targeted attacks on governments and infrastructure
- Recent events: massive DoS on Estonia
  - April 27, 2007 – Mid-May, 2007
  - Closed off most government and business websites
  - Attack hosts from US, Canada, Brazil, Vietnam, …
  - Web posts indicate attacks controlled by Russians
  - All because Estonia moved a memorial of WWII soldier
- Is this a glimpse of the future?


1
2
OK
3
$
4

Detecting / Deterring Botnets

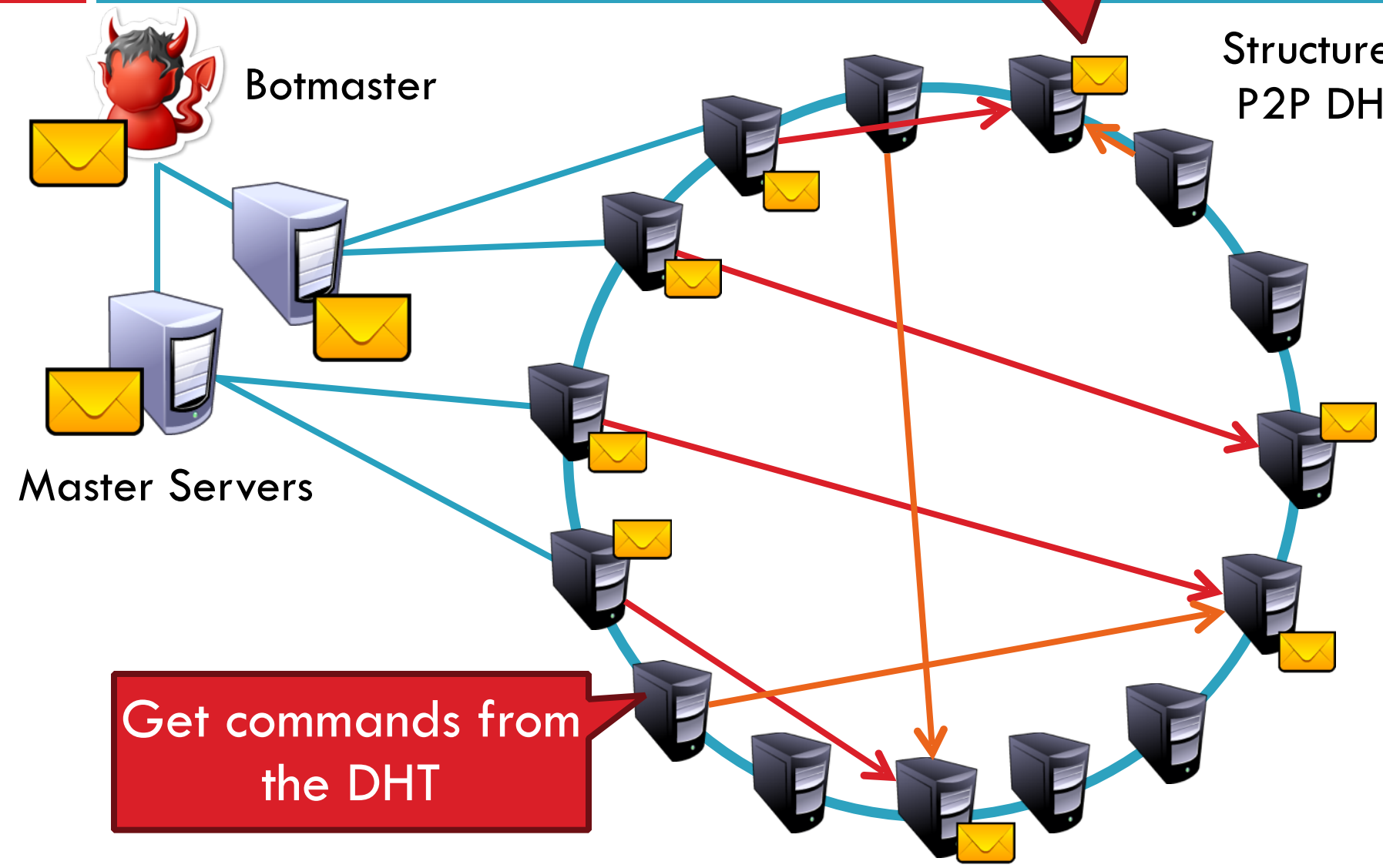
- Bots controlled via C&C channels
  - Potential weakness to disrupt botnet operation
  - Traditionally relied on IRC channels run by ephemeral servers
    - Can rotate single DNS name to different IPs on minute-basis
  - Can be found by mimicing bots (using honeypots)
- Bots also identified via DNS blacklist requests
- A constant cat and mouse game
  - Attackers evolving to decentralized C&C structures
  - Peer to peer model, encrypted traffic
  - Storm botnet, estimated 1-50 million members in 9/2007

Old-School C&C: IRC Channels

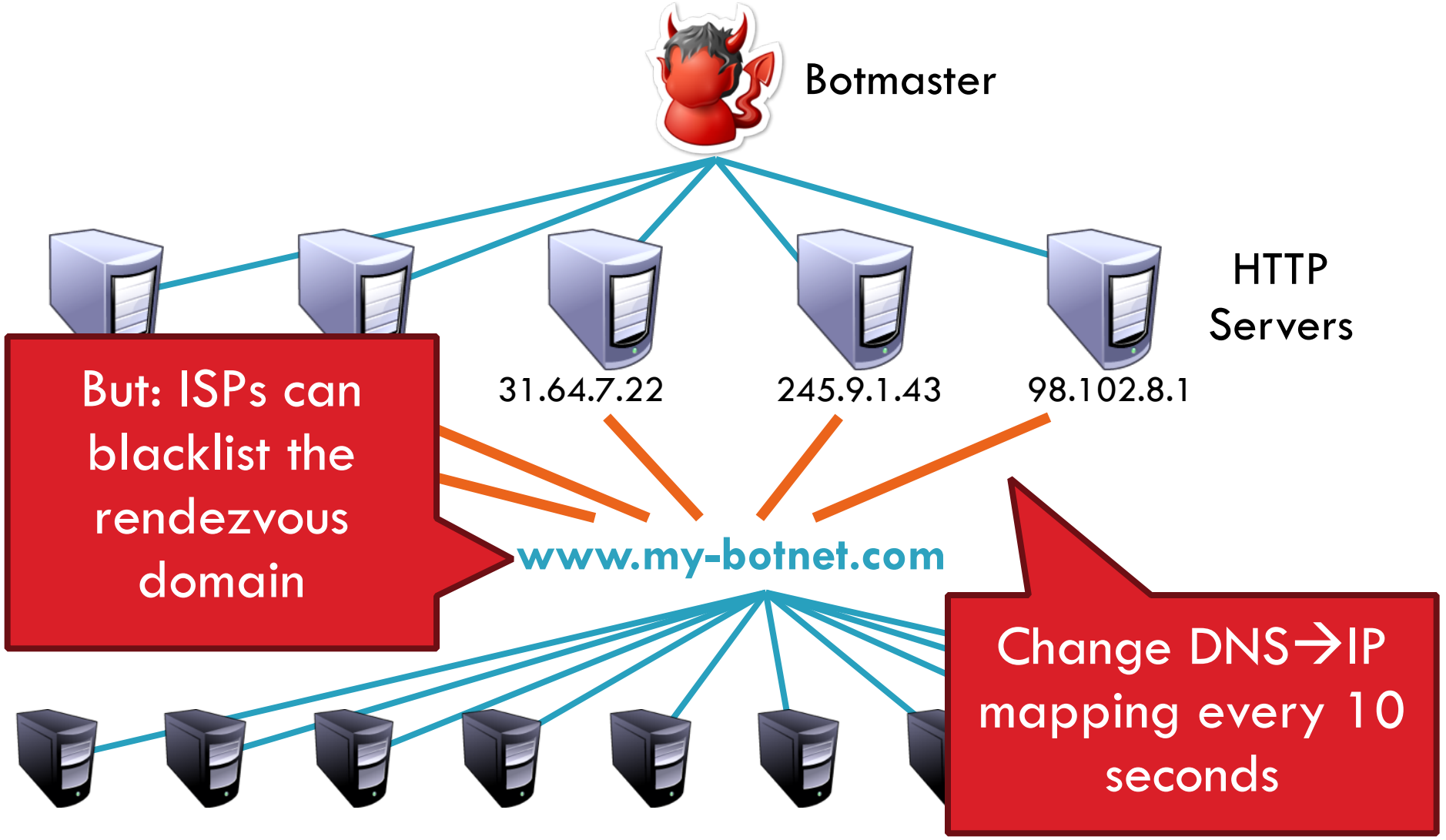


- Problem: single point of failure
- Easy to locate and take down

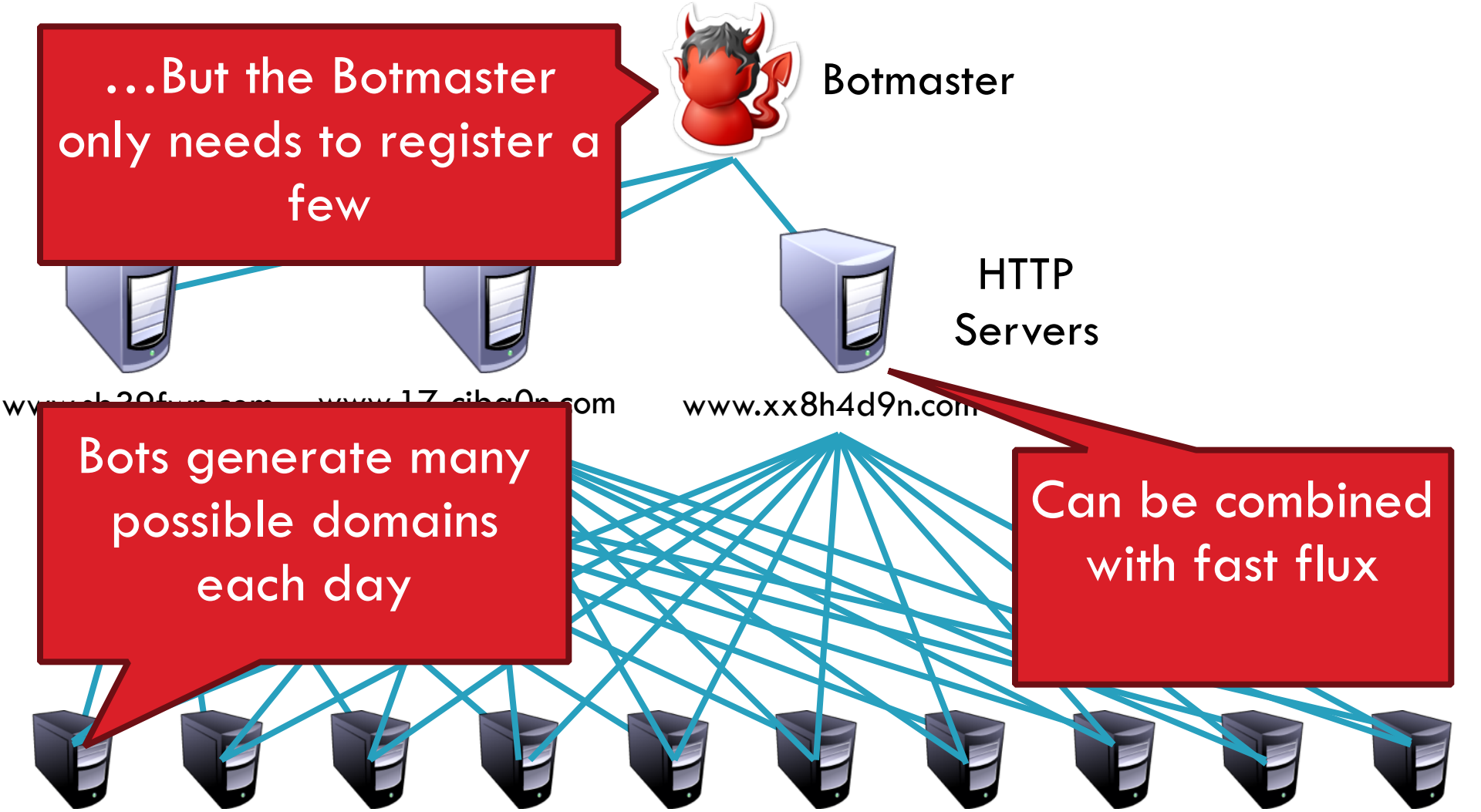
P2P Botnets


Insert commands into the DHT
Botmaster
Structured P2P DHT
Master Servers
Get commands from the DHT

Fast Flux DNS



Random Domain Generation



"Your Botnet is My Botnet"

- Takeover of the Torpig botnet
  - Random domain generation + fast flux
  - Team reverse engineered domain generation algorithm
  - Registered 30 days of domains before the botmaster!
  - Full control of the botnet for 10 days
- Goal of the botnet: theft and phishing
  - Steals credit card numbers, bank accounts, etc.
  - Researchers gathered all this data
- Other novel point: accurate estimation of botnet size

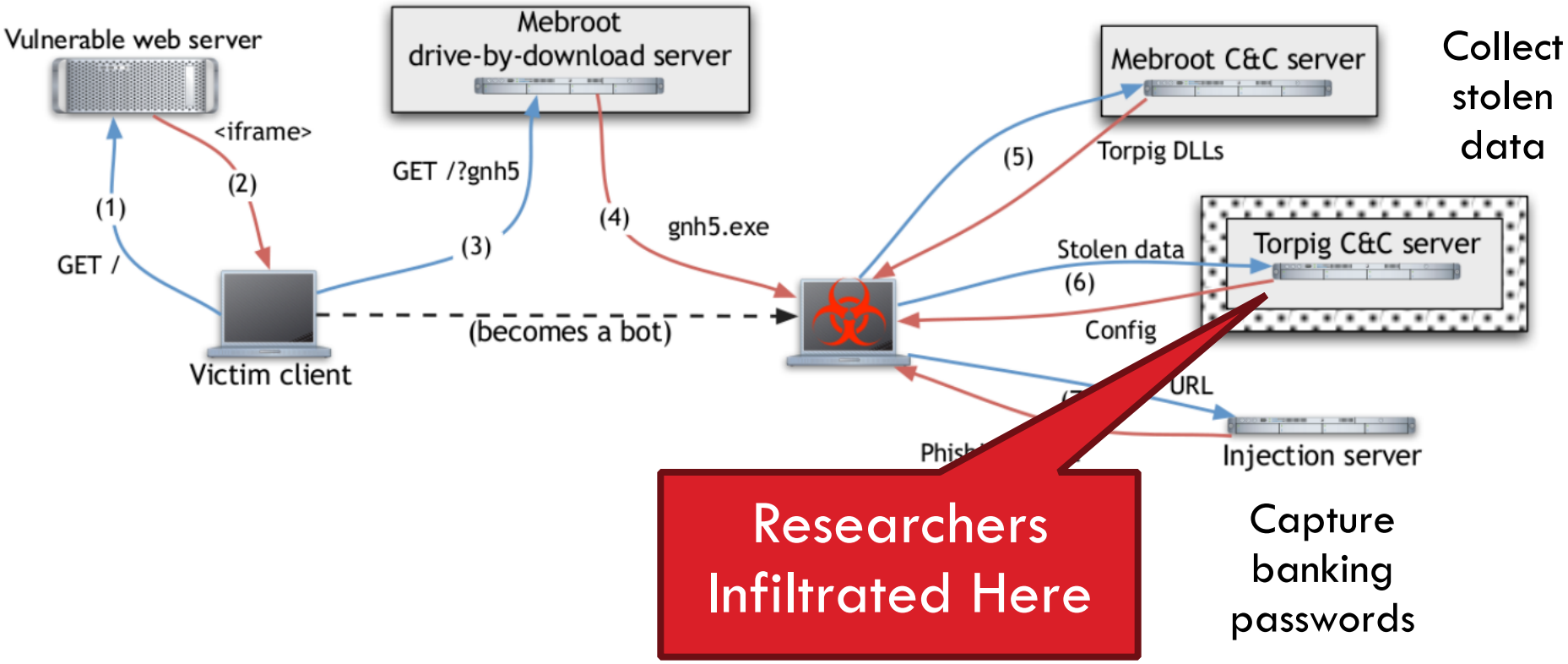
Torpig Architecture

Host gets infected via drive-by-download

Rootkit installation

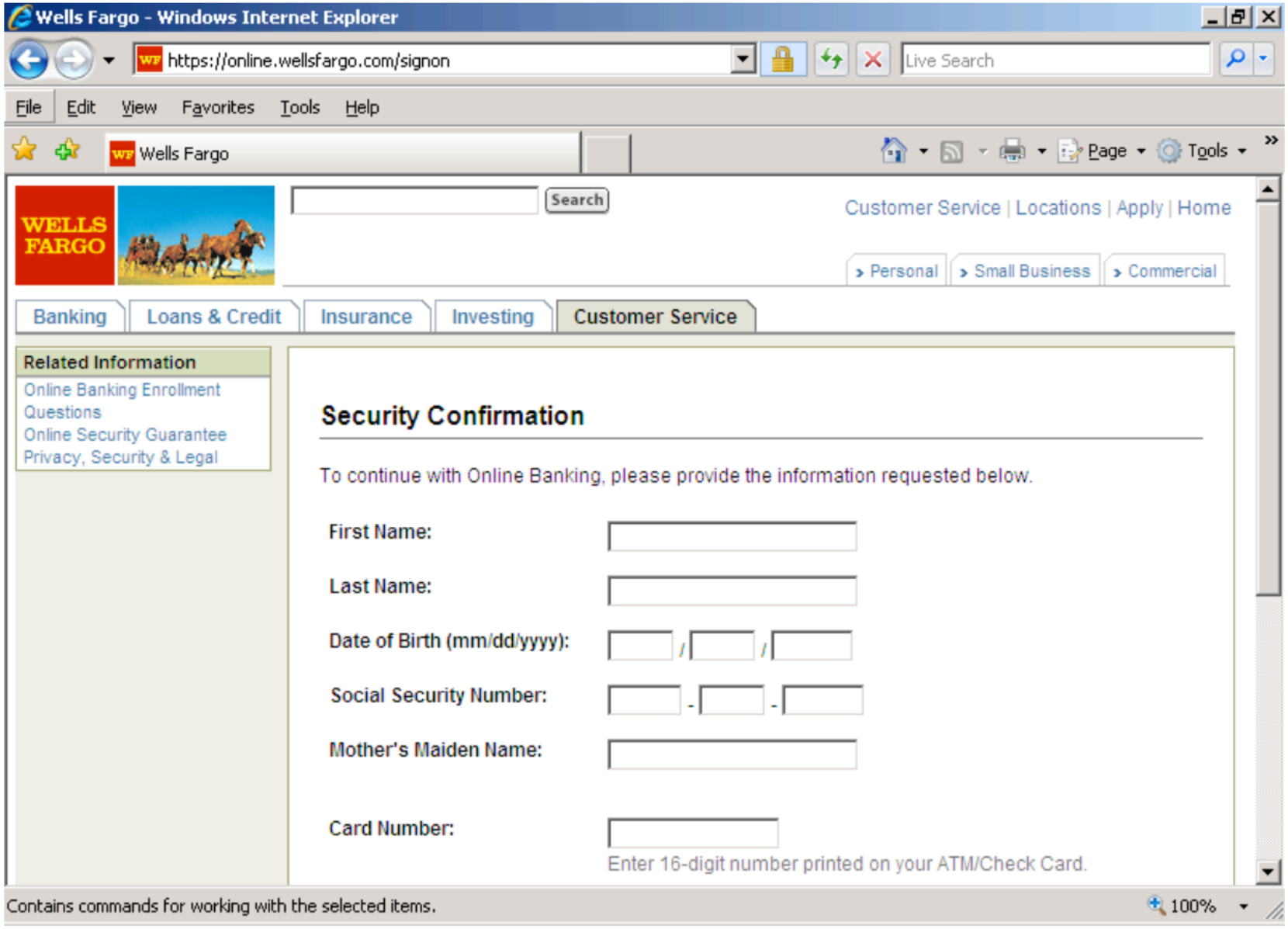
Trojan installation

Collect stolen data



Capture banking passwords

Man-in-the-Browser Attack



Stolen Information

- Data gathered from Jan 25-Feb 4 2009

User Accounts

Data Type	Data Items (#)
Mailbox account	54,090
Email	1,258,862
Form data	11,966,532
HTTP account	411,039
FTP account	12,307
POP account	415,206
SMTP account	100,472
Windows password	1,235,122

Banks Accounts

Country	Institutions (#)	Accounts (#)
US	60	4,287
IT	34	1,459
DE	122	641
ES	18	228
PL	14	102
Other	162	1,593
Total	410	8,310

- How much is this data worth?
  - Credit cards: $0.10-$25 Banks accounts: $10-$1000
  - $83K-$8.3M

How to Estimate Botnet Size?

- Passive data collection methodologies
  - Honeypots
    - Infect your own machines with Trojans
    - Observe network traffic
  - Look at DNS traffic
    - Domains linked to fast flux C&C
  - Networks flows
    - Analyze all packets from a large ISP and use heuristics to identify botnet traffic
- **None of these methods give a complete picture**

Size of the Torpig Botnet

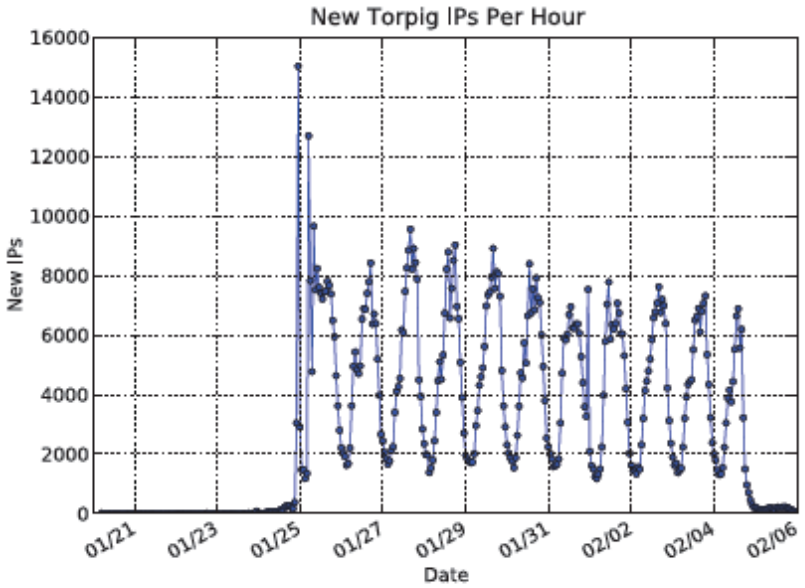


Figure 5: New unique IP addresses per hour.

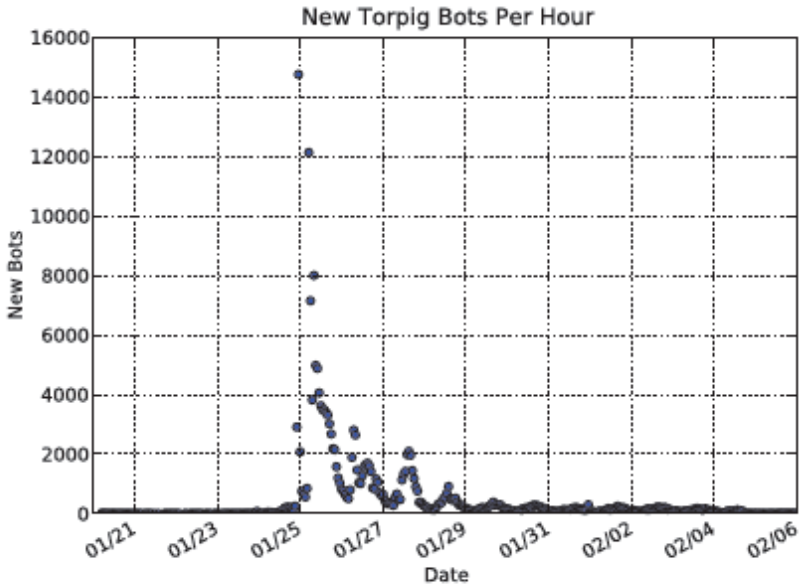


Figure 6: New bots per hour.

- Why the disconnect between IPs and bots?
  - Dynamic IPs, short DHCP leases
- Casts doubt on prior studies, enables more realistic estimates of botnet size