Worms and Bots

CS155

Elie Bursztein

Outline

- Worm Generation 1
- Botnet
- Fast Flux
- Worm Generation 2
- Underground Economy

Worms generation 1

Worm

- A worm is self-replicating software designed to spread through the network
  - Typically, exploit security flaws in widely used services
  - Can cause enormous damage
    - Launch DDOS attacks, install bot networks
    - Access sensitive information
    - Cause confusion by corrupting the sensitive information

Cost of worm attacks

- Morris worm, 1988
  - Infected approximately 6,000 machines
    - 10% of computers connected to the Internet
  - cost ~ $10 million in downtime and cleanup
- Code Red worm, July 16 2001
  - Direct descendant of Morris' worm
  - Infected more than 500,000 servers
    - Programmed to go into infinite sleep mode July 28
  - Caused ~ $2.6 Billion in damages,
- Love Bug worm: $8.75 billion

Statistics: Computer Economics Inc., Carlsbad, California

Internet Worm (First major attack)

- Released November 1988
  - Program spread through Digital, Sun workstations
  - Exploited Unix security vulnerabilities
    - VAX computers and SUN-3 workstations running versions 4.2 and 4.3 Berkeley UNIX code
- Consequences
  - No immediate damage from program itself
  - Replication and threat of damage
    - Load on network, systems used in attack
    - Many systems shut down to prevent further attack

Some historical worms of note

Worm	Date	Distinction
Morris	11/88	Used multiple vulnerabilities, propagate to "nearby" sys
ADM	5/98	Random scanning of IP address space
Ramen	1/01	Exploited three vulnerabilities
Lion	3/01	Stealthy, rootkit worm
Cheese	6/01	Vigilante worm that secured vulnerable systems
Code Red	7/01	First sig Windows worm; Completely memory resident
Walk	8/01	Recompiled source code locally
Nimda	9/01	Windows worm: client-to-server, c-to-c, s-to-s, …
Scalper	6/02	11 days after announcement of vulnerability; peer-to-peer network of compromised systems
Slammer	1/03	Used a single UDP packet for explosive growth

Increasing propagation speed

- Code Red, July 2001
  - Affects Microsoft Index Server 2.0,
    - Windows 2000 Indexing service on Windows NT 4.0.
    - Windows 2000 that run IIS 4.0 and 5.0 Web servers
  - Exploits known buffer overflow in Idq.dll
  - Vulnerable population (360,000 servers) infected in 14 hours
- SQL Slammer, January 2003
  - Affects in Microsoft SQL 2000
  - Exploits known buffer overflow vulnerability
    - Server Resolution service vulnerability reported June 2002
    - Patched released in July 2002 Bulletin MS02-39
  - Vulnerable population infected in less than 10 minutes

Code Red

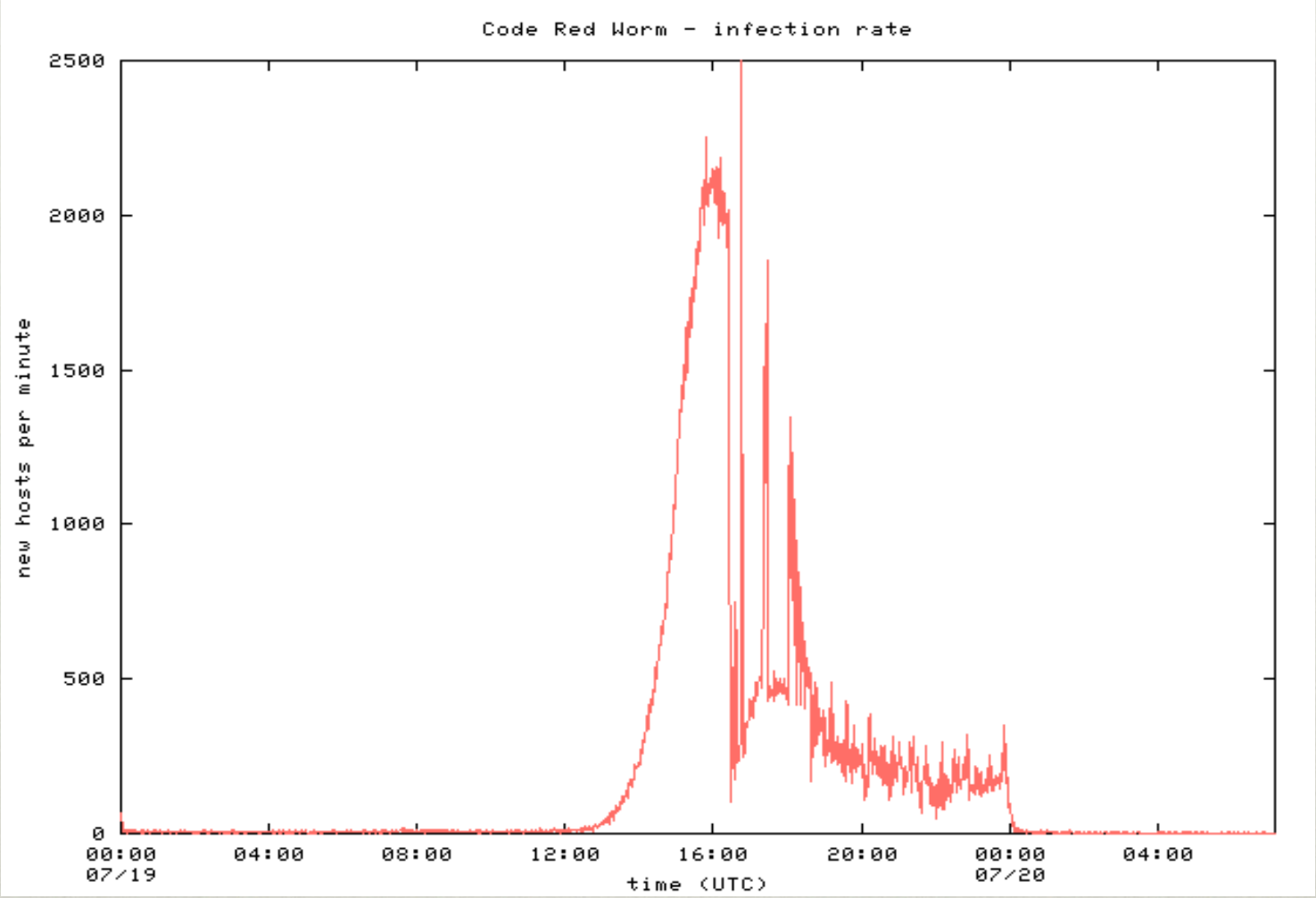
- Initial version released July 13, 2001
  - Sends its code as an HTTP request
  - HTTP request exploits buffer overflow
  - Malicious code is not stored in a file
    - Placed in memory and then run
- When executed,
  - Worm checks for the file C:\Notworm
    - If file exists, the worm thread goes into infinite sleep state
  - Creates new threads
    - If the date is before the 20th of the month, the next 99 threads attempt to exploit more computers by targeting random IP addresses

Code Red of July 13 and July 19

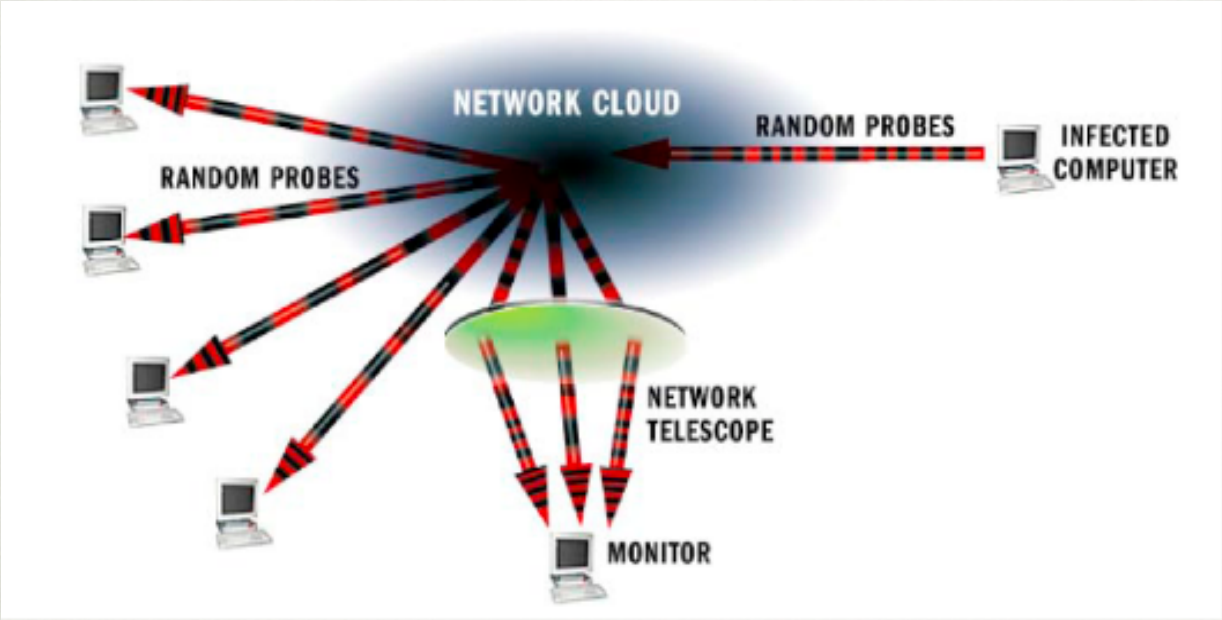
- Initial release of July 13
  - 1st through 20th month: Spread
    - via random scan of 32-bit IP addr space
  - 20th through end of each month: attack.
    - Flooding attack against 198.137.240.91 (*www.whitehouse.gov)*
  - Failure to seed random number generator ⇒ *linear growth*
- Revision released July 19, 2001.
  - White House responds to threat of flooding attack by changing the address of *www.whitehouse.gov*
  - Causes Code Red to die for date ≥ 20th of the month.
  - But: this time random number generator correctly seeded

Slides: Vern Paxson

Infection rate



Measuring activity: network telescope



- Monitor cross-section of Internet address space, measure traffic
  - "Backscatter" from DOS floods
  - Attackers probing blindly
  - Random scanning from worms
- LBNL's cross-section: 1/32,768 of Internet
- UCSD, UWisc's cross-section: 1/256.

Spread of Code Red

- Network telescopes estimate of # infected hosts: 360K. (Beware DHCP & NAT)
- Course of infection fits classic *logistic*.
- Note: larger the vulnerable population, *faster* the worm spreads.

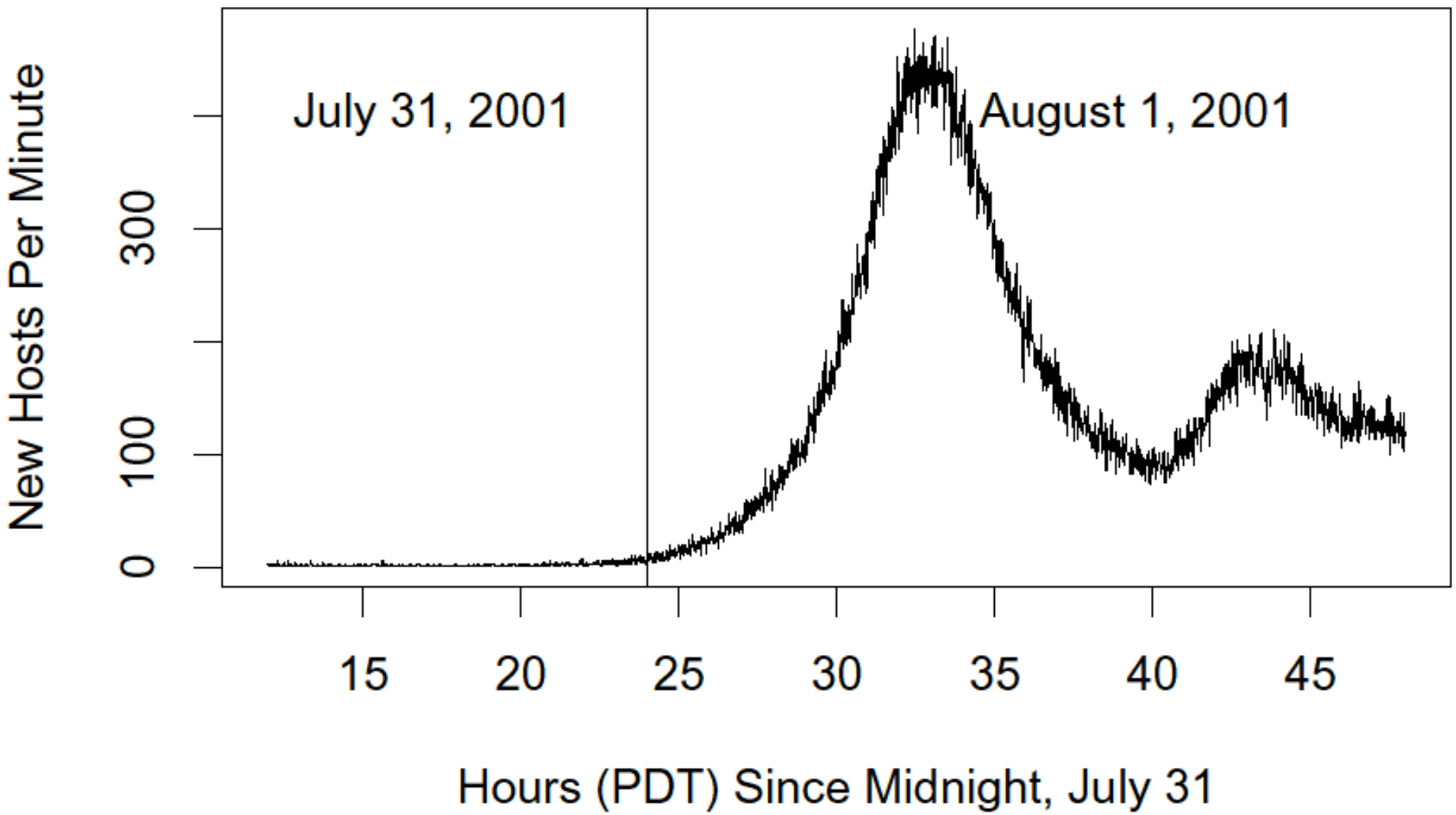
- That night ($\Rightarrow 20^{th}$), worm dies …

  … except for hosts with inaccurate clocks!

- It just takes one of these to restart the worm on August 1^{st} …

Slides: Vern Paxson

Return of Code Red Worm



Slides: Vern Paxson

Code Red 2

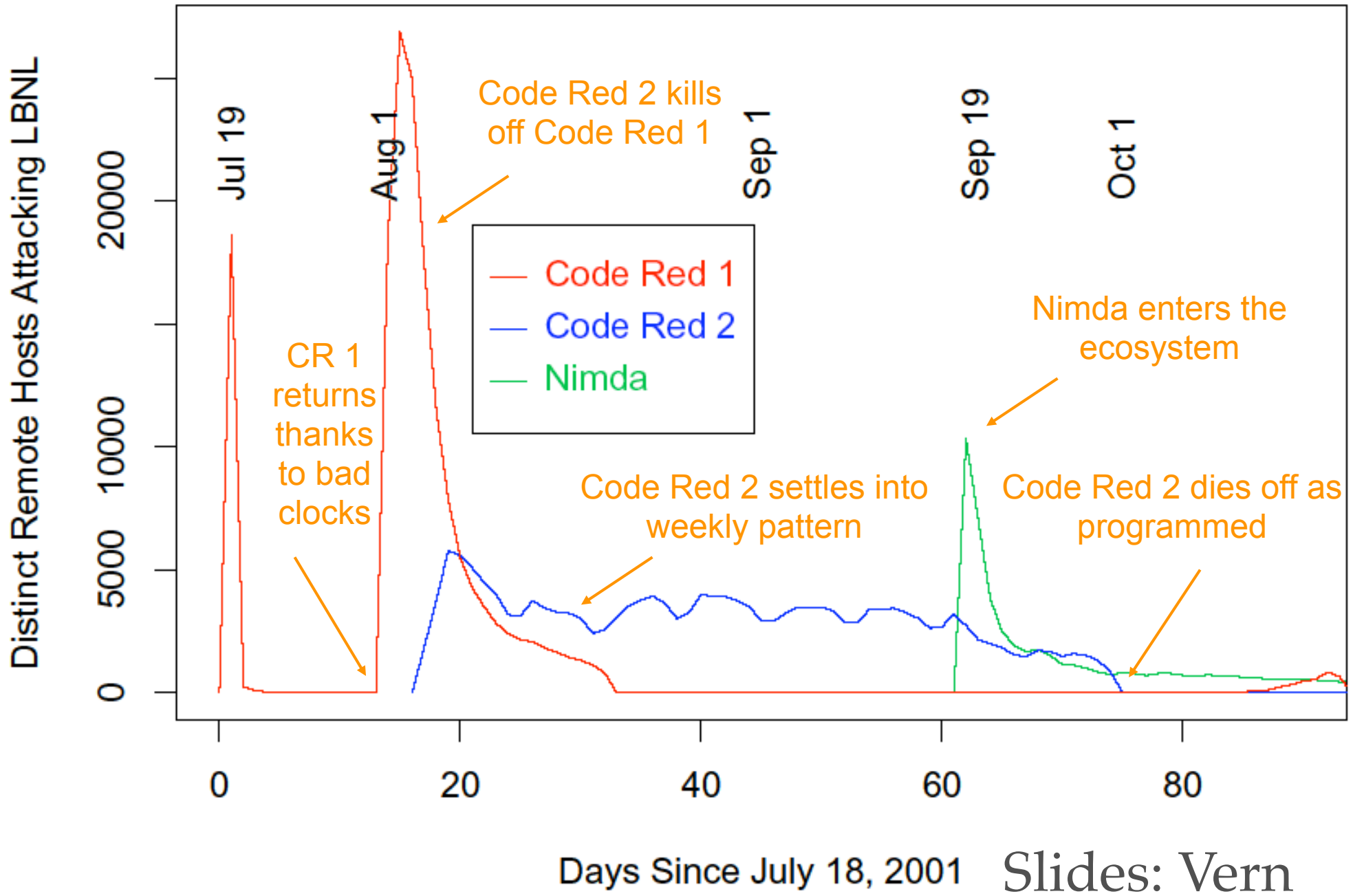
- Released August 4, 2001.
- Comment in code: “Code Red 2.”
  - But in fact completely different code base.
- Payload: a root backdoor, resilient to reboots.
- Bug: crashes NT, only works on Windows 2000.
- *Localized scanning*: prefers nearby addresses.
- Kills Code Red 1.
- Safety valve: programmed to die Oct 1, 2001.

Slides: Vern Paxson

Striving for Greater Virulence: Nimda

- Released September 18, 2001.
- Multi-mode spreading:
  - attack IIS servers via infected clients
  - email itself to address book as a virus
  - copy itself across open network shares
  - modifying Web pages on infected servers w/ client exploit
  - scanning for Code Red II backdoors (!)
- worms form an ecosystem!
- Leaped across firewalls.

Slides: Vern Paxson



Slides: Vern Paxson

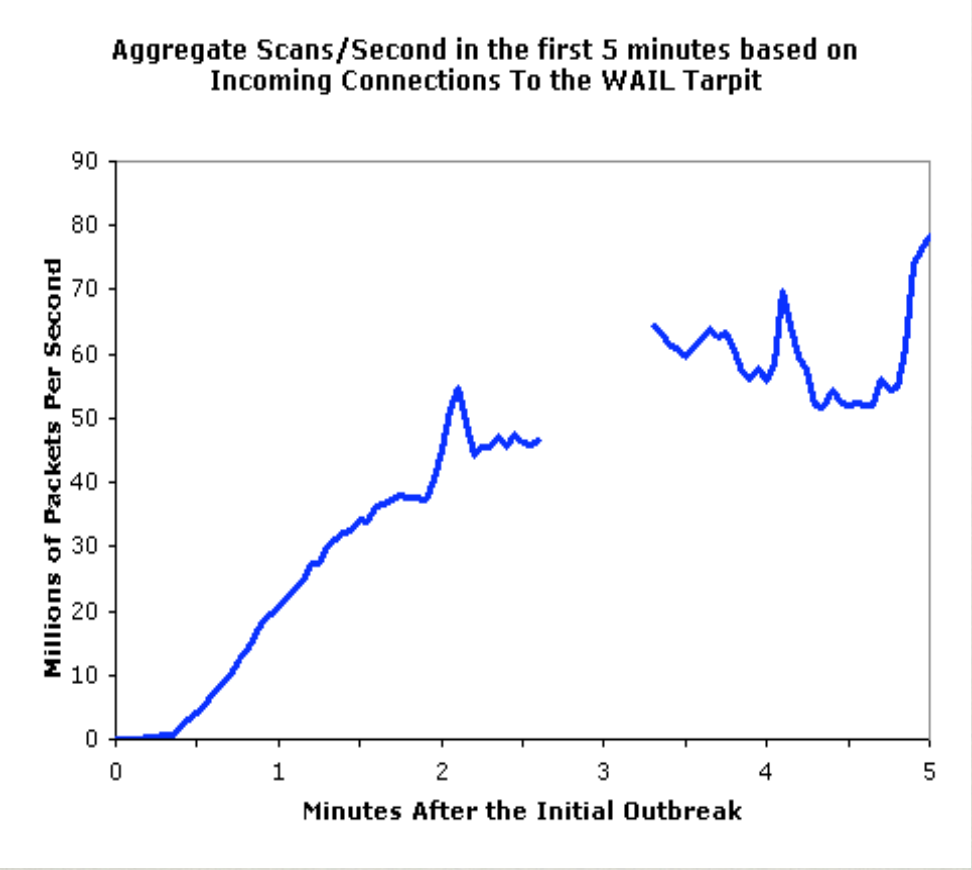
How do worms propagate?

- **Scanning** worms : Worm chooses "random" address
- **Coordinated** scanning : Different worm instances scan different addresses
- **Flash** worms
  - Assemble tree of vulnerable hosts in advance, propagate along tree
    - Not observed in the wild, yet
    - Potential for 106 hosts in < 2 sec ! [Staniford]
- **Meta-server** worm :Ask server for hosts to infect (e.g., Google for "powered by phpbb")
- **Topological** worm: Use information from infected hosts (web server logs, email address books, config files, SSH "known hosts")
- **Contagion** worm : Propagate parasitically along with normally initiated communication

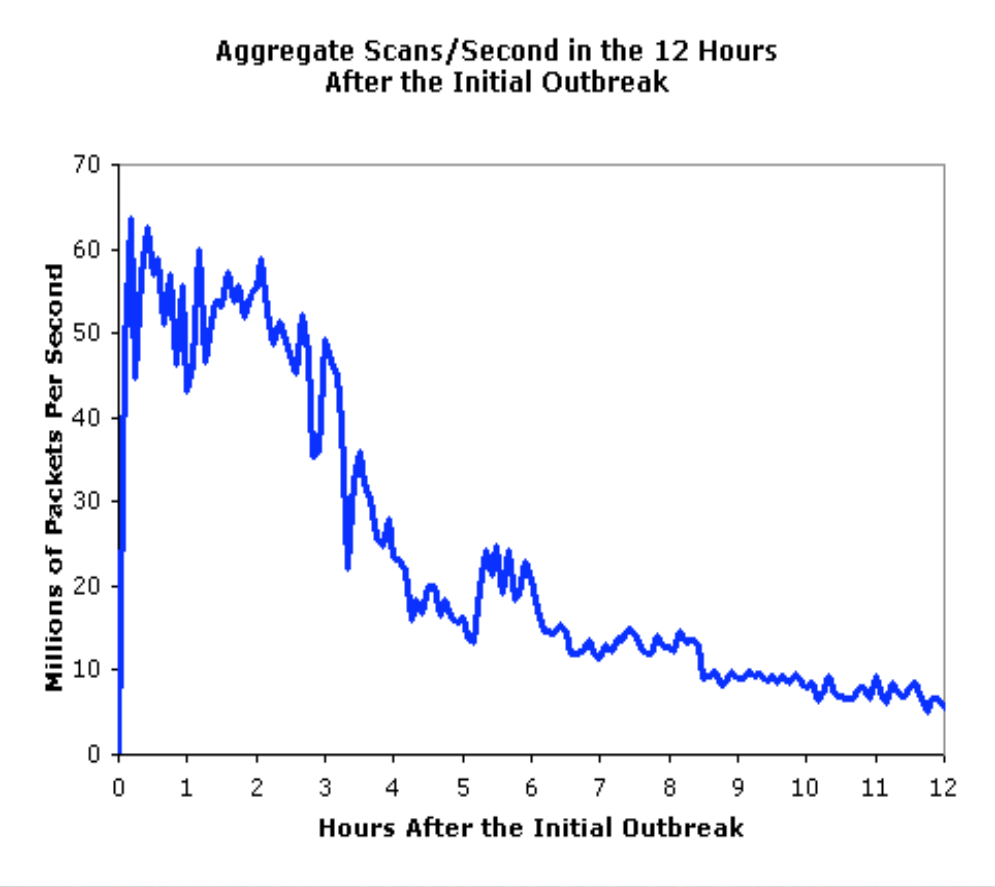
SLAMMER

- 01/25/2003
- Vulnerability disclosed : 25 june 2002
- Better scanning algorithm
- UDP Single packet : 380bytes

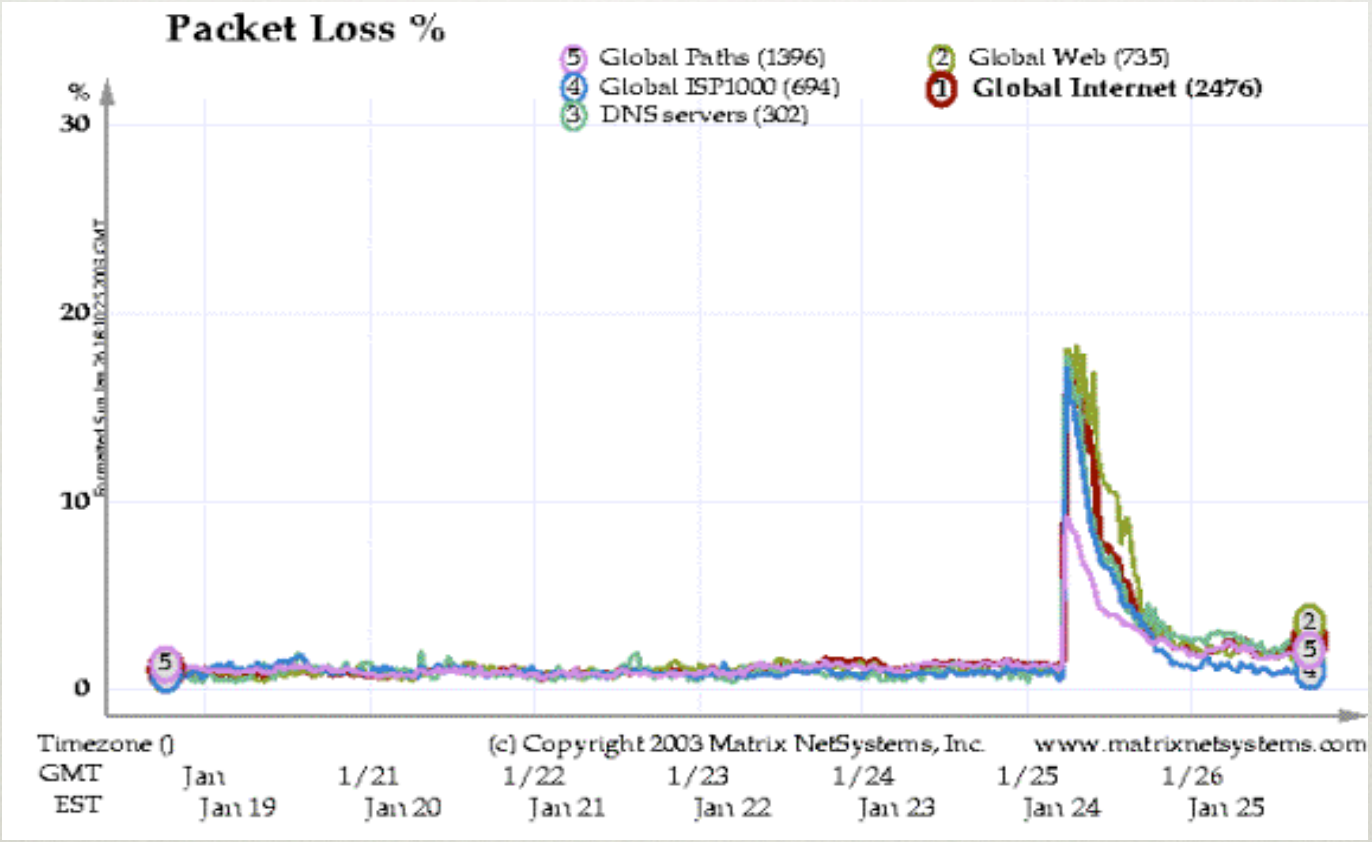
Slammer propagation



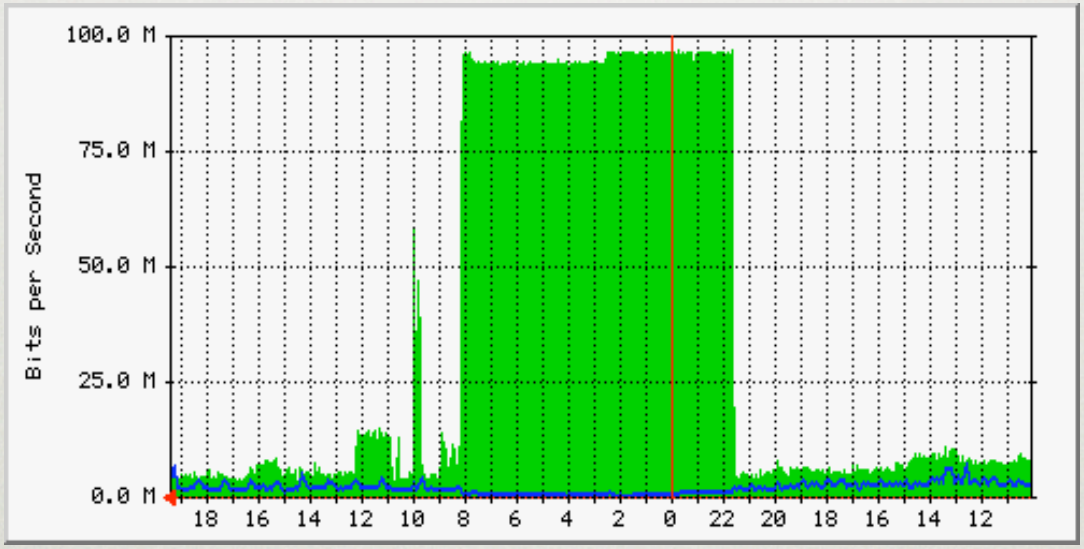
Number of scan/sec



Packet loss


Packet Loss %
5 Global Paths (1396)
4 Global ISP1000 (694)
3 DNS servers (302)
2 Global Web (735)
1 Global Internet (2476)
%
30
20
10
0
Timezone ()
GMT
EST
(c) Copyright 2003 Matrix NetSystems, Inc.
www.matrixnetsystems.com
Jan
1/21
1/22
1/23
1/24
1/25
1/26
Jan 19
Jan 20
Jan 21
Jan 22
Jan 23
Jan 24
Jan 25

A server view



Consequences

- ATM systems not available
- Phone network overloaded (no 911!)
- 5 DNS root down
- Planes delayed

Worm Detection and Defense

- Detect via *honeyfarms*: collections of "honeypots" fed by a network telescope.
  - Any outbound connection from honeyfarm = worm.

    (at least, that's the theory)
  - Distill *signature* from inbound/outbound traffic.
  - If telescope covers N addresses, expect detection when worm has infected 1/N of population.
- Thwart via *scan suppressors*: network elements that block traffic from hosts that make failed connection attempts to too many other hosts
  - 5 minutes to several weeks to write a signature
  - Several hours or more for testing

Need for automation

- Current threats can spread faster than defenses can reaction
- Manual capture/analyze/signature/rollout model too slow

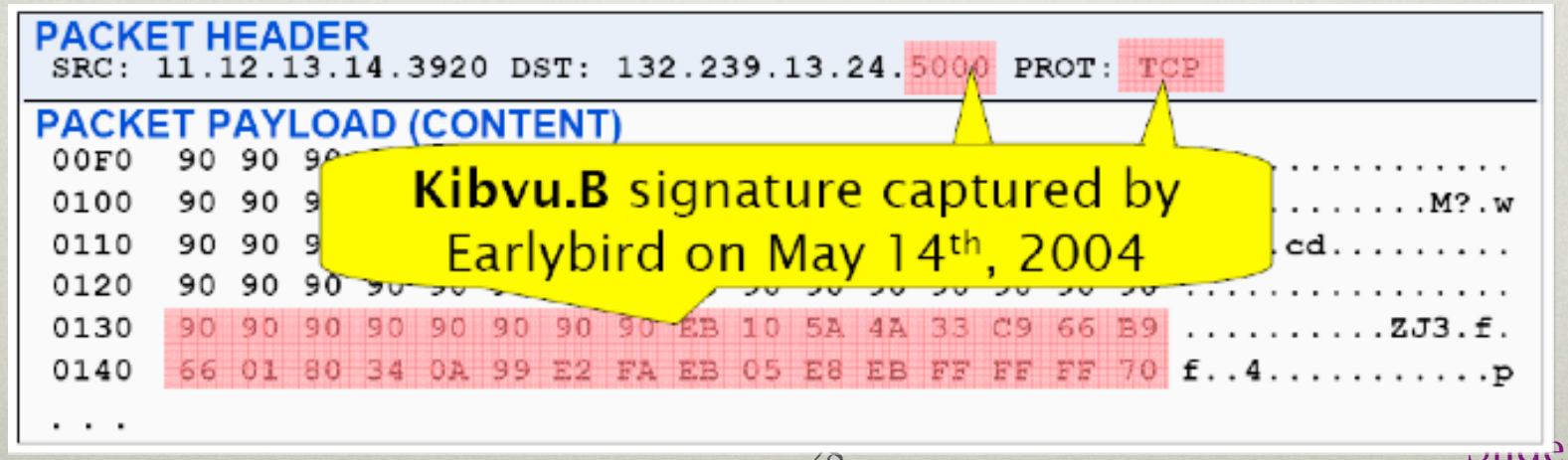
Slide: Carey Nachenberg, Symantec

Signature inference

- Challenge
  - need to automatically learn a content "signature" for each new worm – potentially in less than a second!
- Some proposed solutions
  - Singh et al, Automated Worm Fingerprinting, OSDI '04
  - Kim et al, Autograph: Toward Automated, Distributed Worm Signature Detection, USENIX Sec '04

Signature inference

- Monitor network and look for strings common to traffic with worm-like behavior
  - Signatures can then be used for content filtering



Slide: S Savage

Content sifting

- Assume there exists some (relatively) unique invariant bitstring W across all instances of a particular worm (true today, not tomorrow...)
- Two consequences
  - **Content Prevalence**: W will be more common in traffic than other bitstrings of the same length
  - **Address Dispersion**: the set of packets containing W will address a disproportionate number of distinct sources and destinations
- Content sifting: find W's with high content prevalence and high address dispersion and drop that traffic

Slide: S Savage

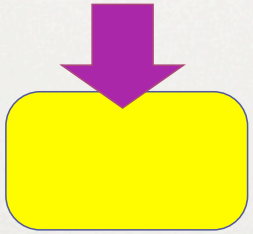
Observation:
High-prevalence strings are rare

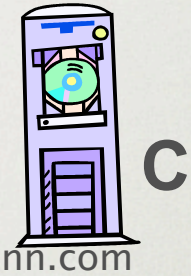
Only 0.6% of the 40 byte **substrings repeat more than 3 times** in a minute

(Stefan Savage, UCSD *)

The basic algorithm

Detector in network



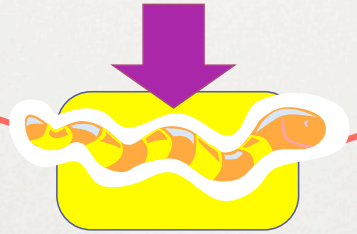


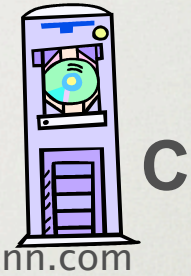
Prevalence Table

Address **Dispersion** Table

Sources Destinations

Detector in network





Prevalence Table

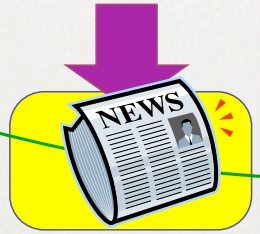
	1

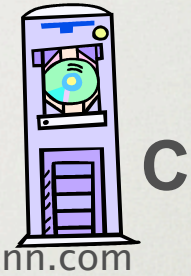
Address **Dispersion** Table

Sources	Destinations
1 (A)	**1 (B)**

(Stefan Savage, UCSD *)

Detector in network





Prevalence Table

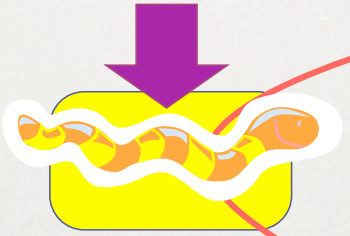
	**1**
	**1**

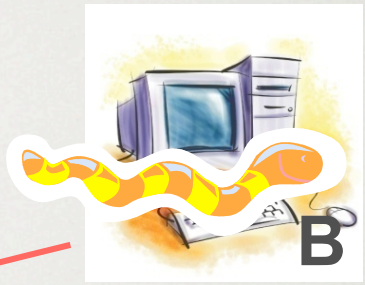
Address **Dispersion** Table

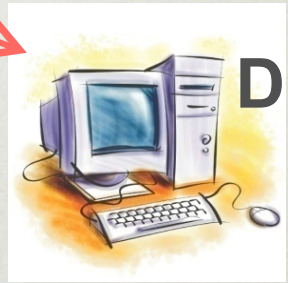
Sources	Destinations
1 (A)	**1 (B)**
1 (C)	**1 (A)**

(Stefan Savage, UCSD *)

Detector in network







Prevalence Table

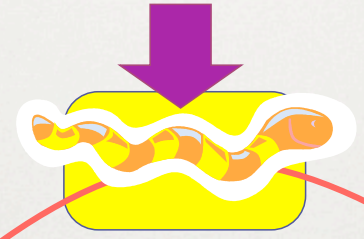
	**2**
	**1**

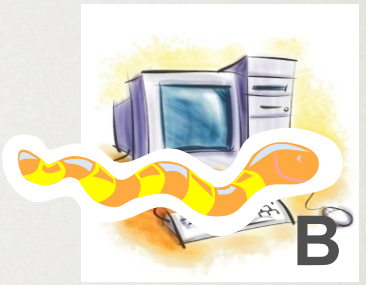
Address **Dispersion** Table

Sources	Destinations
2 (A,B)	**2 (B,D)**
1 (C)	**1 (A)**

(Stefan Savage, UCSD *)

Detector in network





Prevalence Table

3	
1	

Address **Dispersion** Table

Sources	Destinations
3	3
(A,B,D)	(B,D,E)
1 (C)	1 (A)

Project 2

Project Status

- 30% of submission came in before 4pm
- Some submission are late

Background

- Network security is about packets manipulation
  - DDOS
  - Firewall / NAT
  - Man in the middle
  - Network Scouting

Project goal

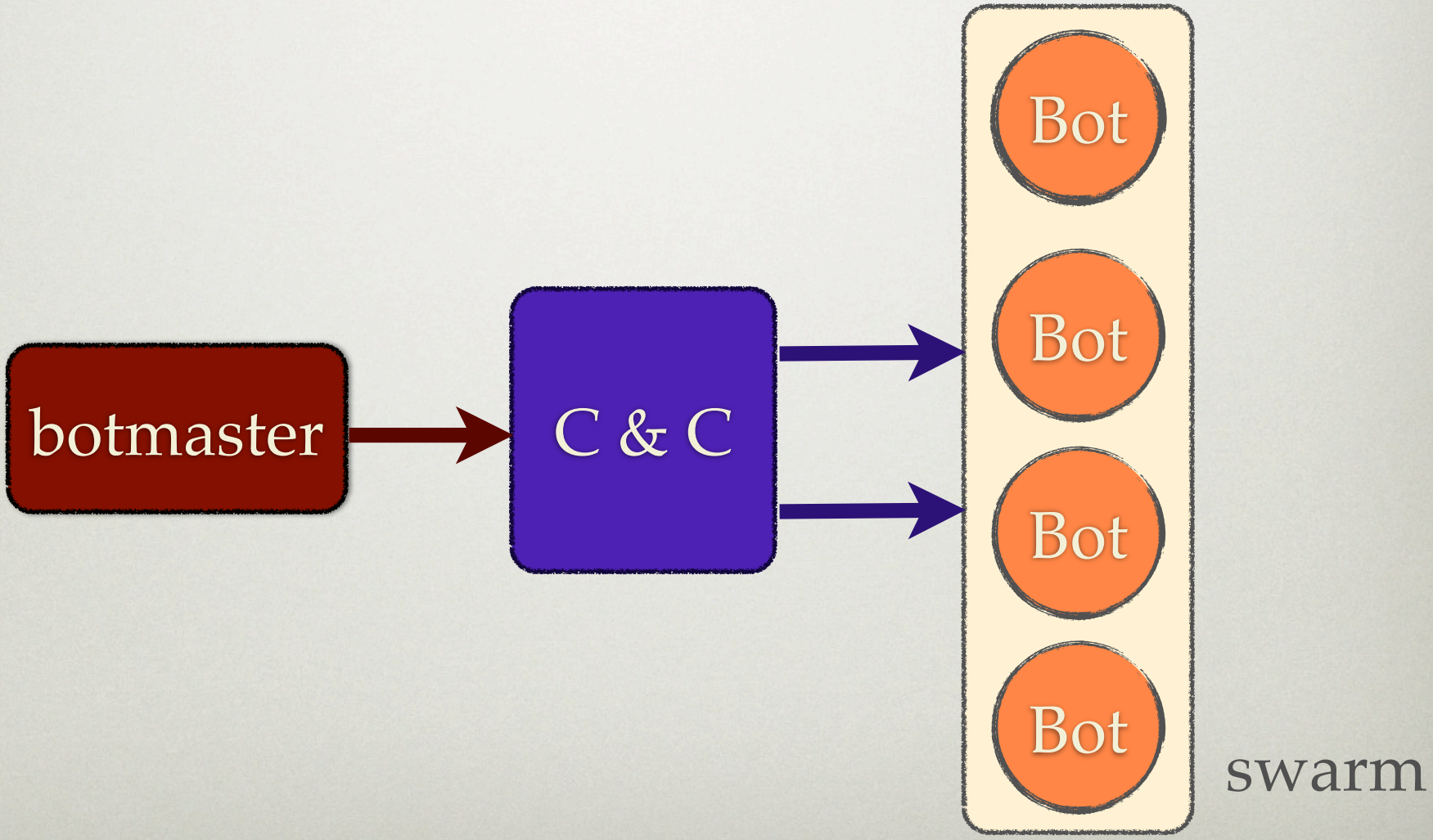
- Crafting packet
- Understand sniffing
- Understand Firewall and routing
- Understand Network debugging

Botnet

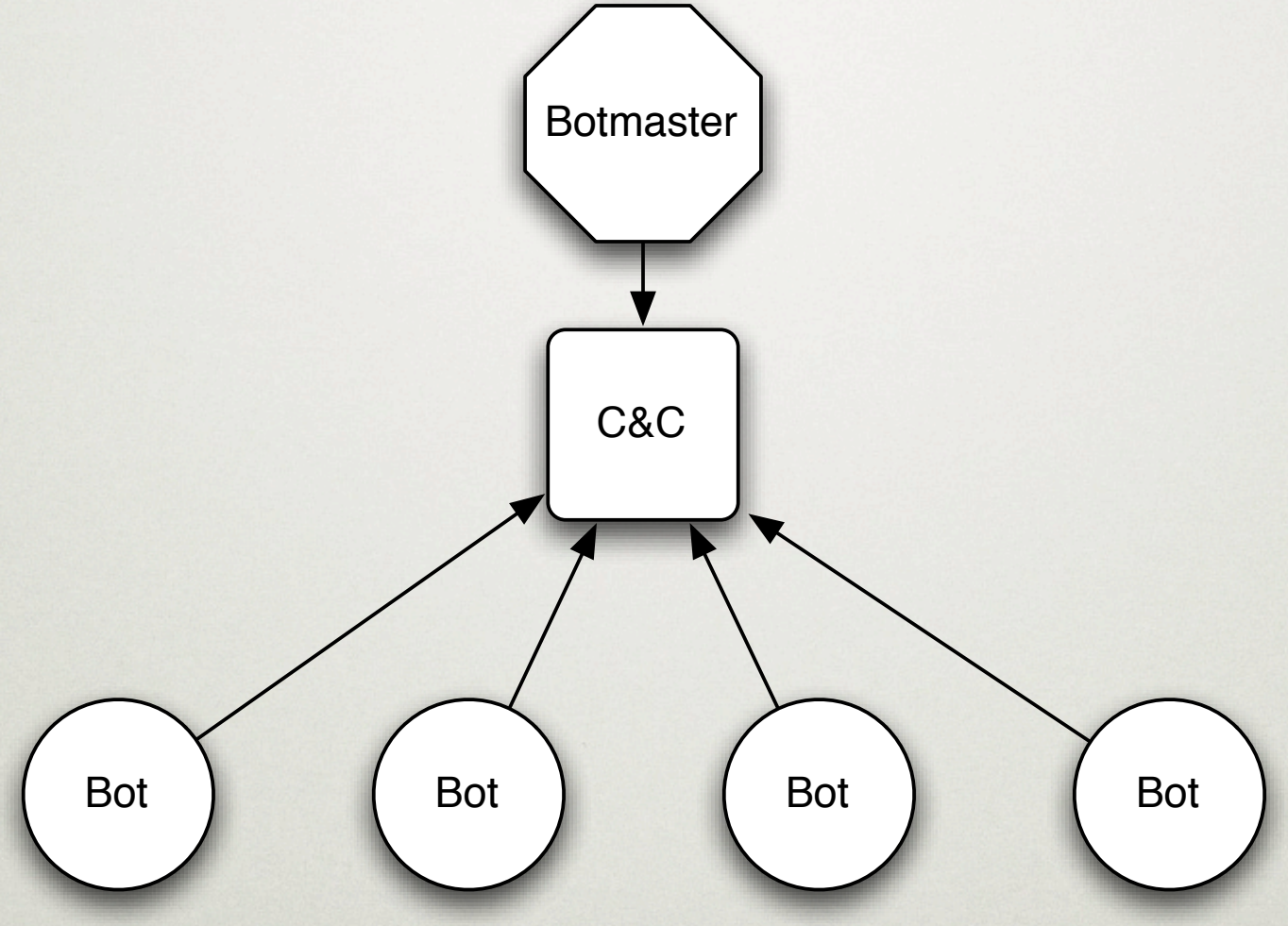
Outline

- Worm Generation 1
- Botnet
- Fast Flux
- Worm Generation 2
- Underground Economy

What is a botnet ?

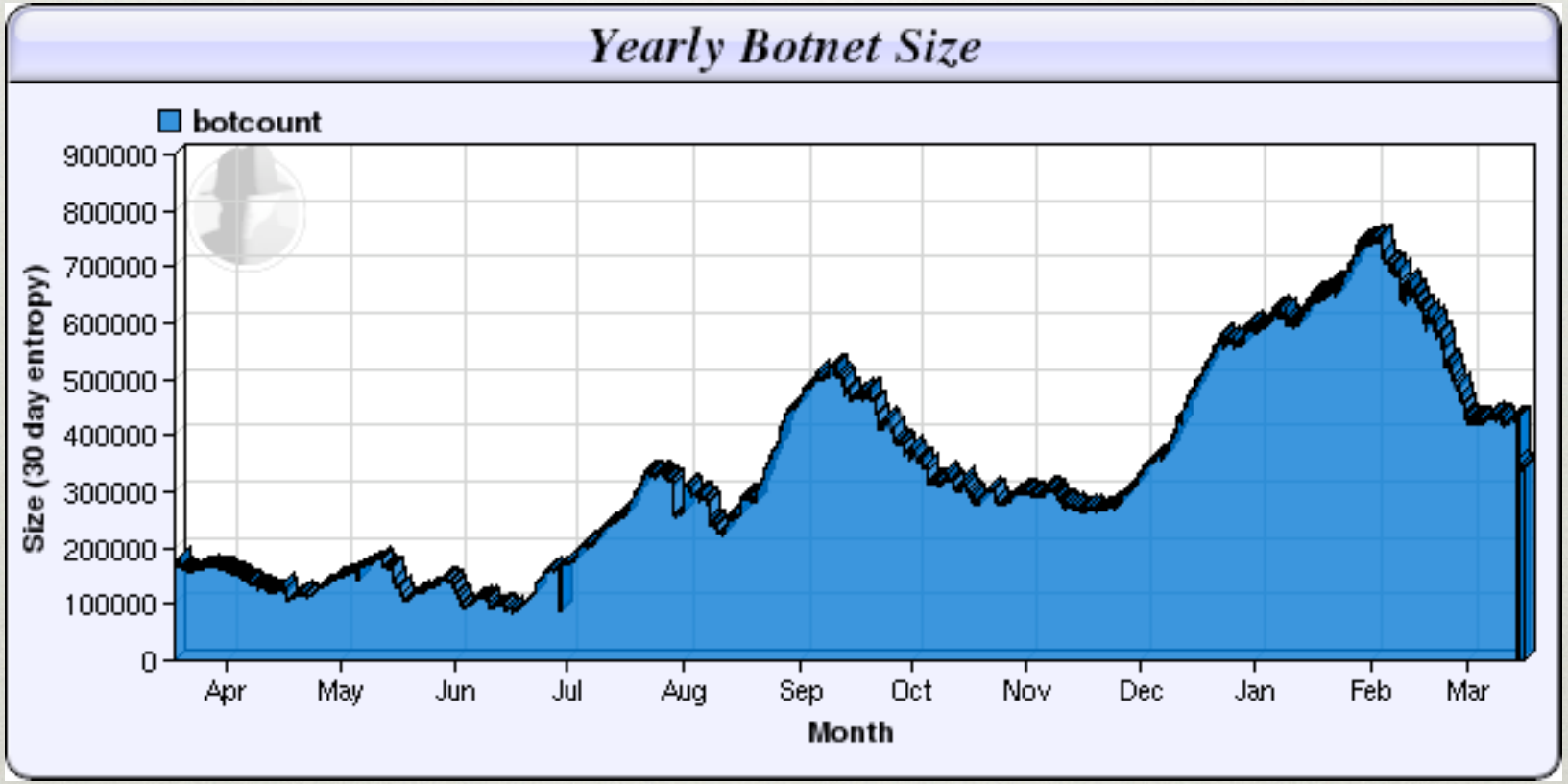


Centralized botnet

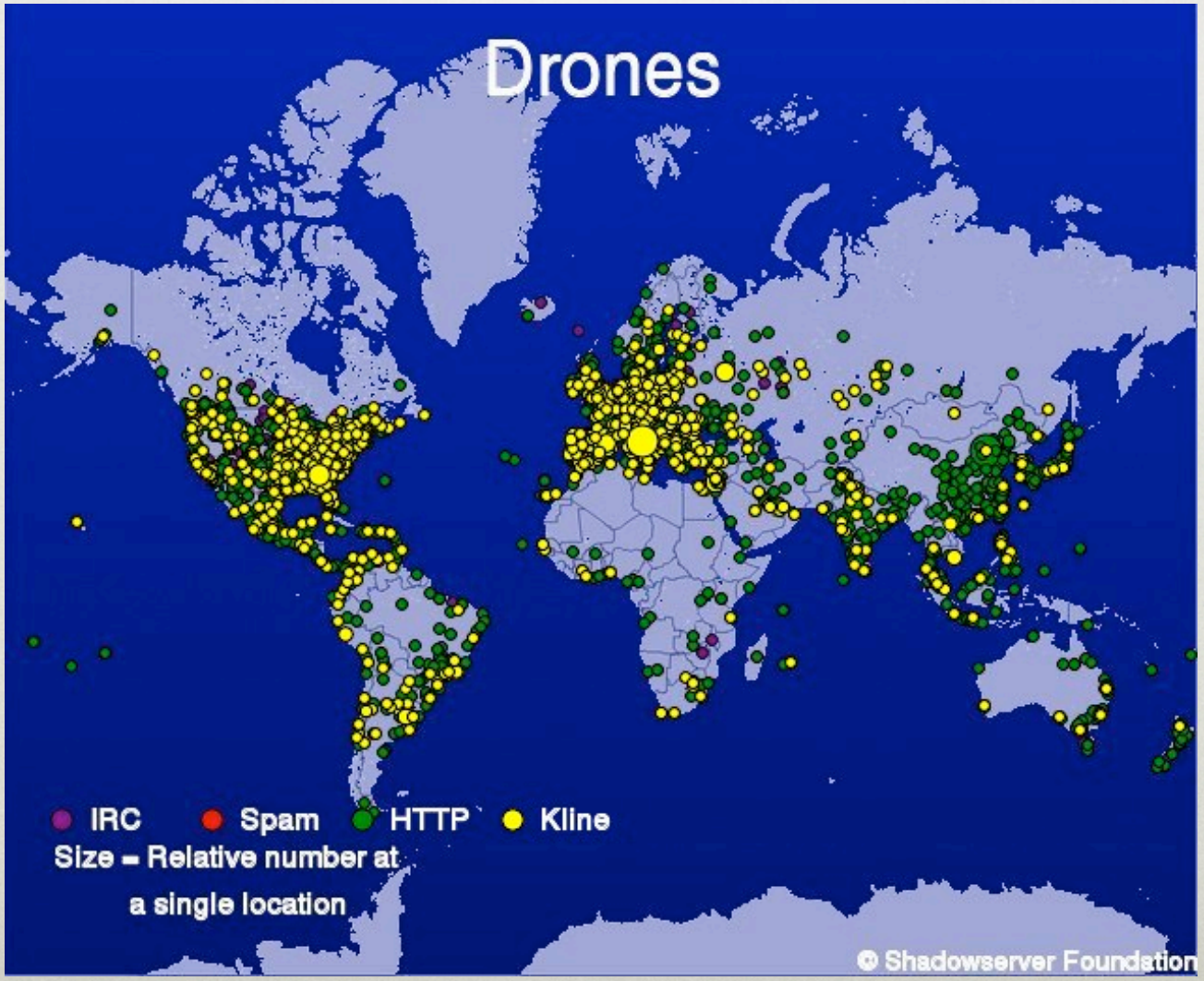


Centralized

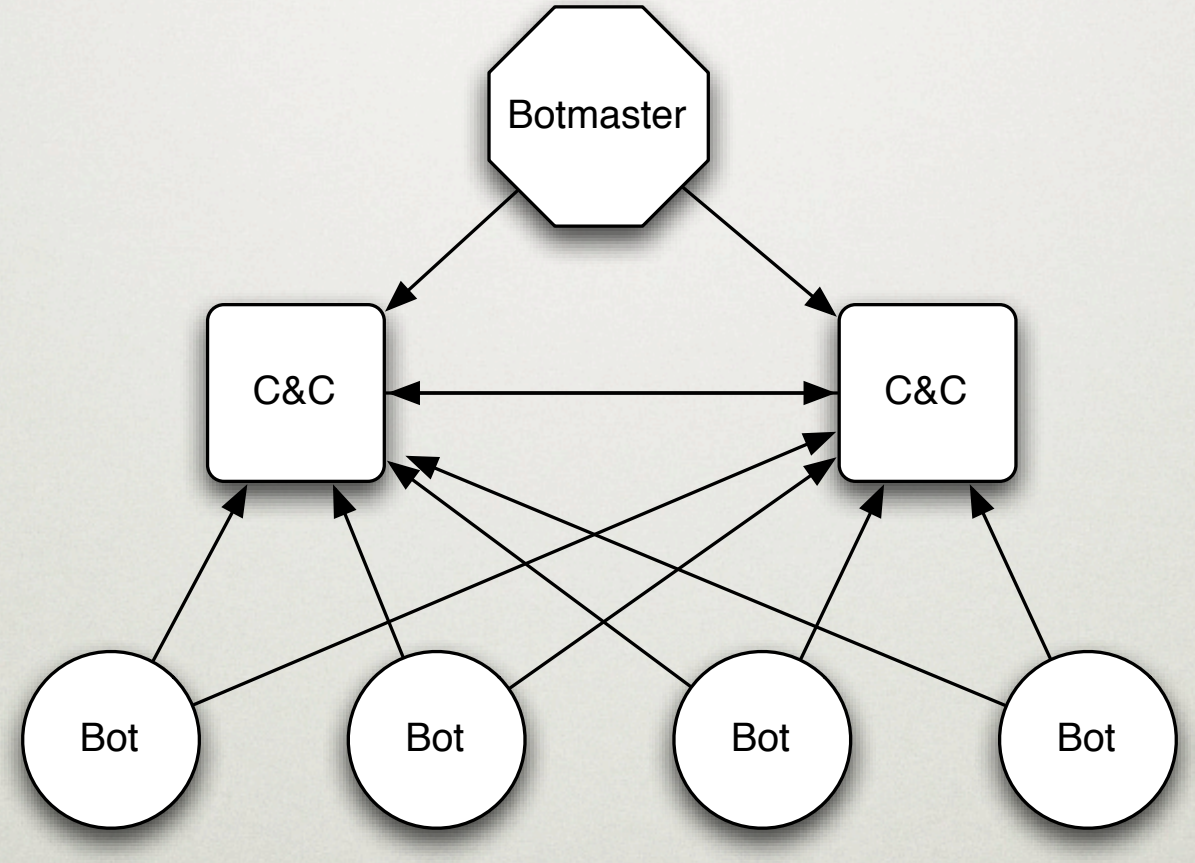
C&C centralized Stat



World wild problem



Type of botnet



Distributed

Example Storm

- Also known as W32/Peacomm Trojan
- Use P2P communication : kademlia
- Command are stored into the DHT table

History

- Started in January 2007
- First email title : 230 dead as storm batters Europe

Key feature

- Smart social engineering
- Use client side vulnerabilities
- Hijack chat session to lure user
- Obfuscated C&C
- Actively updated
- Use Spam templates

Smart SPAM

- Venezuelan leader: "Let's the War beginning".
- U.S. Southwest braces for another winter blast. More then 1000 people are dead.
- The commander of a U.S. nuclear submarine lunch the rocket by mistake.
- The Supreme Court has been attacked by terrorists. Sen. Mark Dayton dead!
- Third World War just have started!
- U.S. Secretary of State Condoleezza Rice has kicked German Chancellor Angela Merkel

More recently

- Valentine day
- Obama victory
- 1 april

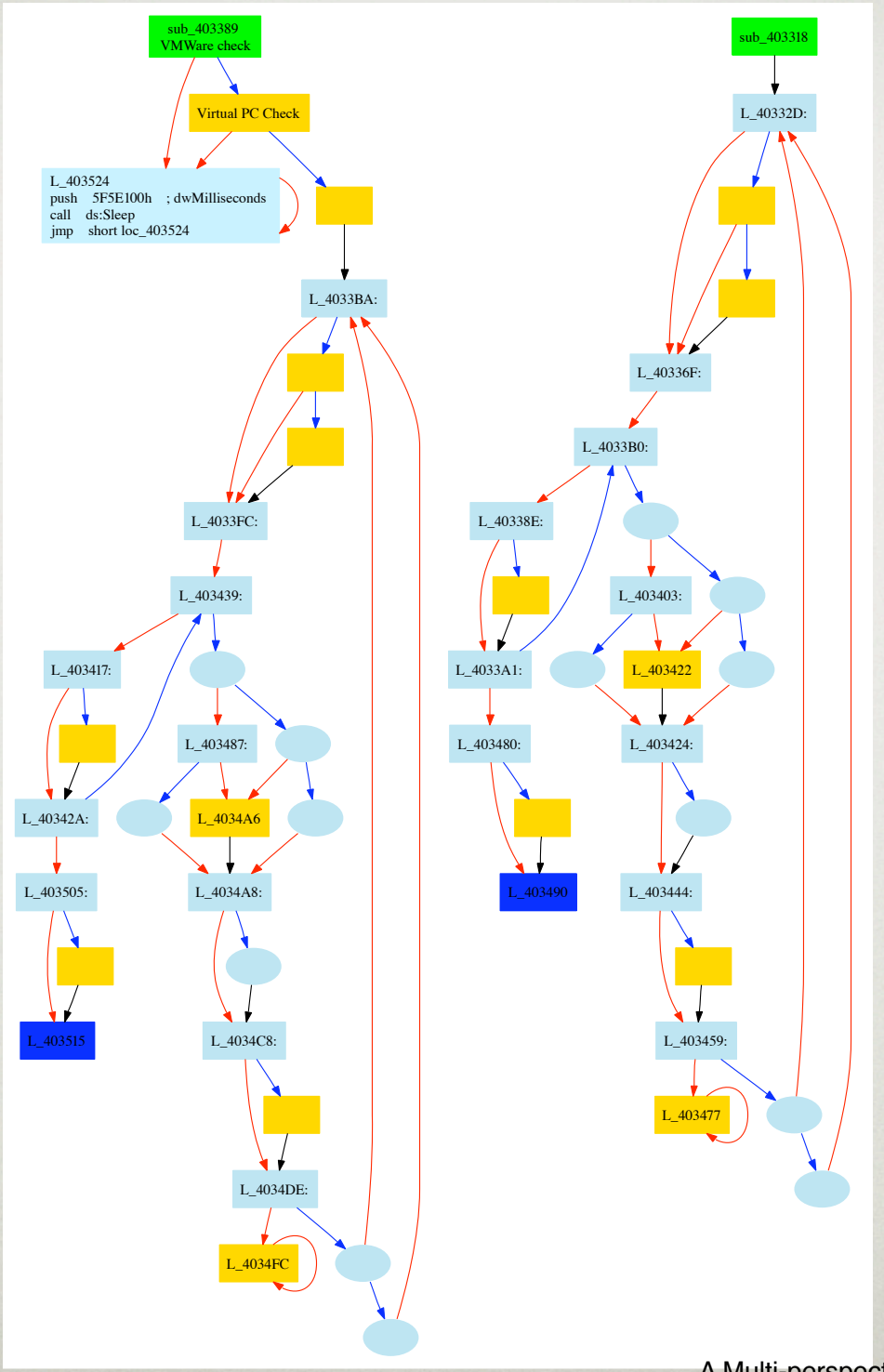
Composition

- game0.exe - Backdoor/downloader
- game1.exe - SMTP relay
- game2.exe - E-mail address stealer
- game3.exe - E-mail virus spreader
- game4.exe - Distributed denial of service (DDos) attack tool
- game5.exe - Updated copy of Storm

- 128 bit md4=<ip><port><2 bytes flag>

RDV POINT

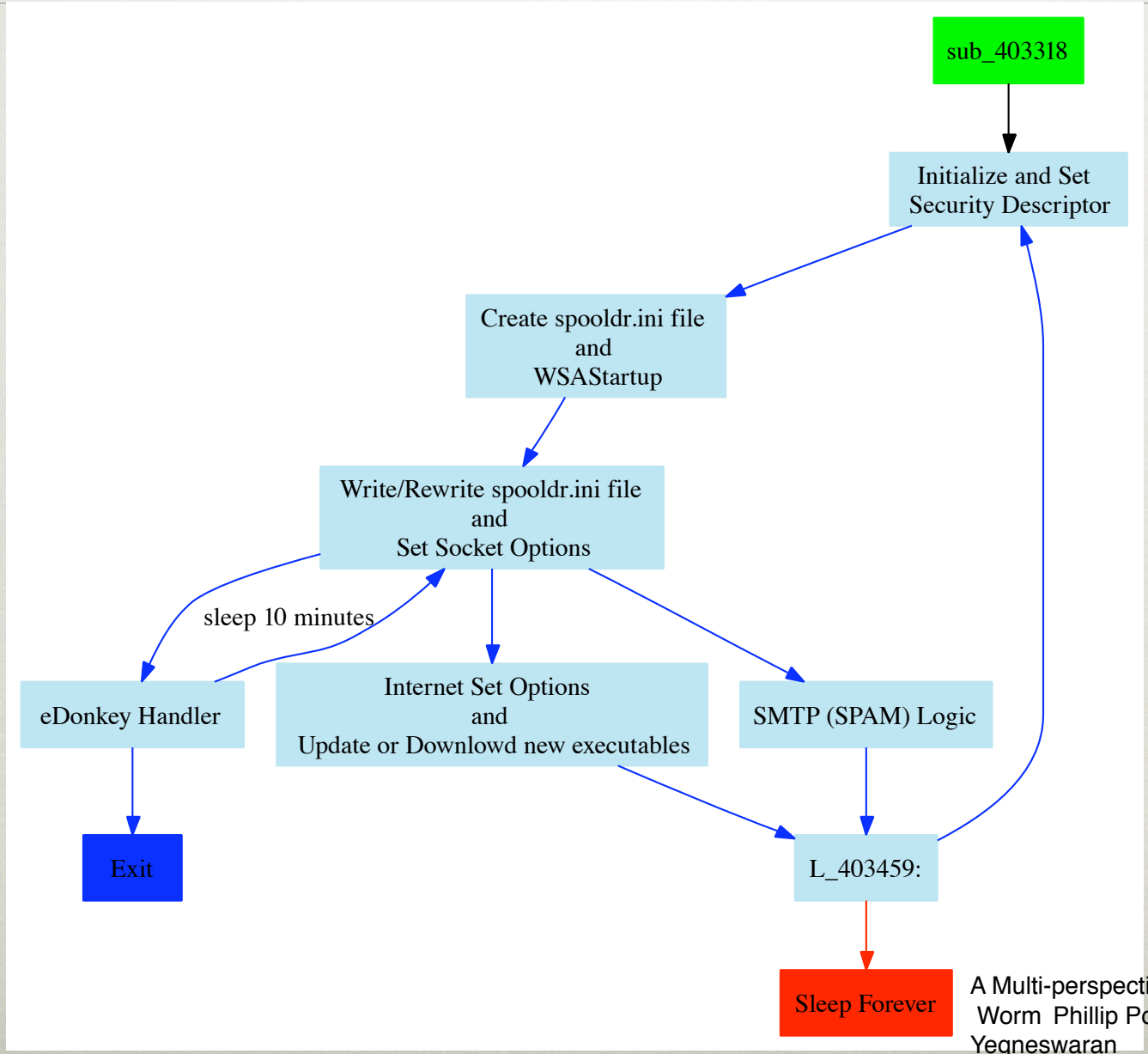
- Compute a secret Key value
  - Use a random generator
  - A secret seed
  - The time



A Multi-perspective Analysis of the Storm (Peacomm) Worm

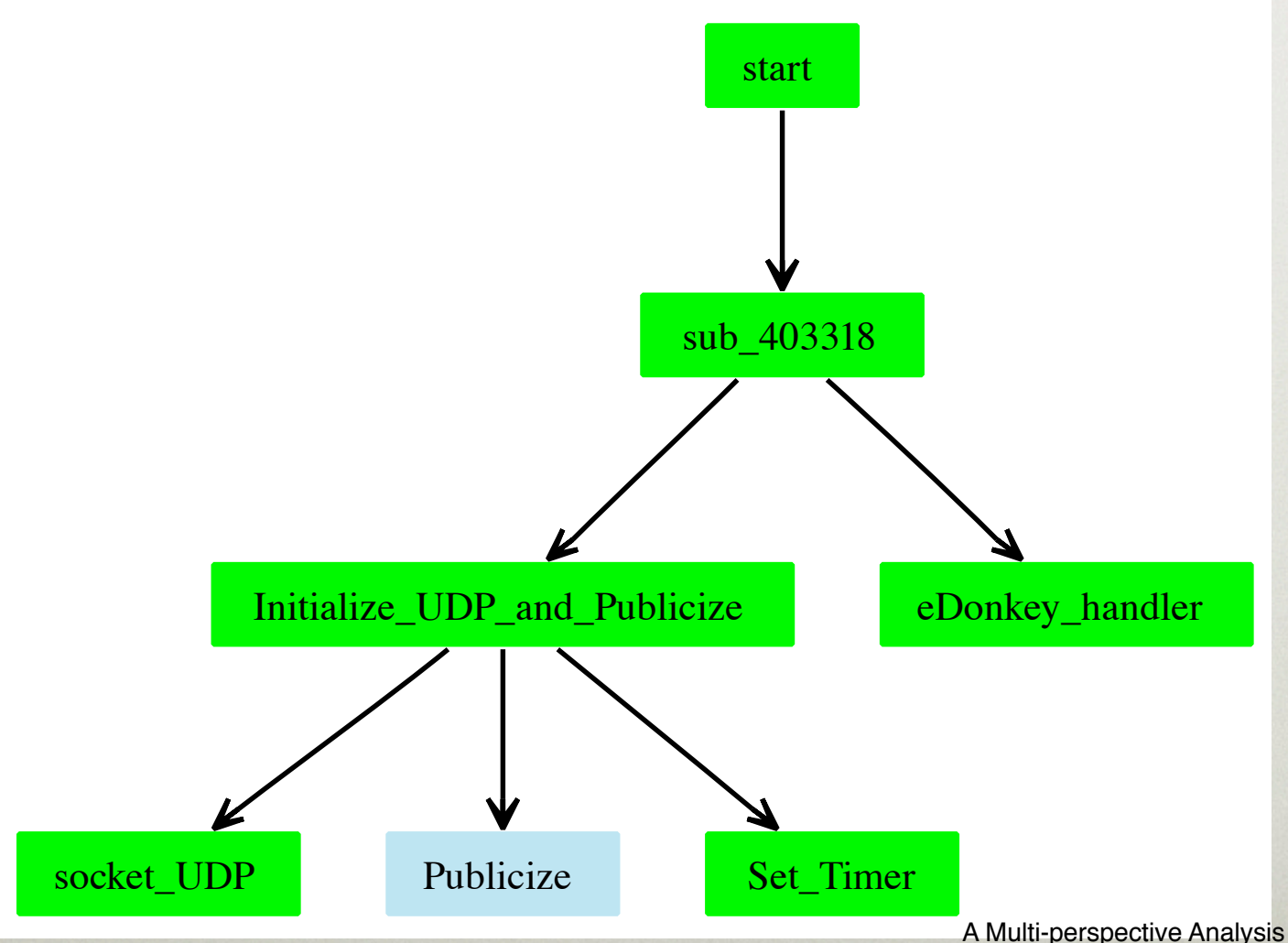
Phillip Porras and Hassen Sa¨ıdi and Vinod Yegneswaran

Overview of the logig

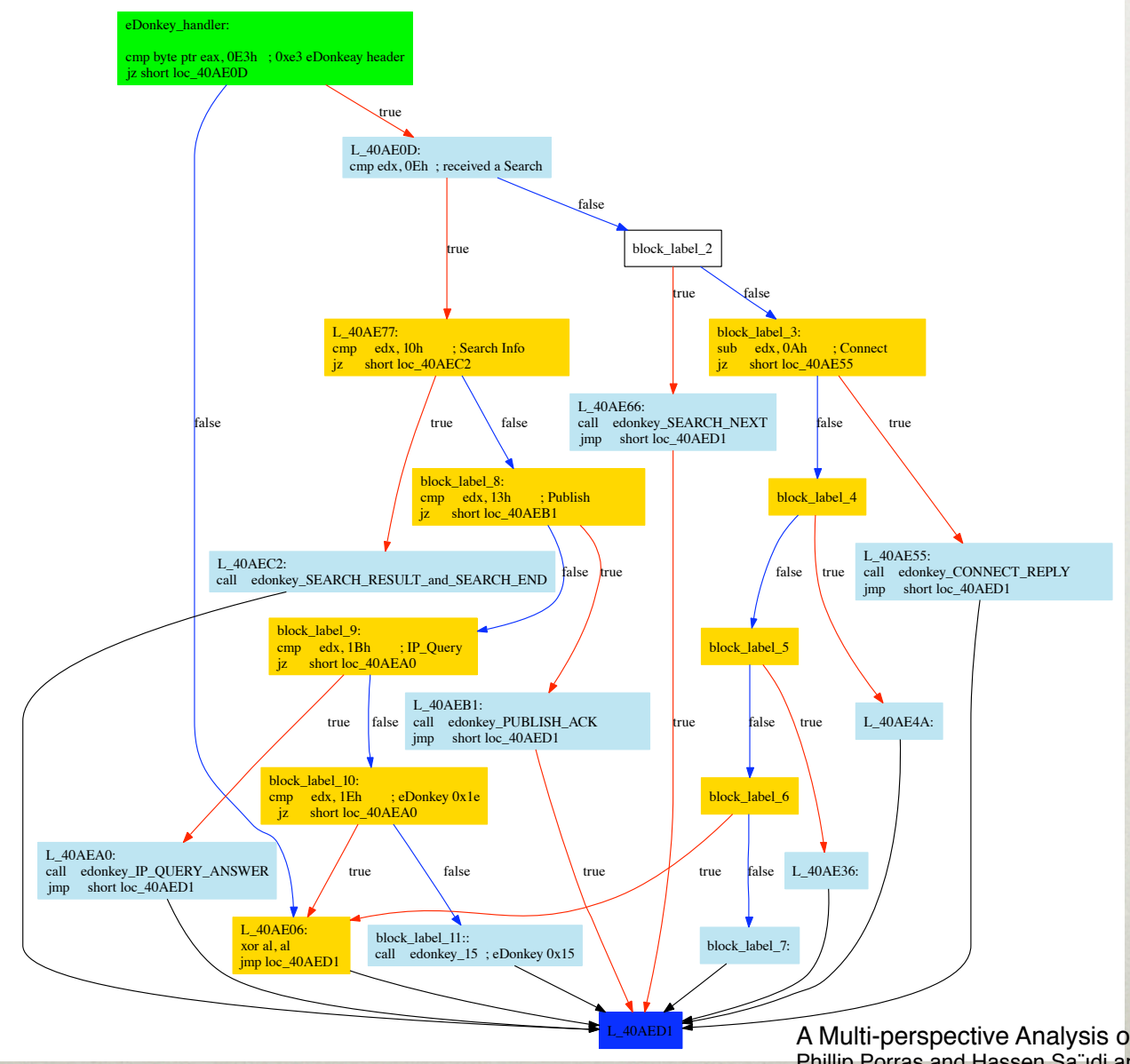


A Multi-perspective Analysis of the Storm (Peacomm
Worm Phillip Porras and Hassen Saïdi and Vinod
Yegneswaran

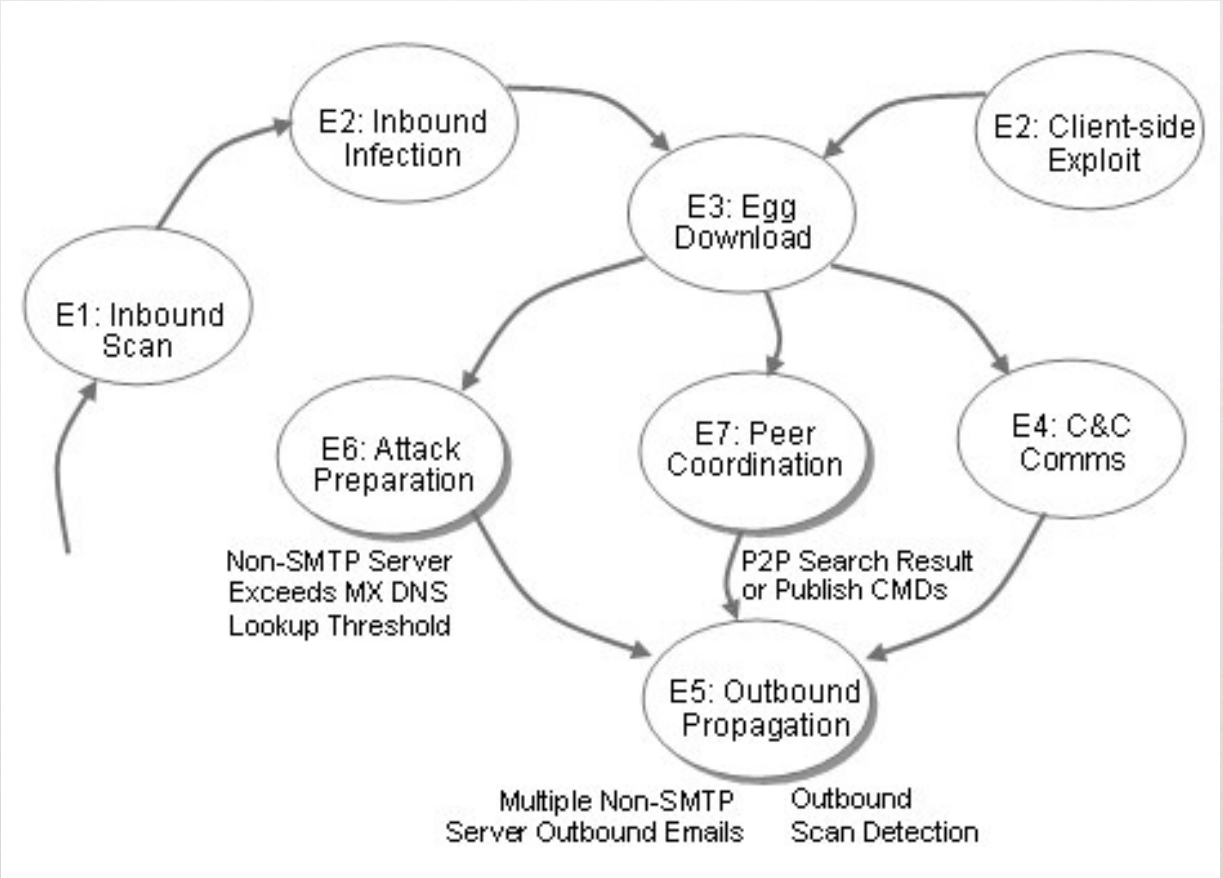
OVERNET PROTOCOL



Overnet protocol handler



Detecting Storm



A Multi-perspective Analysis of the Storm (Peacomm) Worm
Phillip Porras and Hassen Sa¨ıdi and Vinod Yegneswaran

How storm work

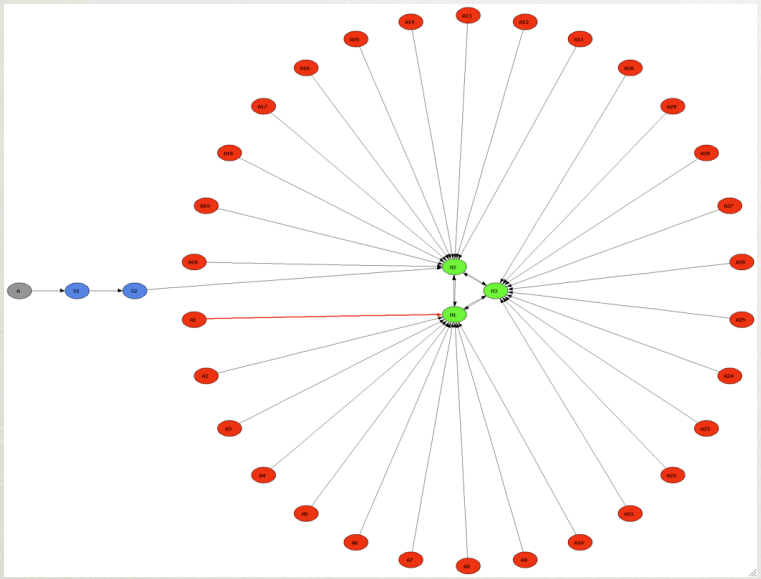
- *Connect to Overnet*
- *Download Secondary Injection URL (hard coded key)*
- *Decrypt Secondary Injection URL*
- *Download Secondary Injection*
- *Execute Secondary Injection*

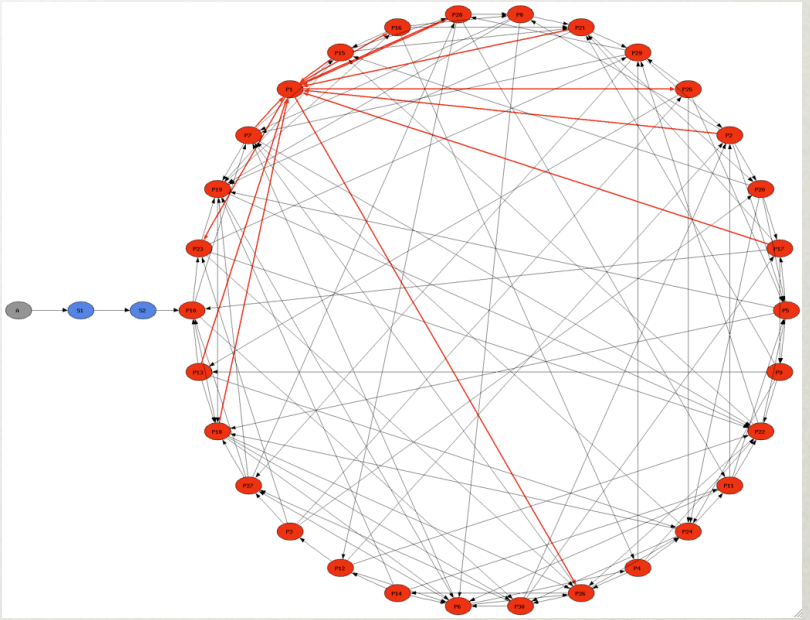
Peer-to-Peer Botnets: Overview and Case Study
Julian B. Grizzard Vikram Sharma, Chris Nunnery, David Dagon

Weakness

- Initial peer list
- sybil attack
- Index poisoning

Network view





Command and control structures in malware: From Handler/Agent to P2P, by Dave Dittrich and Sven Dietrich, USENIX ;login: vol. 32, no. 6, December 2007, pp. 8-17

Comparison

		Communication system		Security	
	Design complexity	Channel type	Message latency	Detectability	Resilience
Centralized	**Low**	**Bidirectional**	**Low**	High	Low
Distributed	High	Unidirectional	High	**Low**	**High**

Fast Flux

Outline

- Worm Generation 1
- Botnet
- Fast Flux
- Worm Generation 2
- Underground Economy

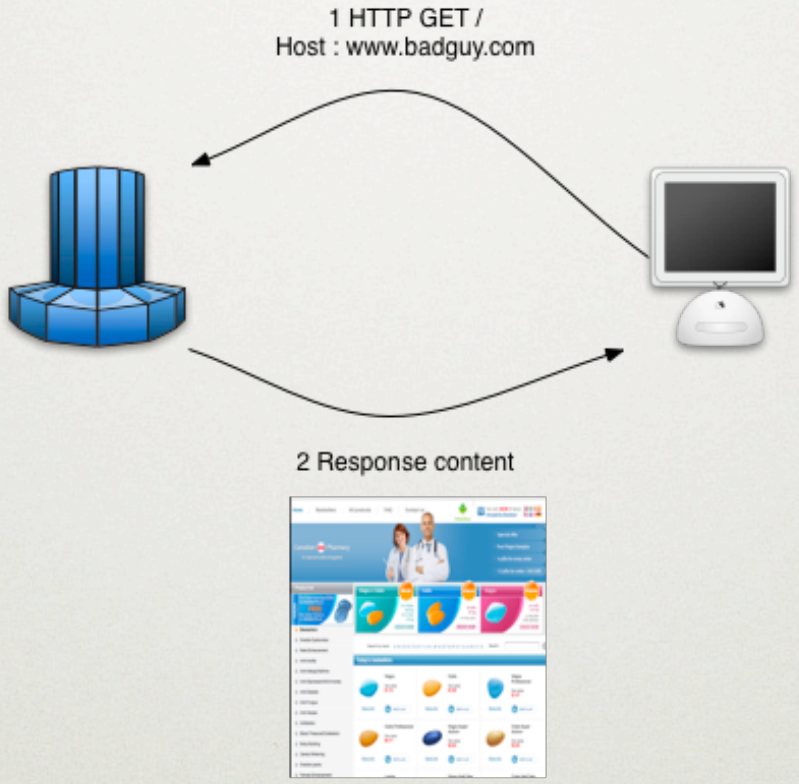
Goal

- Resilient service hosting
- Prevent tracing

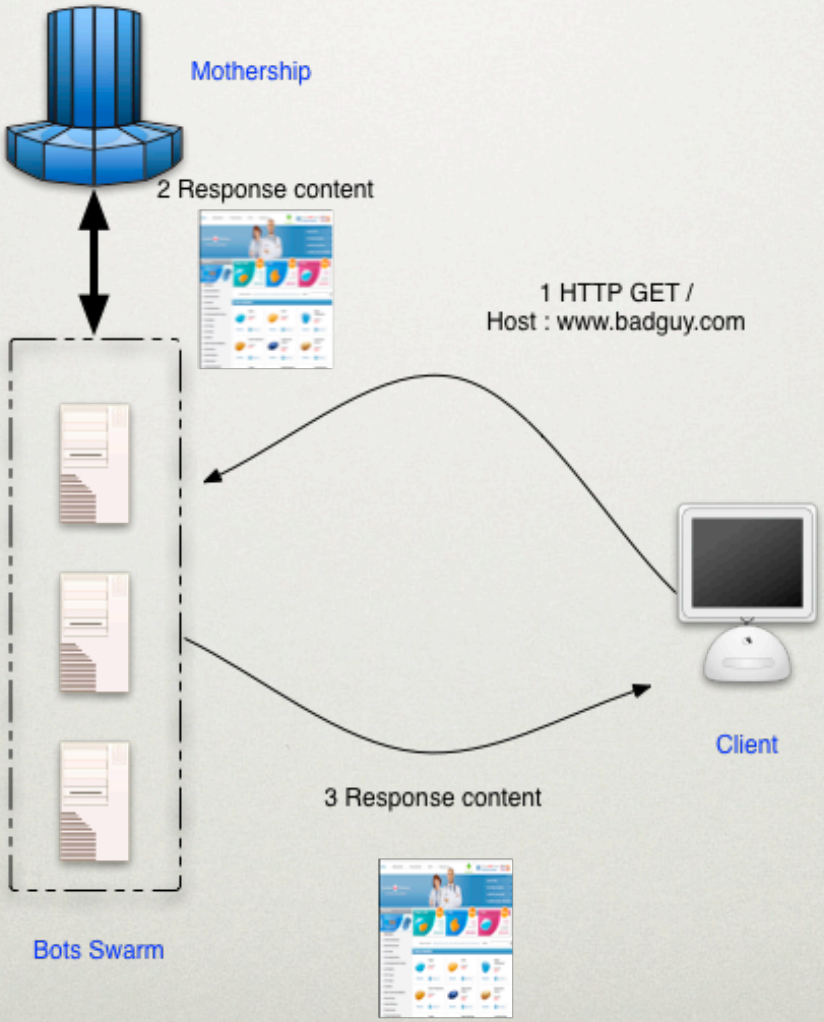
Receipt

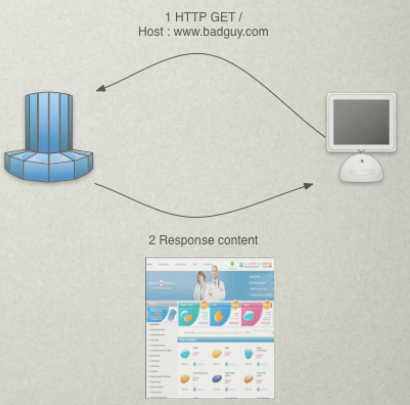
- One domain
- Round robin DNS capability
- Thousand of IP (bots)
- Short TTL

Normal Hosting



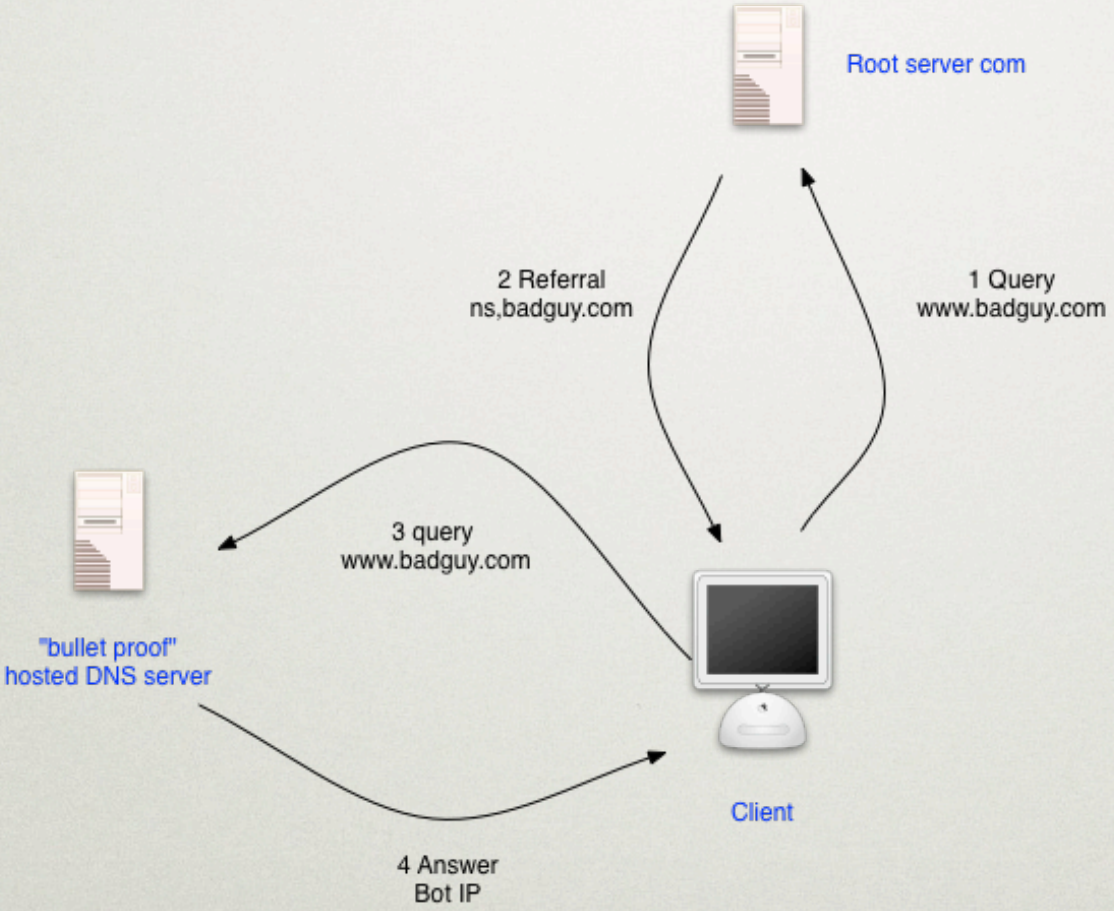
Single Fast Flux

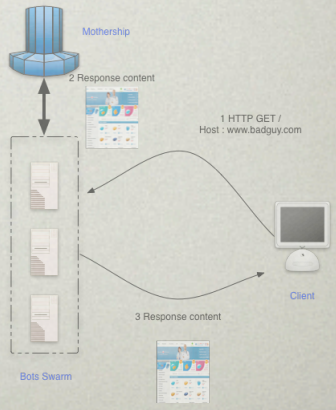




normal hosting

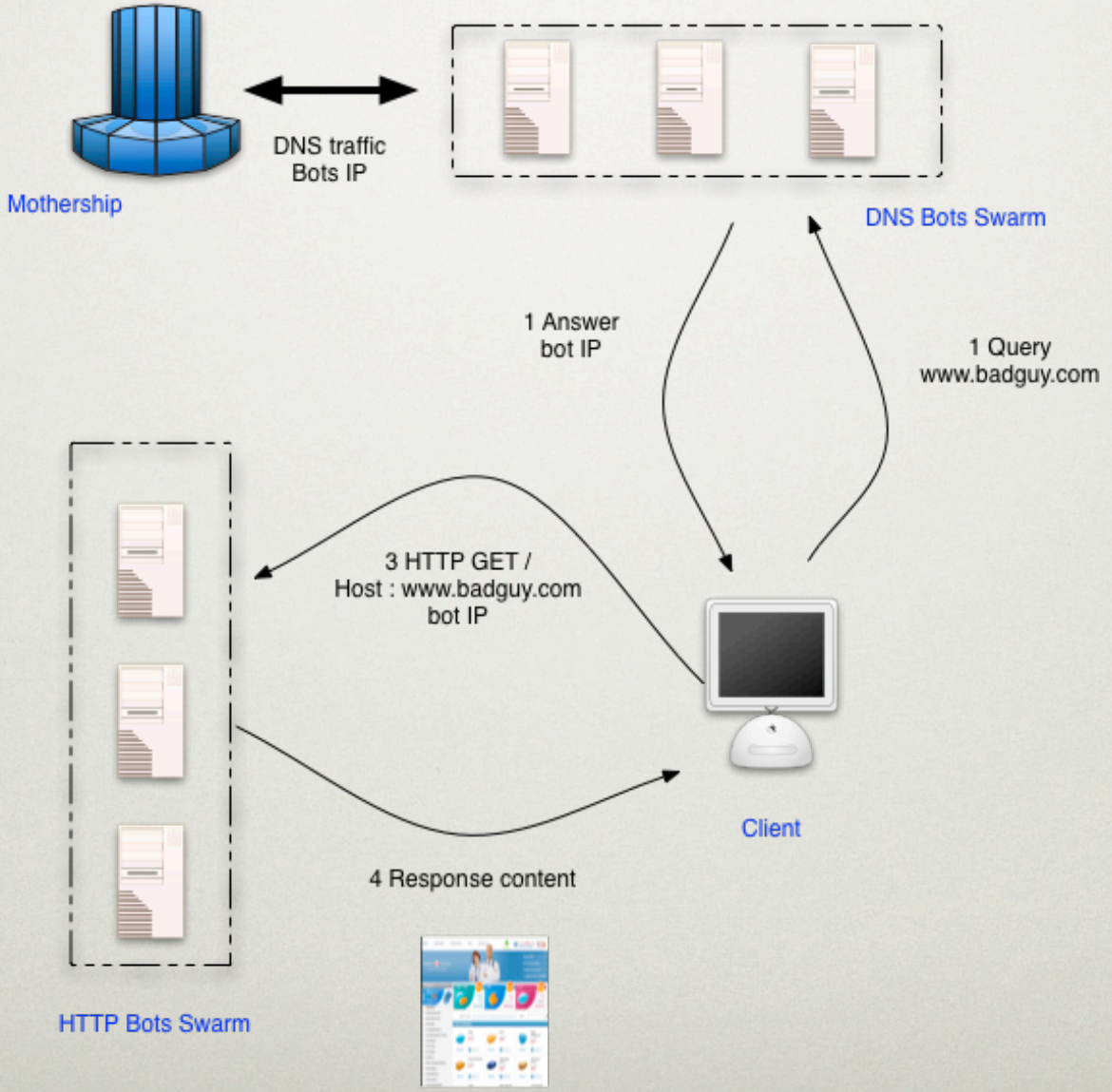
DNS

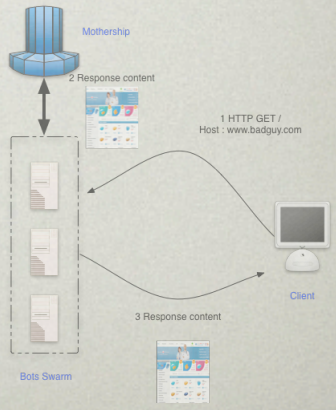




Simple flux

Double Fast Flux





Simple flux

Real world Fast flux

;; WHEN: Wed Apr 4 18:47:50 2007

login.mylspacee.com. 177 IN A 66.229.133.xxx [c-66-229-133-xxx.hsd1.fl.comcast.net]

login.mylspacee.com. 177 IN A 67.10.117.xxx [cpe-67-10-117-xxx.gt.res.rr.com]

login.mylspacee.com. 177 IN A 70.244.2.xxx [adsl-70-244-2-xxx.dsl.hrlntx.swbell.net]

login.mylspacee.com. 177 IN A 74.67.113.xxx [cpe-74-67-113-xxx.stny.res.rr.com]

login.mylspacee.com. 177 IN A 74.137.49.xxx [74-137-49-xxx.dhcp.insightbb.com]

mylspacee.com. 108877 IN NS ns3.myheroisyourslove.hk.

mylspacee.com. 108877 IN NS ns4.myheroisyourslove.hk.

mylspacee.com. 108877 IN NS ns5.myheroisyourslove.hk.

mylspacee.com. 108877 IN NS ns1.myheroisyourslove.hk.

mylspacee.com. 108877 IN NS ns2.myheroisyourslove.hk.

ns1.myheroisyourslove.hk.854 IN A 70.227.218.xxx [ppp-70-227-218-xxx.dsl.sfldmi.ameritech.net]

ns2.myheroisyourslove.hk.854 IN A 70.136.16.xxx [adsl-70-136-16-xxx.dsl.bumttx.sbcglobal.net]

ns3.myheroisyourslove.hk. 854 IN A 68.59.76.xxx [c-68-59-76-xxx.hsd1.al.comcast.net]

WEB ROTATION ~4 MN LATER

;; WHEN: Wed Apr 4 18:51:56 2007 (~4 minutes/186 seconds later)

login.mylspacee.com. 161 IN A 74.131.218.xxx [74-131-218-xxx.dhcp.insightbb.com] NEW

login.mylspacee.com. 161 IN A 24.174.195.xxx [cpe-24-174-195-xxx.elp.res.rr.com] NEW

login.mylspacee.com. 161 IN A 65.65.182.xxx [adsl-65-65-182-xxx.dsl.hstntx.swbell.net] NEW

login.mylspacee.com. 161 IN A 69.215.174.xxx [ppp-69-215-174-xxx.dsl.ipltin.ameritech.net] NEW

login.mylspacee.com. 161 IN A 71.135.180.xxx [adsl-71-135-180-xxx.dsl.pltn13.pacbell.net] NEW

mylspacee.com. 108642 IN NS ns3.myheroisyourslove.hk.

mylspacee.com. 108642 IN NS ns4.myheroisyourslove.hk.

mylspacee.com. 108642 IN NS ns5.myheroisyourslove.hk.

mylspacee.com. 108642 IN NS ns1.myheroisyourslove.hk.

mylspacee.com. 108642 IN NS ns2.myheroisyourslove.hk.

ns1.myheroisyourslove.hk. 608 IN A 70.227.218.xxx [ppp-70-227-218-xxx.dsl.sfldmi.ameritech.net]

ns2.myheroisyourslove.hk. 608 IN A 70.136.16.xxx [adsl-70-136-16-xxx.dsl.bumttx.sbcglobal.net]

ns3.myheroisyourslove.hk. 608 IN A 68.59.76.xxx [c-68-59-76-xxx.hsd1.al.comcast.net]

NS rotation ~90mn later

;; WHEN: Wed Apr 4 21:13:14 2007 (~90 minutes/4878 seconds later)

ns1.myheroisyourslove.hk. 3596 IN A 75.67.15.xxx [c-75-67-15-xxx.hsd1.ma.comcast.net] NEW

ns2.myheroisyourslove.hk. 3596 IN A 75.22.239.xxx [adsl-75-22-239-xxx.dsl.chcgil.sbcglobal.net] NEW

ns3.myheroisyourslove.hk. 3596 IN A 75.33.248.xxx [adsl-75-33-248-xxx.dsl.chcgil.sbcglobal.net] NEW

ns4.myheroisyourslove.hk. 180 IN A 69.238.210.xxx [ppp-69-238-210-xxx.dsl.irvnca.pacbell.net] NEW

ns5.myheroisyourslove.hk. 3596 IN A 70.64.222.xxx [xxx.mj.shawcable.net] NEW

Detection / Mitigation

- Fast Flux are very “noisy”
  - Many A name
  - Quick rotation
  - Many NS
  - Quick rotation

Worms
Generation 2

Outline

- Worm Generation 1
- Botnet
- Fast Flux
- Worm Generation 2
- Underground Economy

Conficker 2008-2009

- Most important Worm since Slammer
- 4 years have passed..
- Vulnerability in Server Service
- 2000, XP, Vista, 2003, and 2008

Windows of Vulnerability

- Found in the wild
- Announced by MS 22 Oct 2008
- Out of band patch 26 Oct 2008
- Public Exploit 26 Oct 2008
- Conficker : Early november

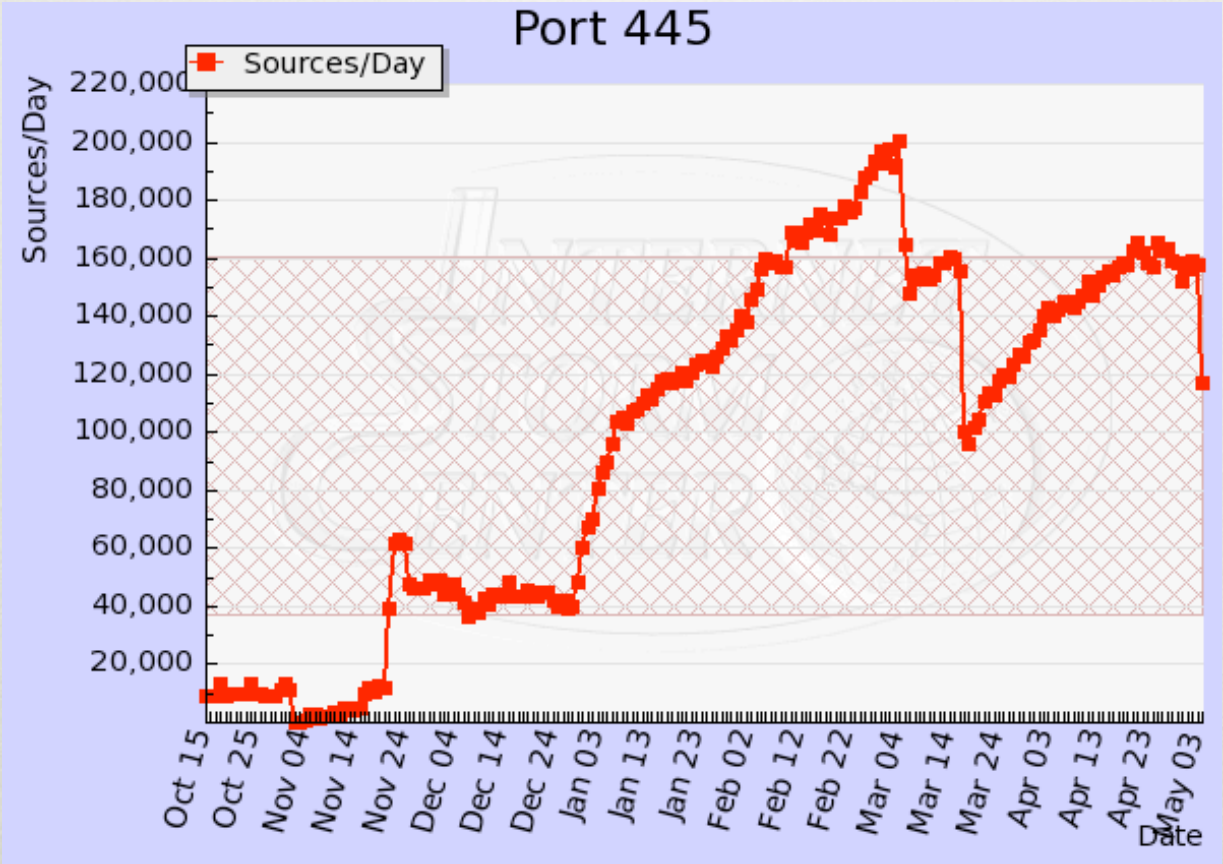
Tech details

- Buffer overflow in the RPC code
- Port 139 / 445
- Neeris did adopt it as well (Apr 09)
- First version dev by chinese hackers (37$)

Tech Details 2

- Use a non standard overflow
- Use a fixed shellcode
- Re-infection is used to update binary
- Blacklist Ukrainian ISP / Language
- Use named mutex for version conflict
- Use HTTP request to popular domains for time sync (A / B)

Port activity



Numbers

- Total IP Addresses: 10,512,451
- Total Conficker A IPs: 4,743,658
- Total Conficker B IPs: 6,767,602
- Total Conficker AB IPs: 1,022,062

Conficker A 2008-11-21

- **Infection** : Netbios MS08-067
- **propagation** HTTP pull / 250 rand / 8 TLD
- **Defense** : N/A
- **End usage** : update to version B,C or D

Conficker B 2008-12-29

- **Infection** :
  - Netbios MS08-067
  - Removable Media via DLL
- **propagation**
  - HTTP pull / 250 rand / 8 TLD
  - Netbios Push : patch for reinjection
- **Defense** :
  - Blocks DNS lookups
  - Disables AutoUpdate
- **End usage** : update to version C or D

Difference between B/C

- Designed to counter counter-measure
- 15% of the original B code base untouched
- New thread architecture
- P2P addition

Conficker C 2009-03-04

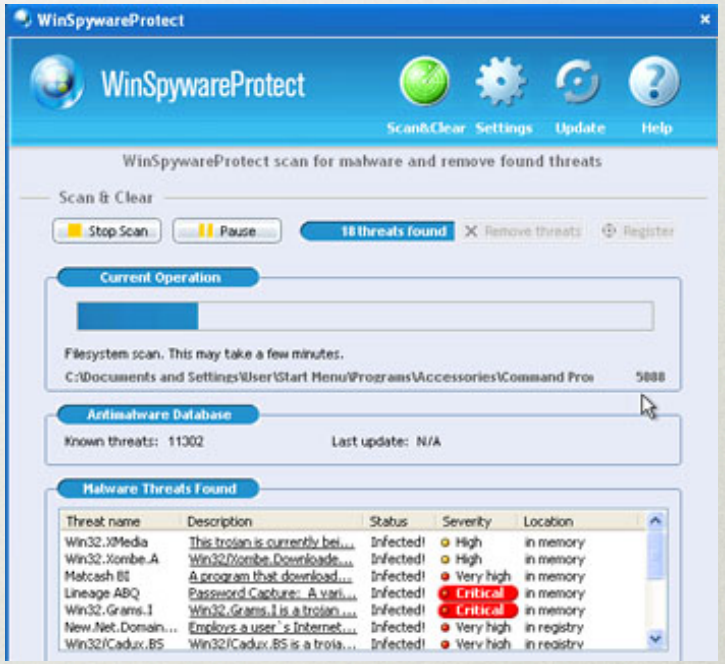
- **Infection** :
  - Netbios MS08-067
  - Removable Media via DLL
  - Dictionary attack on $Admin
- **propagation**
  - HTTP pull / 250 rand / 8 TLD
  - Netbios Push : patch for reinjection
  - Create named pipe
- **Defense** :
  - Blocks DNS lookups
  - Disables AutoUpdate

Conficker D 2009-03-04

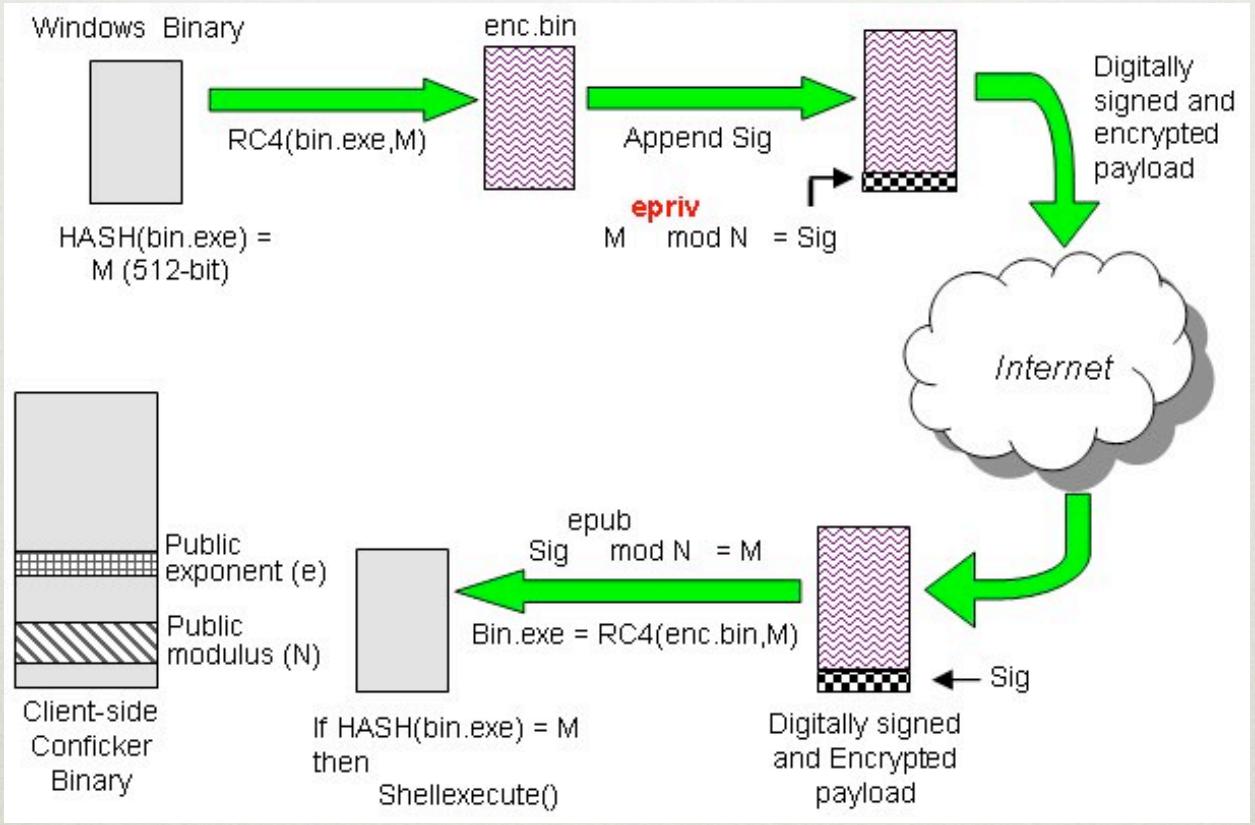
- **propagation**
  - HTTP pull / 50 000 rand / 110 TLD
  - P2P push / pull custom protocol
- **Defense** :
  - Disables Safe Mode
  - Kills anti-malware
  - in-memory patch of DNSAPI.DLL to block lookups of anti-malware related web sites
- **End usage** : update to version E

Conficker E 2009-07-04

- Downloads and installs additional malware:
  - Waledac spambot
  - SpyProtect 2009 scareware
- Removes self on 3 May 2009 (Does not remove accompanying copy of W32.Downadup.C) [37]

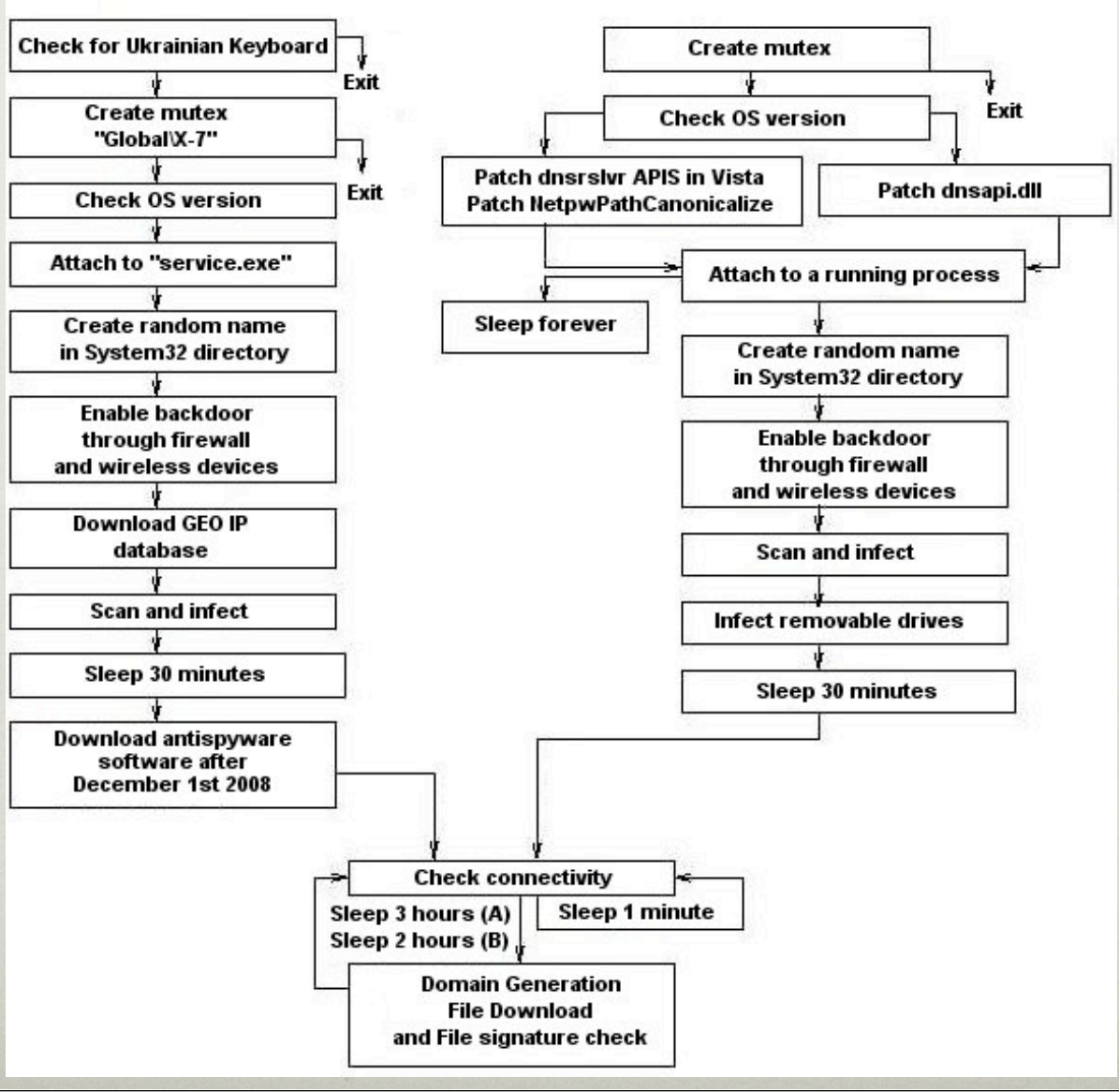


Binary Security



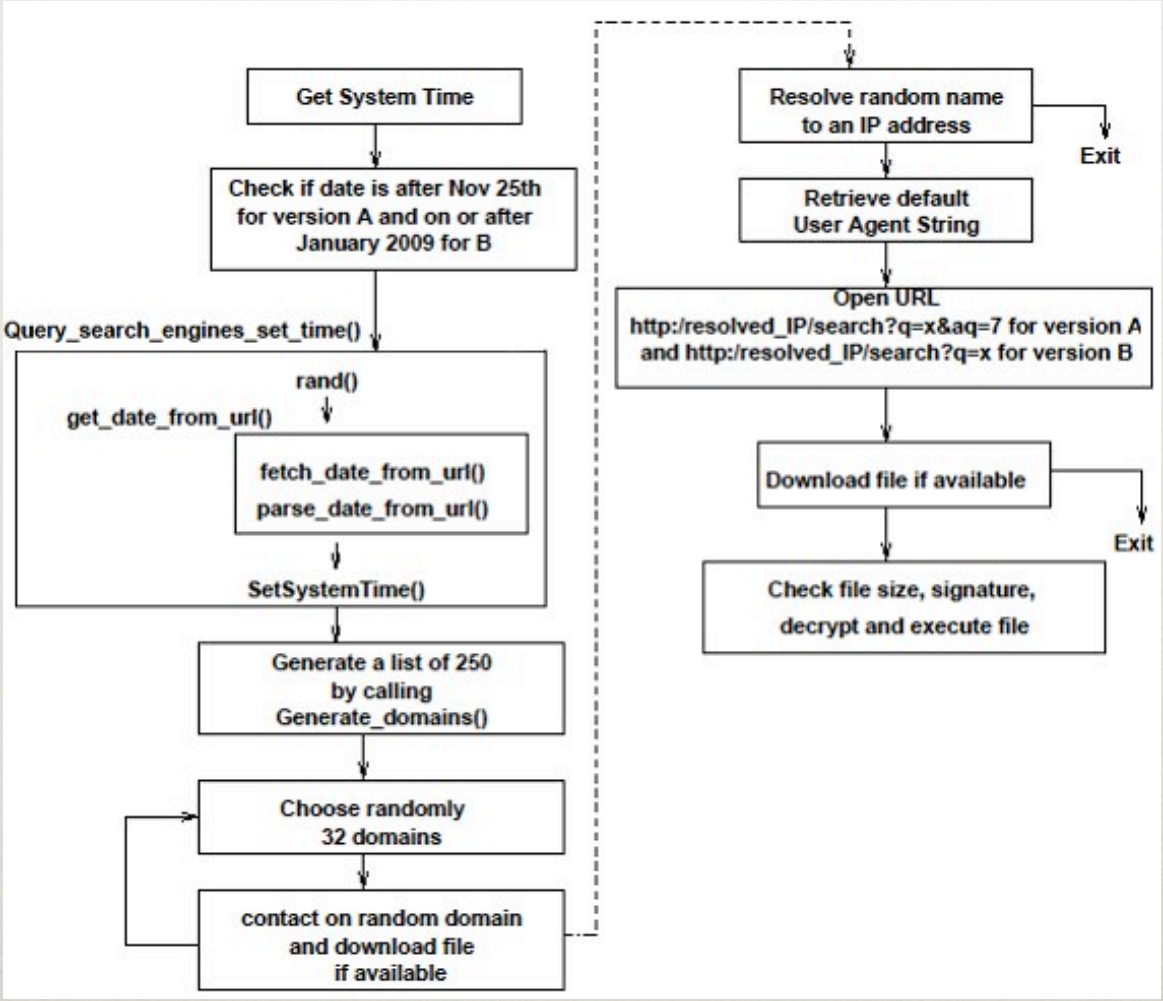
SRI

Conficker A/B logic



SRI

Rendez vous point



SRI

What does it take to build such code

- Internet-wide programming skill
- advanced cryptographic skill
- custom dual-layer code packing
- code obfuscation skills
- in-depth knowledge of Windows internals and security products.

Underground Economy

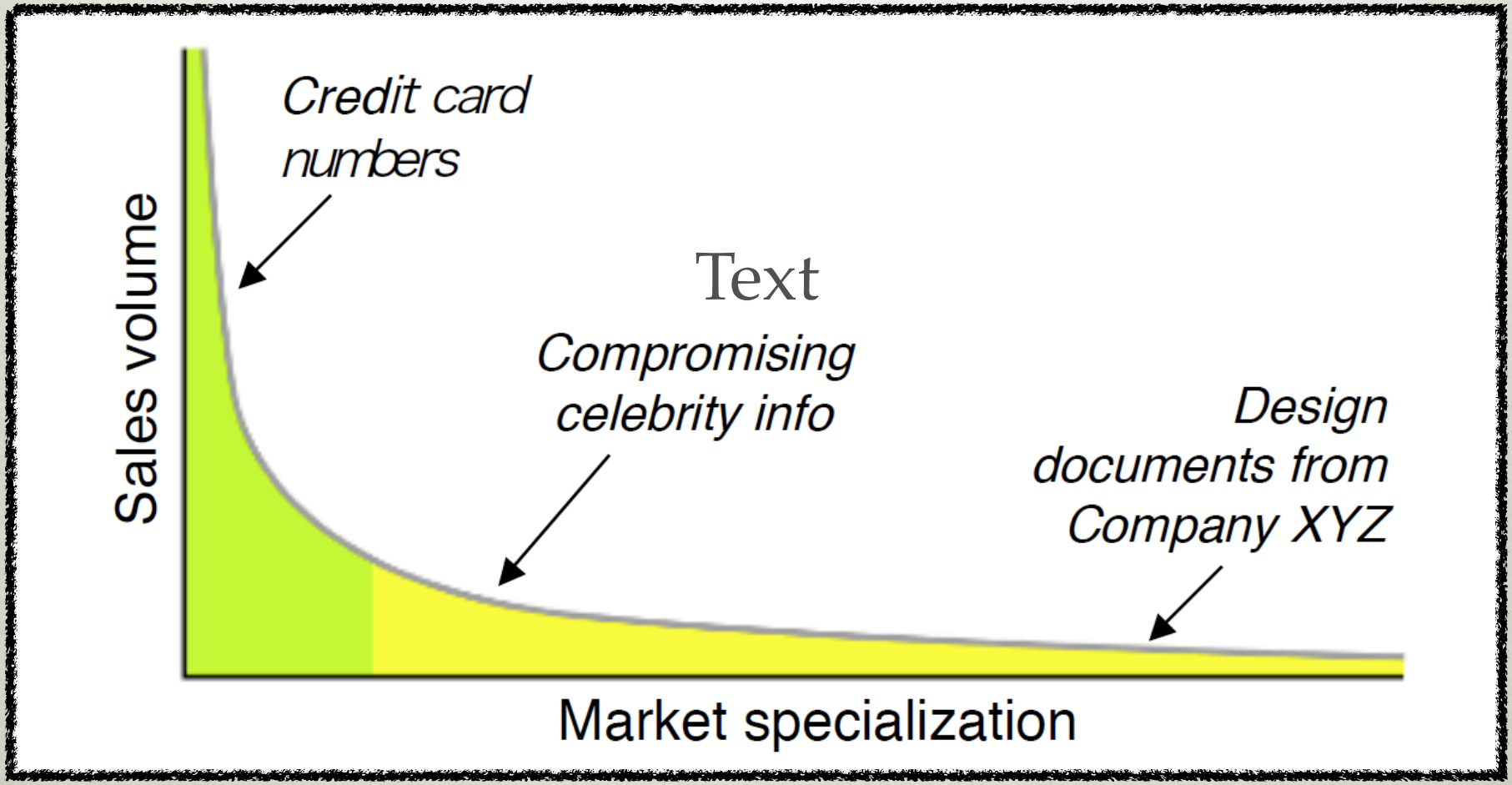
Outline

- Worm Generation 1
- Botnet
- Fast Flux
- Worm Generation 2
- Underground Economy

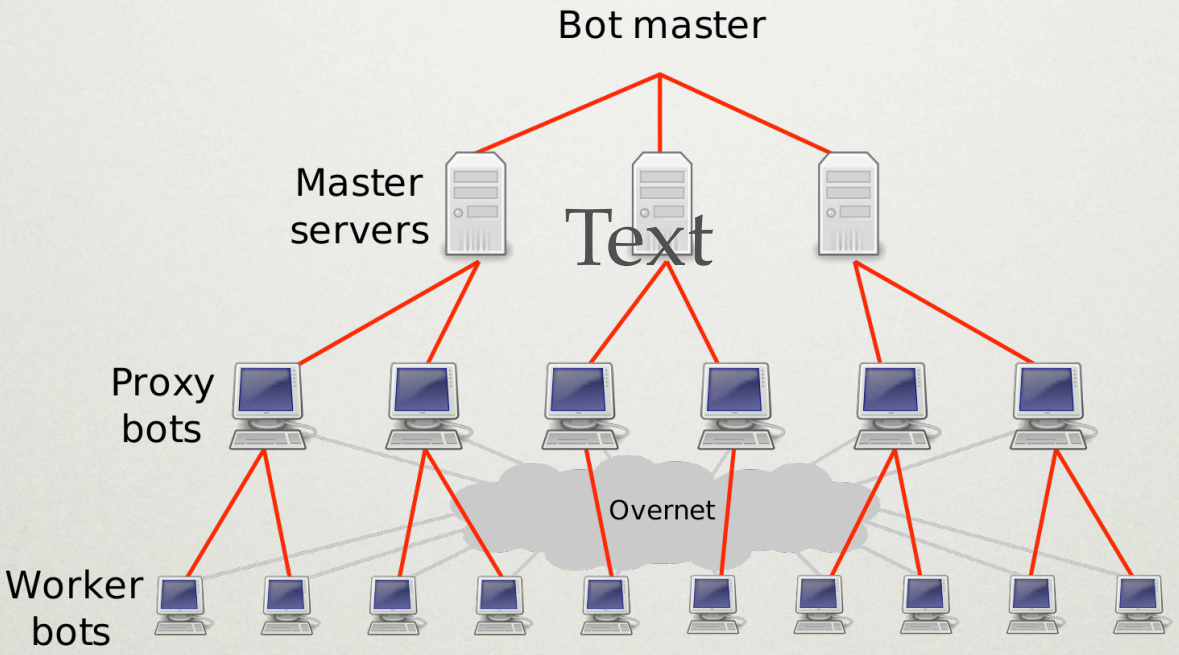
Illicit Activities

- D-DOS
- Extortion
- Identity theft
- Warez hosting
- Spam
- Phising
- Click fraud
- malware distribution

Long Tail application



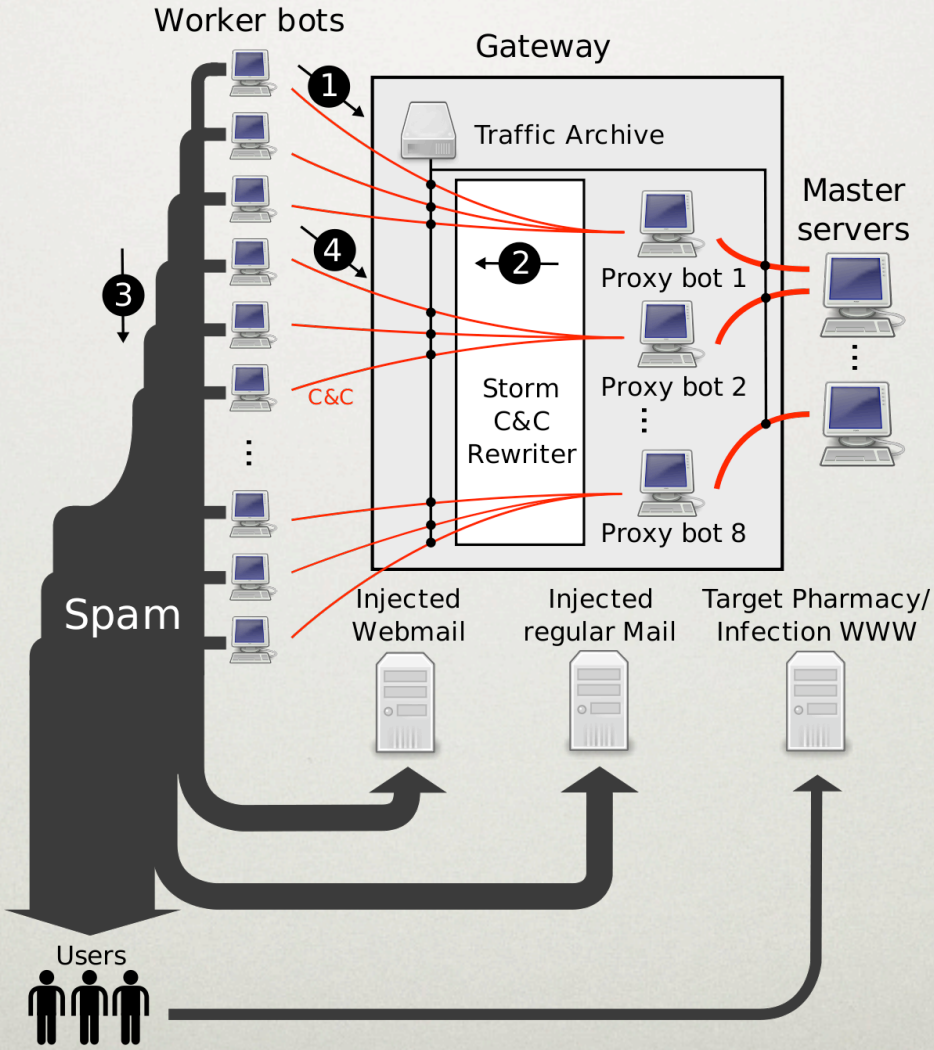
Storm architecture



Spamalytics: An Empirical Analysis of Spam Marketing Conversion

Chris Kanich* Christian Kreibich† Kirill Levchenko* Brandon Enright* Geoffrey M. Voelker* Vern Paxson† Stefan Savage*

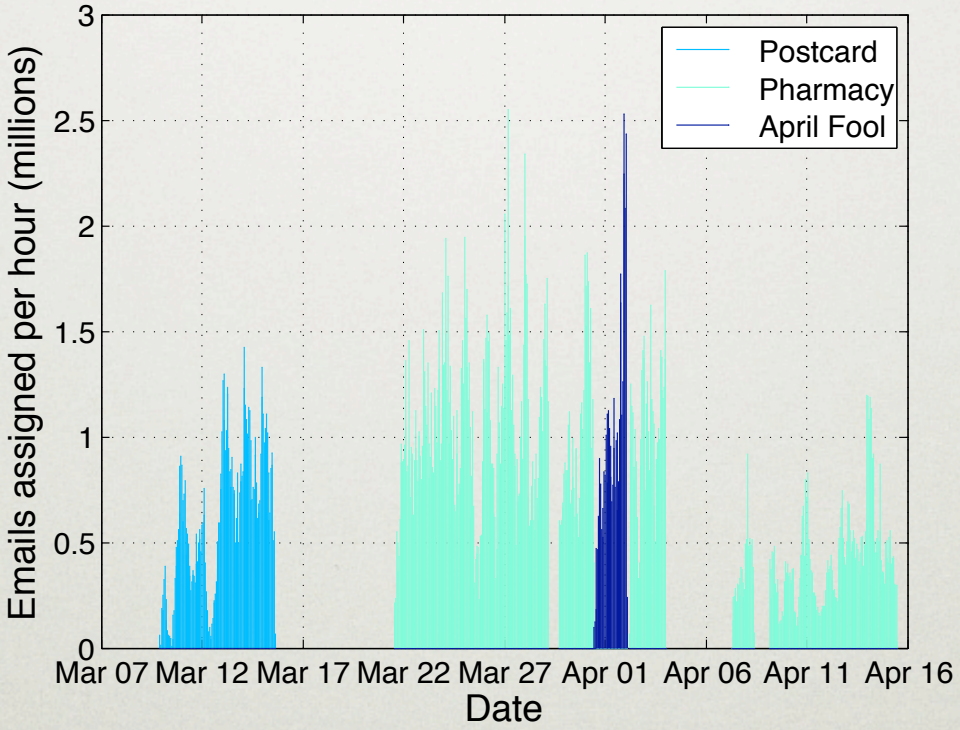
Spam craft



Spamalytics: An Empirical Analysis of Spam Marketing Conversion

Chris Kanich* Christian Kreibich† Kirill Levchenko* Brandon Enright* Geoffrey M. Voelker* Vern Paxson† Stefan Savage*

Spam stat



Spamalytics: An Empirical Analysis of Spam Marketing Conversion

Chris Kanich* Christian Kreibich† Kirill Levchenko* Brandon Enright* Geoffrey M. Voelker* Vern Paxson† Stefan Savage*

CAMPAIGN	DATES	WORKERS	E-MAILS
Pharmacy	Mar 21 – Apr 15	31,348	347,590,389
Postcard	Mar 9 – Mar 15	17,639	83,665,479
April Fool	Mar 31 – Apr 2	3,678	38,651,124
		Total	469,906,992

Spamalytics: An Empirical Analysis of Spam Marketing Conversion

Chris Kanich* Christian Kreibich† Kirill Levchenko* Brandon Enright* Geoffrey M. Voelker* Vern Paxson† Stefan Savage*

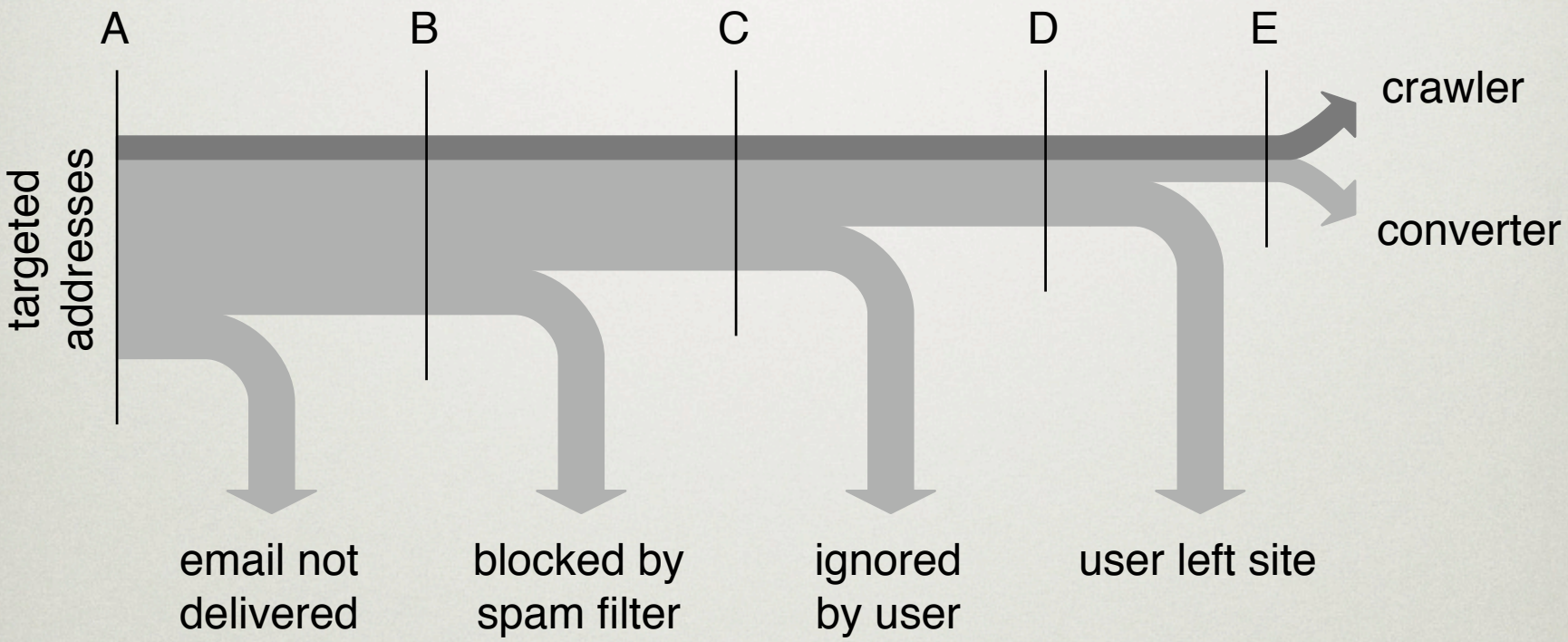
Domain repartion

Domain	Freq.
hotmail.com	8.47%
yahoo.com	5.05%
gmail.com	3.17%
aol.com	2.37%
yahoo.co.in	1.13%
sbcglobal.net	0.93%
mail.ru	0.86%
shaw.ca	0.61%
wanadoo.fr	0.61%
msn.com	0.58%
Total	23.79%

Spamalytics: An Empirical Analysis of Spam Marketing Conversion

Chris Kanich* Christian Kreibich† Kirill Levchenko* Brandon Enright* Geoffrey M. Voelker* Vern Paxson† Stefan Savage*

Spam pipeline



Spamalytics: An Empirical Analysis of Spam Marketing Conversion

Chris Kanich* Christian Kreibich† Kirill Levchenko* Brandon Enright* Geoffrey M. Voelker* Vern Paxson† Stefan Savage*

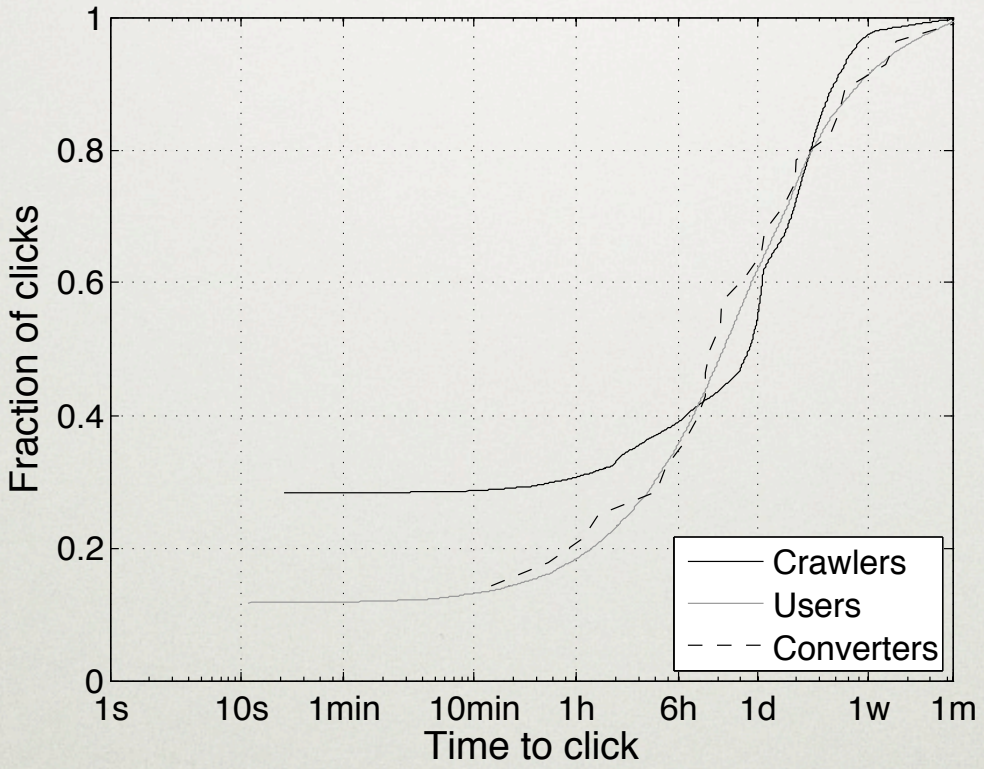
Percentage

Stage	Pharmacy		Postcard		April Fool	
A – Spam Targets	347,590,389	100%	83,655,479	100%	40,135,487	100%
B – MTA Delivery (est.)	82,700,000	23.8%	21,100,000	25.2%	10,100,000	25.2%
C – Inbox Delivery	—	—	—	—	—	—
D – User Site Visits	10,522	0.00303%	3,827	0.00457%	2,721	0.00680%
E – User Conversions	28	0.0000081%	316	0.000378%	225	0.000561%

Spamalytics: An Empirical Analysis of Spam Marketing Conversion

Chris Kanich* Christian Kreibich† Kirill Levchenko* Brandon Enright* Geoffrey M. Voelker* Vern Paxson† Stefan Savage*

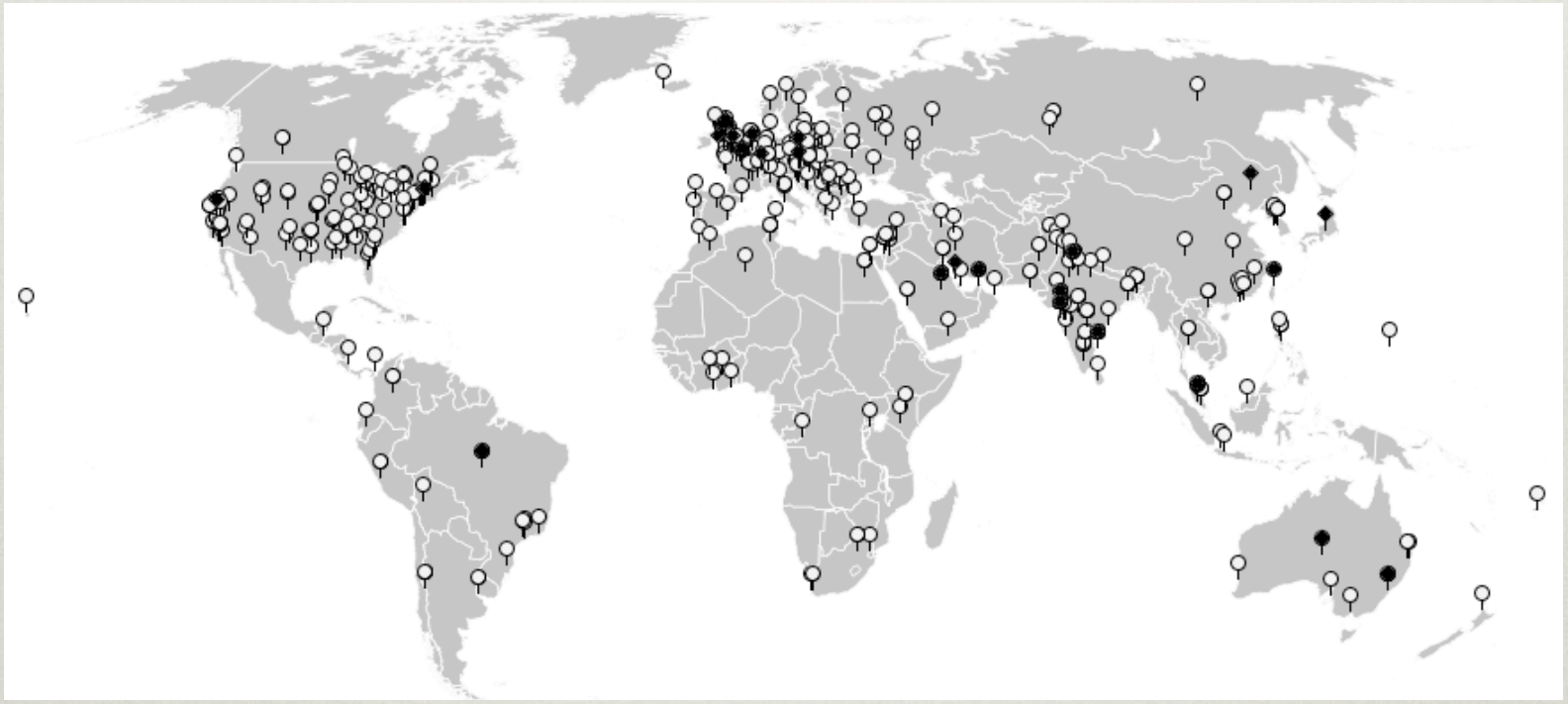
Click response time



Spamalytics: An Empirical Analysis of Spam Marketing Conversion

Chris Kanich* Christian Kreibich† Kirill Levchenko* Brandon Enright* Geoffrey M. Voelker* Vern Paxson† Stefan Savage*

Geographic Repartition



Spamalytics: An Empirical Analysis of Spam Marketing Conversion

Chris Kanich* Christian Kreibich† Kirill Levchenko* Brandon Enright* Geoffrey M. Voelker* Vern Paxson† Stefan Savage*