CSCE 465 Computer & Network Security

Instructor: Abner Mendoza

Recap

- Homework?
- Buffer Overflows
- Cybersecurity Club

Malware

Roadmap

- Malware basics
- Worms and examples
- Botnets and examples

Malicious Programs

- Needs host program
  - trap doors
  - logic bombs
  - Trojan horses
  - Viruses
- Independent
  - Worms
  - bots

Malware

- Malicious software
  - Trojan horses, virus, worms,.. Etc
- Today's malware is all about stealth
- Infected machines report back to attacker
- Attacker uses backdoor to control the infected machine…. Make it a *zombie.* A collection of zombies is called a *botnet*

Malware

- Criminals can rent out botnets
- Keyloggers
- Identity theft
- Malware can lay in wait for something interesting
- Malware can interfere with competition's production process
- Malware could target another person in the company to discredit that person

Types of Malware

- Trojan Horse
- Virus
- Worm
- Spyware
- RootKits

Trojan Horse

- Transport means…Getting victims to download virus without attacker's intervention.

- Now you have to get the victim to run it

Trojan Horse

- Download program somewhere in users PATH.

(Find directory not secured)

- Pick a name of a mistyped command ‘la’. If the user mistypes ‘ls’ as ‘la’, the Trojan will run.

Trojan Horse

- Legitimate, but malicious, user
- Puts an infected version of ‘ls’ on the system.
- Call admin……

  cd/home/mal

  ls –l

- Admin just ran Trojan with superuser privileges

Viruses

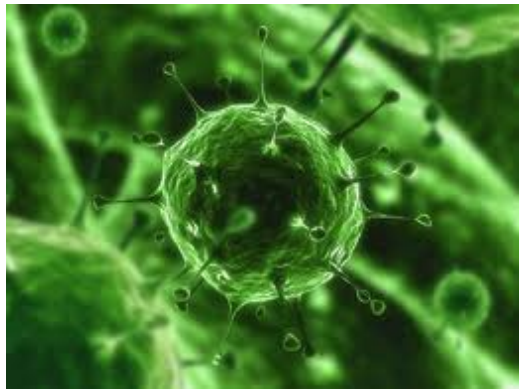
- Virus is a program that can reproduce itself by attaching its code to another program.
- Often written in assembler or C.
- Attacker infects a program on his own machine, then gets that program distributed.
- Once installed on victim's machine, it remains dormant until executed.

Virus

- Once *activated*…
  - Executes it **payload**
  - Often waits for a specific date or time
  - …. We want to make sure the virus is well distributed before people start noticing it.

Viruses

- “Infect” a program by modifying it
- Self-copied into the program to spread
- Four stages:
  - dormant phase
  - propagation phase
    - E.g., attachment to email
  - triggering phase
  - execution phase



Virus Structure

- First line: go to “main” of virus program
- Second line: a special mark (infected or not)
- Main:
  - find uninfected programs
    - infect and mark them
  - do something damaging to the system
  - now “go to” the first line of the original program
    - appear to do the normal work
- Avoid detection by looking at size of program:
  - compress/decompress the original program

Types of Viruses

- Parasitic virus
  - search and infect executable files
- Memory-resident virus
  - infect running programs
- Boot sector virus
  - spreads whenever the system is booted
- Polymorphic virus
  - encrypt part of the virus program using randomly generated key

Macro Viruses

- Macro
  - an executable program (e.g., opening a file, starting an application) embedded in a word processing document, e.g. MS Word
- Common technique for spreading
  - A virus macro is attached to a Word document
  - Document is loaded and opened in the local system
  - When the macro executes, it copies itself to the global macro file
  - The global macro can be activated/spread when new documents are opened.

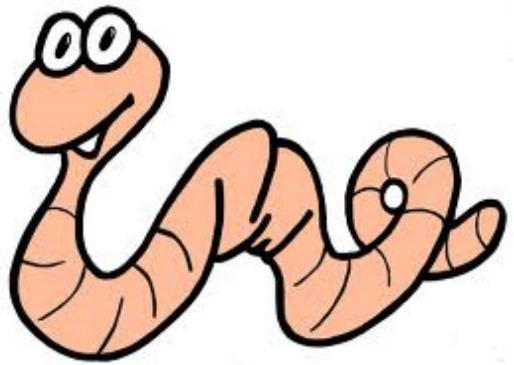
Myths about Viruses

- Can only infect Microsoft Windows
- Can modify hidden and read-only files
- Spread only on disks or in email
- Cannot remain in memory after reboot
- Cannot infect hardware
- Can be malevolent, benign, or benevolent

Antivirus Approach

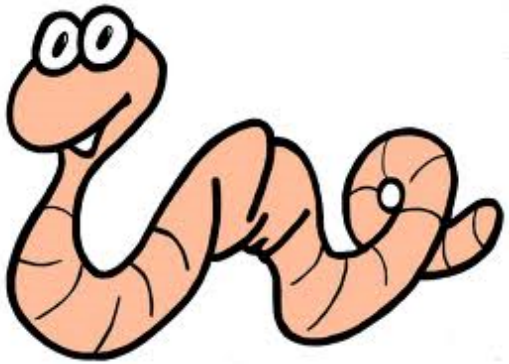
- Prevention
  - Limit contact to outside world
- Detection and identification
- Removal
- 4 generations of antivirus software
  - simple scanners
    - use “signatures” of known viruses
  - heuristic scanners
    - integrity checking: checksum, encrypted hash
  - activity traps
  - full-featured protection

WORMS AND EXAMPLES



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

The Internet Worm

- What it did
  - Determine where it could spread
  - Spread its infection
  - Remain undiscovered and undiscoverable
- Effect
  - Resource exhaustion – repeated infection due to a programming bug
  - Servers are disconnected from the Internet by sys admin to stop infection

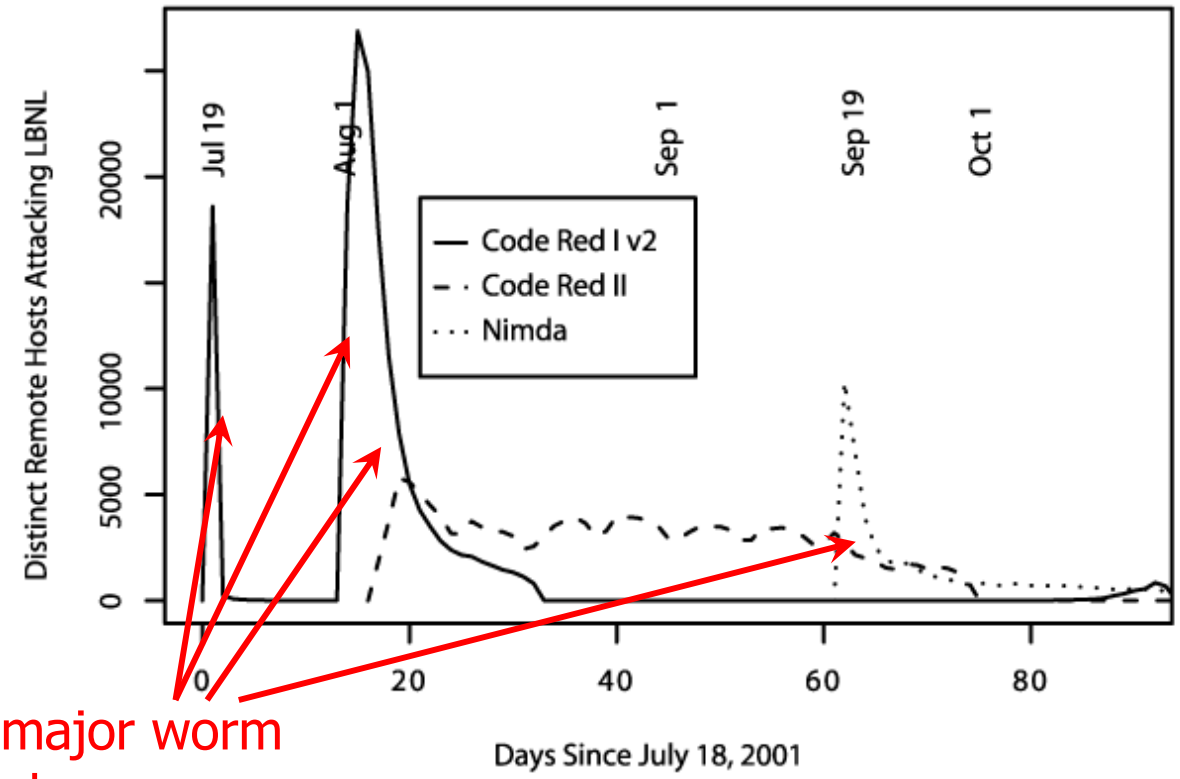
The Internet Worm

- How it worked
  - Where to spread
    - Exploit security flaws
      - Guess password (encrypted passwd file readable)
      - fingerd: buffer overflow
      - sendmail: backdoor (accepts shell commands)
  - Spread
    - Bootstrap loader to target machine, then fetch rest of code (password authenticated)
  - Remain undiscoverable
    - Load code in memory, encrypt, remove file
    - Periodically changed name and process ID

Morris Worm Redux

- 1988: No malicious payload, but bogged down infected machines by uncontrolled spawning
  - Infected 10% of all Internet hosts at the time
- Multiple propagation vectors
  - Remote execution using rsh and cracked passwords
    - Tried to crack passwords using small dictionary and publicly readable password file; targeted hosts from /etc/hosts.equiv
  - Buffer overflow in fingerd on VAX
    - Standard stack smashing exploit
  - DEBUG command in Sendmail
    - In early Sendmail versions, possible to execute a command on a remote machine by sending an SMTP (mail transfer) message

Summer of 2001



Three major worm outbreaks

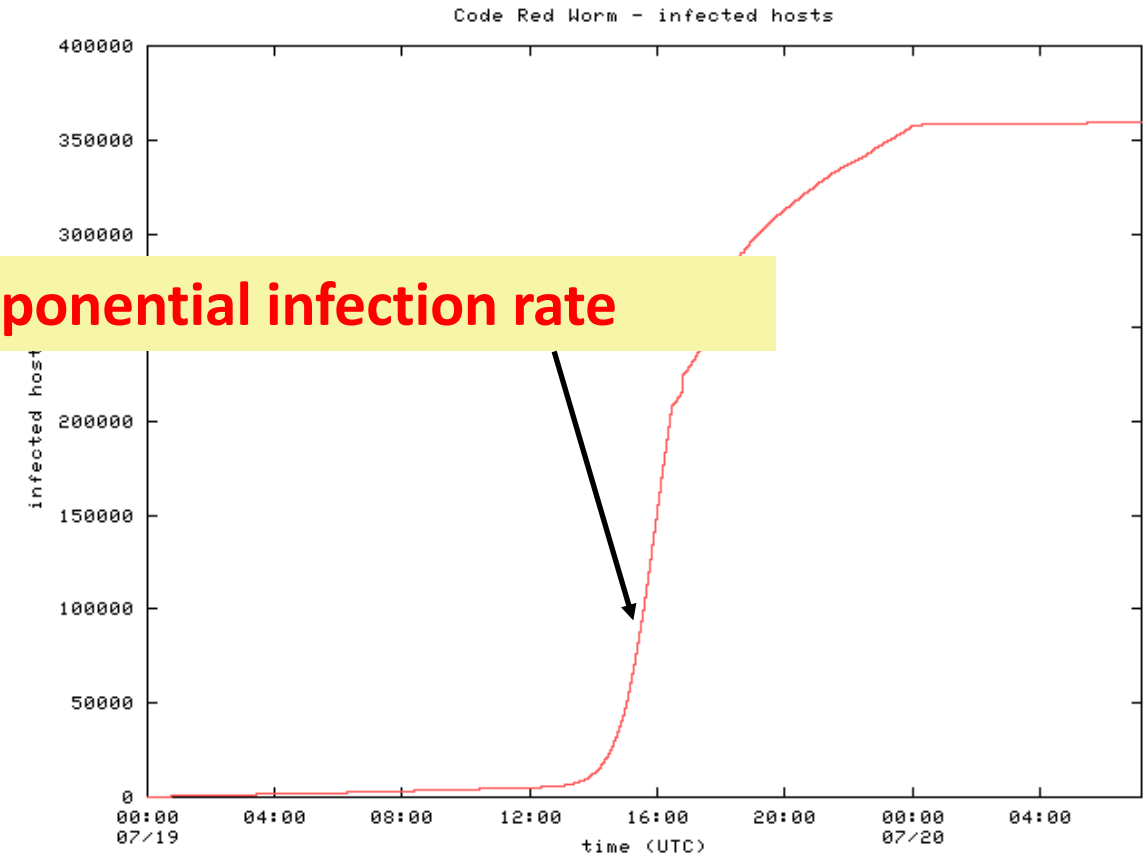
Code Red I

- July 13, 2001: First worm of the modern era
- Exploited buffer overflow in Microsoft's Internet Information Server (IIS)
- 1st through 20th of each month: spread
  - Find new targets by random scan of IP address space
    - Spawn 99 threads to generate addresses and look for IIS
  - Creator forgot to seed the random number generator, and every copy scanned the same set of addresses ☺
- 21st through the end of each month: attack
  - Deface websites with **"HELLO! Welcome to http://www.worm.com! Hacked by Chinese!"**

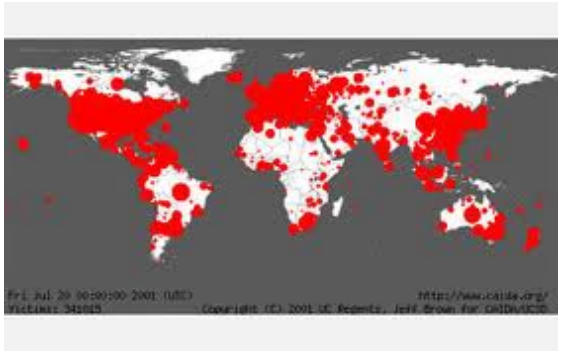
Code Red I v2

- July 19, 2001: Same codebase as Code Red I, but fixed the bug in random IP address generation
  - Compromised all vulnerable IIS servers on the Internet
  - Large vulnerable population meant fast worm spread
    - Scanned address space grew exponentially
    - 350,000 hosts infected in 14 hours!!
- Payload: distributed packet flooding (denial of service) attack on www.whitehouse.gov
  - Coding bug causes it to die on the 20th of each month… but if victim's clock is wrong, resurrects on the 1st

Code Red: Host Infection Rate



Exponential infection rate



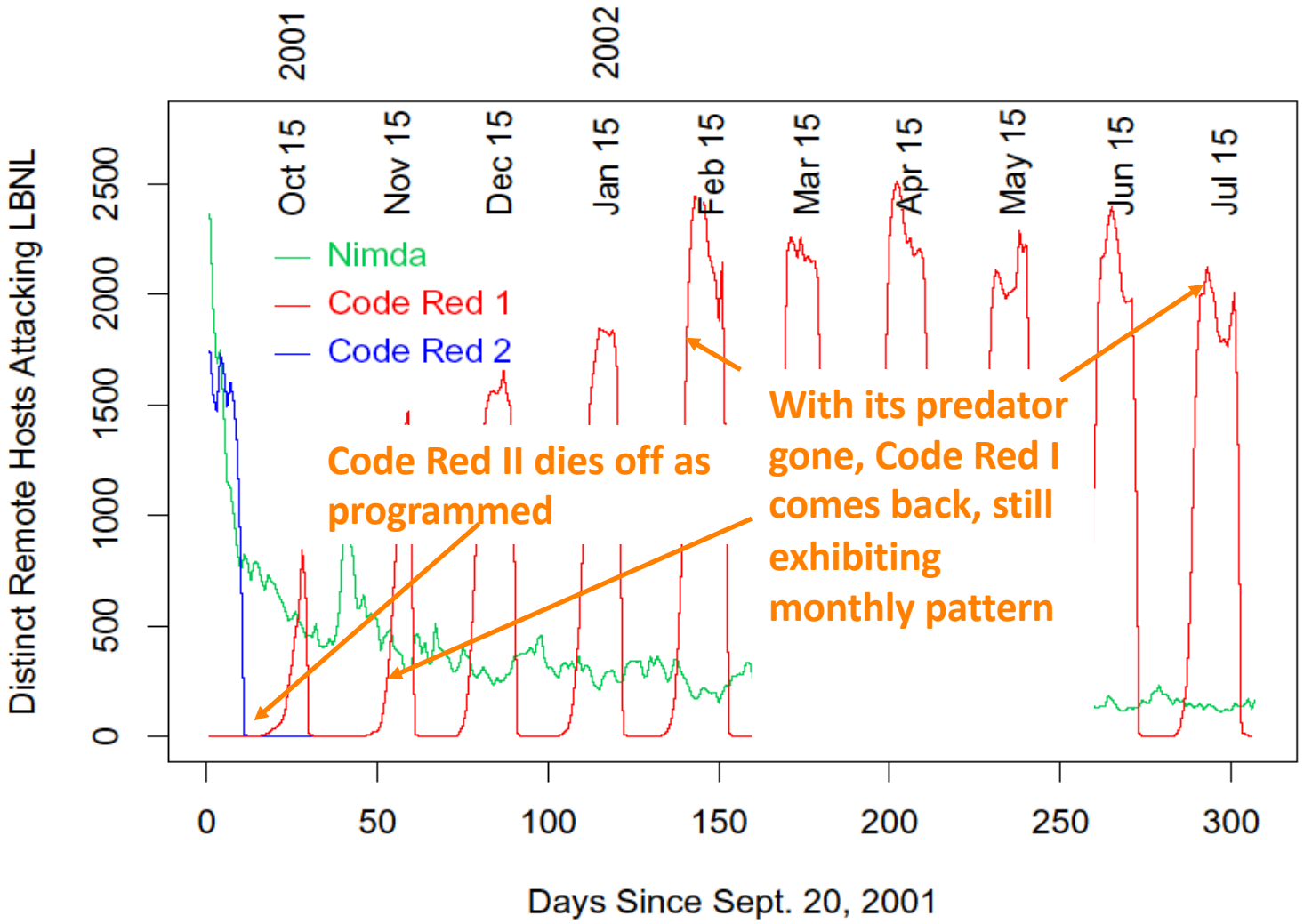
Designing Fast-Spreading Worms

- **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

Code Red II

- August 4, 2001: Same IIS vulnerability, completely different code, kills Code Red I
  - Known as "Code Red II" because of comment in code
  - Worked only on Windows 2000, crashed NT
- Scanning algorithm preferred nearby addresses
  - Chose addresses from same class A with probability ½, same class B with probability 3/8, and randomly from the entire Internet with probability 1/8
- Payload: installed root backdoor in IIS servers for unrestricted remote access
- Died by design on October 1, 2001

Code Red I and II (Paxson)



Nimda

- September 18, 2001: Multi-modal worm using several propagation vectors
  - Exploit same IIS buffer overflow as Code Red I and II
  - Bulk-email itself as an attachment to email addresses harvested from infected machines
  - Copy itself across open network shares
  - Add exploit code to Web pages on compromised sites to infect visiting browsers
  - Scan for backdoors left by Code Red II
- Payload: turned-off code deleting all data on hard drives of infected machines

Signature-Based Defenses Don't Help

- Nimda leaped firewalls
- Many firewalls passed mail untouched, relying on mail servers to filter out infections
  - Most filters simply scan attachments for signatures (code snippets) of known viruses and worms
- Nimda was a brand-new infection with unknown signature, and scanners could not detect it
- Big challenge: detection of zero-day attacks
  - When a worm first appears in the wild, signature is not extracted until minutes or hours later

Slammer (Sapphire) Worm

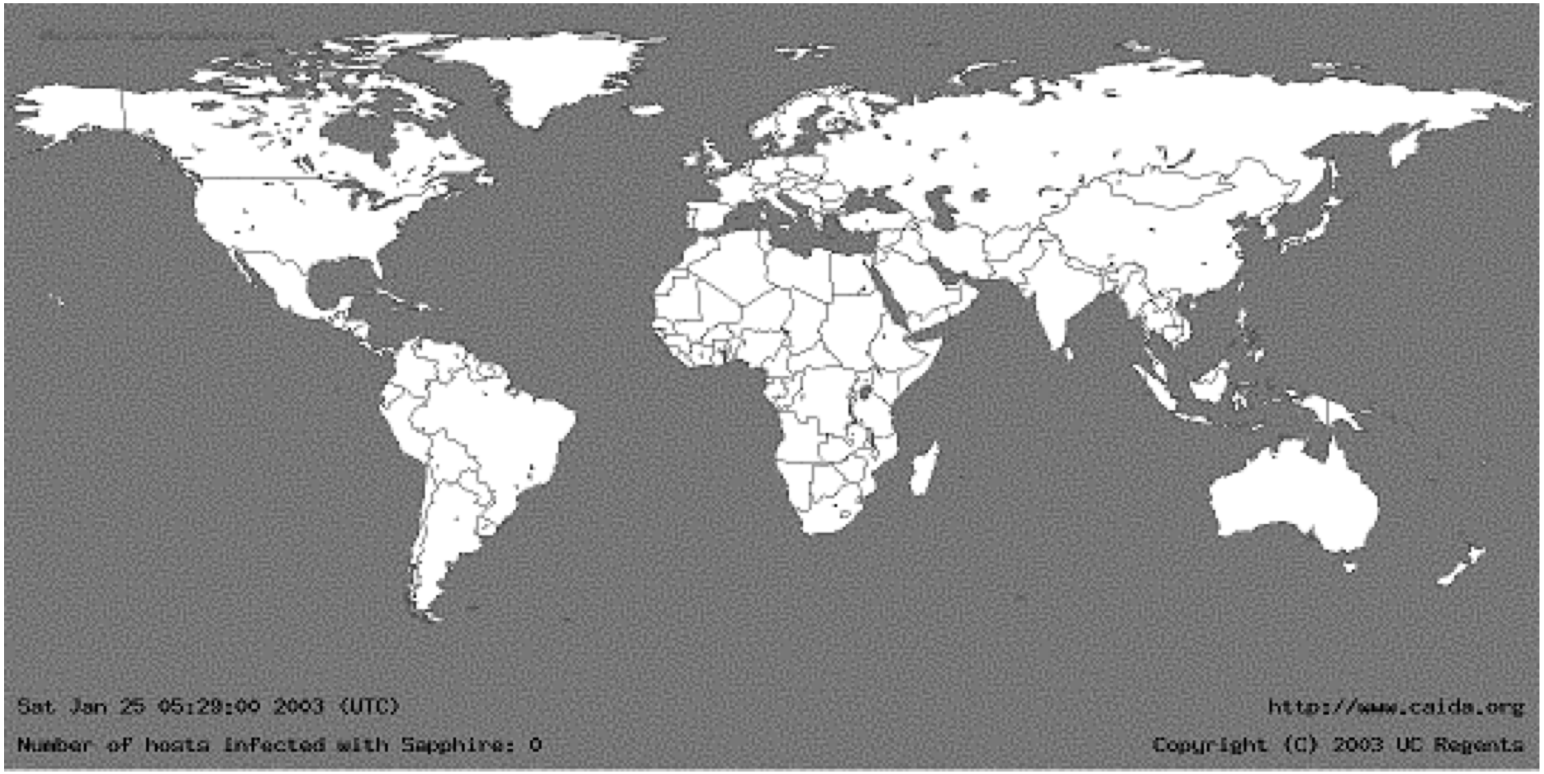
- January 24/25, 2003: UDP worm exploiting buffer overflow in Microsoft's SQL Server
  - Overflow was already known and patched by Microsoft… but not everybody installed the patch
- Entire code fits into a single 404-byte UDP packet
  - Worm binary followed by overflow pointer back to itself
- Classic buffer overflow combined with random scanning: once control is passed to worm code, it randomly generates IP addresses and attempts to send a copy of itself to port 1434
  - MS-SQL listens at port 1434

Slammer Propagation

- Scan rate of 55,000,000 addresses per second
  - Scan rate = rate at which worm generates IP addresses of potential targets
  - Up to 30,000 single-packet worm copies per second
- Initial infection was doubling in 8.5 seconds (!!)
  - Doubling time of Code Red was 37 minutes
- Worm-generated packets saturated carrying capacity of the Internet in 10 minutes
  - 75,000 SQL servers compromised
  - And that's in spite of broken pseudo-random number generator used for IP address generation

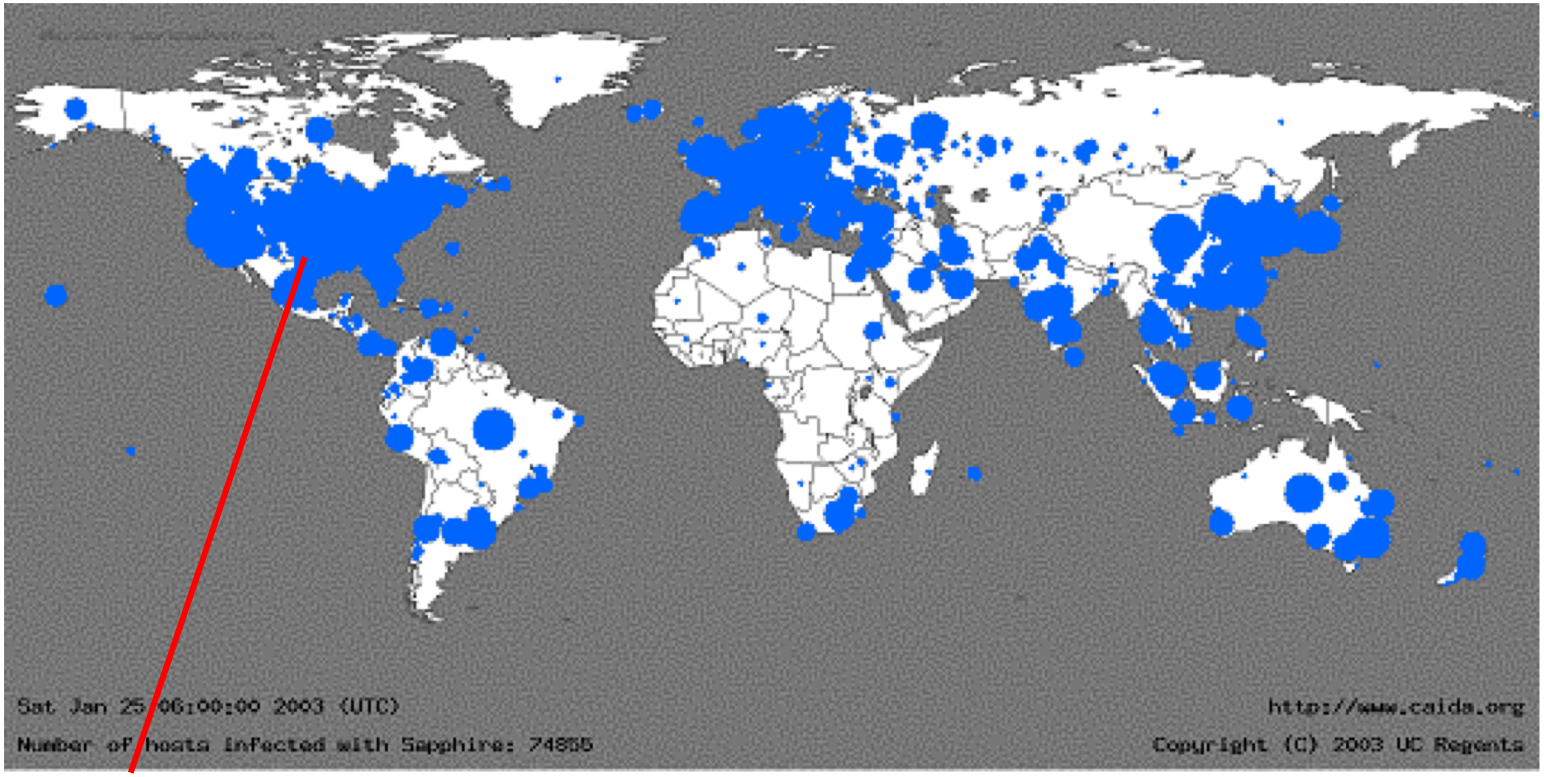
05:29:00 UTC, January 25, 2003

[from Moore et al. "The Spread of the Sapphire/Slammer Worm"]



30 Minutes Later

[from Moore et al. "The Spread of the Sapphire/Slammer Worm"]



Size of circles is **logarithmic** in the number of infected machines

Slammer Impact

- $1.25 Billion of damage
- Temporarily knocked out many elements of critical infrastructure
  - Bank of America ATM network
  - Entire cell phone network in South Korea
  - Five root DNS servers
  - Continental Airlines' ticket processing software
- The worm did not even have malicious payload… simply bandwidth exhaustion on the network and resource exhaustion on infected machines

Secret of Slammer's Speed

- Old-style worms (Code Red) spawn a new thread which tries to establish a TCP connection and, if successful, send a copy of itself over TCP
  - Limited by latency of the network
- Slammer was a connectionless UDP worm
  - No connection establishment, simply send 404-byte UDP packet to randomly generated IP addresses
  - Limited only by bandwidth of the network

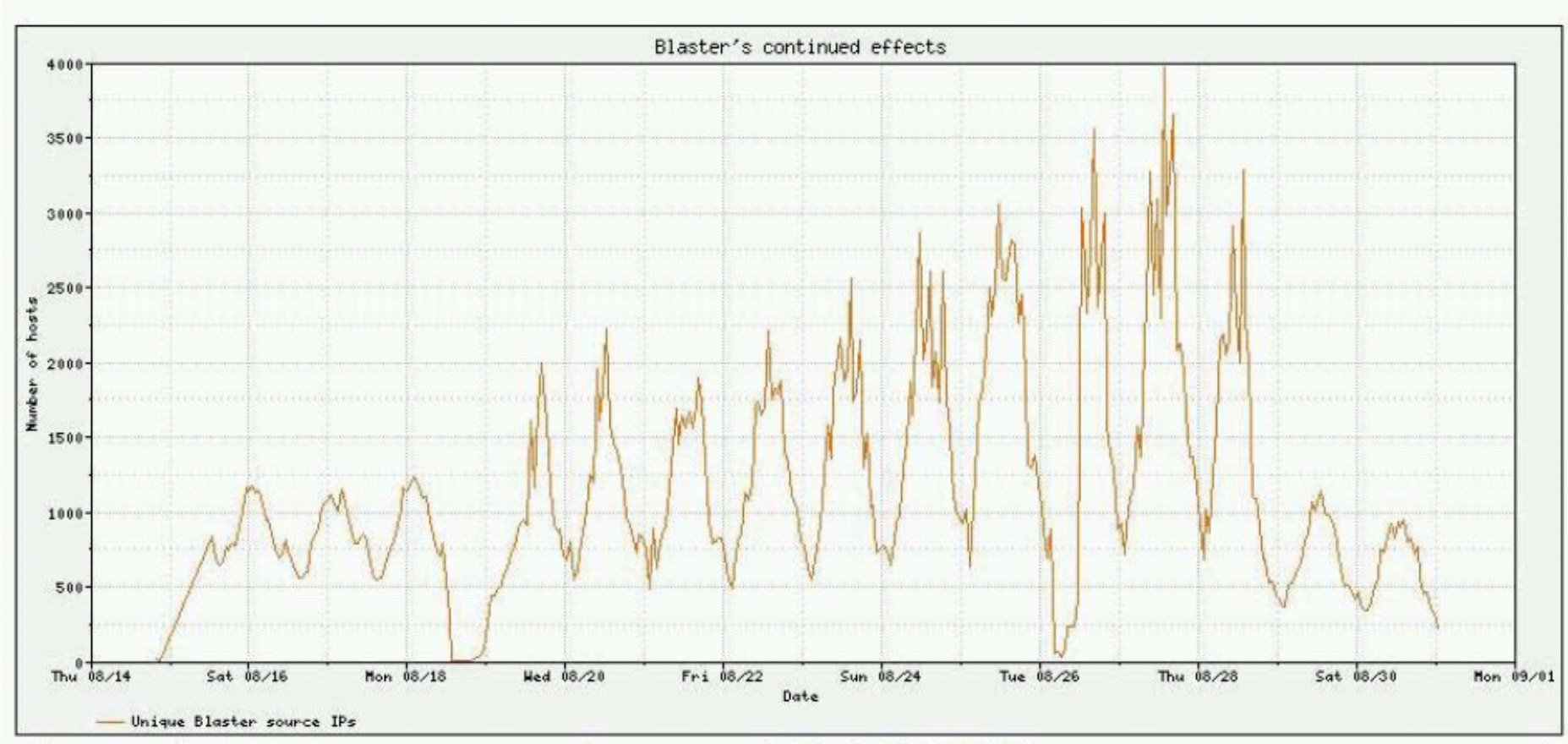
"SLAMMER SHOWED US THAT IT'S HARD FOR EVERYONE TO KEEP UP WITH PATCHES, NO MATTER WHO YOU ARE."

- Mary-Ann Davidson, chief security officer, Oracle

Blaster and Welchia/Nachia

- August 11, 2003: Scanning worm exploiting RPC service in Microsoft Windows XP and 2000
  - First address at random, then sequential upward scan
    - Easy to detect, yet propagated widely and leaped firewalls
- Payload: denial of service against MS Windows Update + installing remotely accessible backdoor
- Welchia/Nachia was intended as a counter-worm
  - Random-start sequential scan, use ICMP to determine if address is live, then copy itself over, patch RPC vulnerability, remove Blaster if found
  - Did **more** damage by flooding networks with traffic

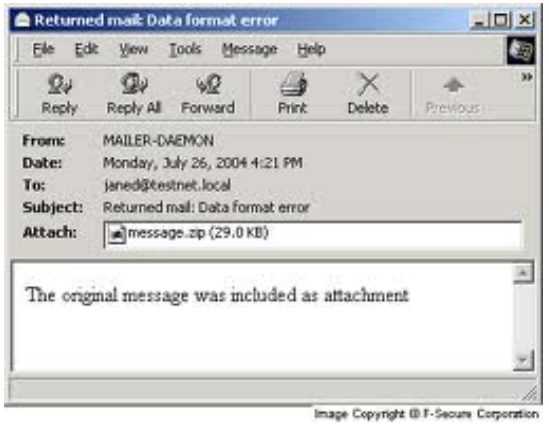
Blaster Worms



Search Worms

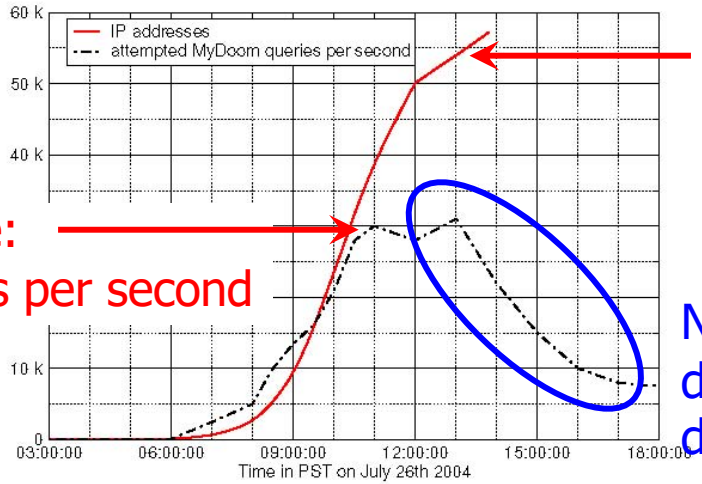
- Generate search query
  - Search for version numbers of vulnerable software to find exploitable targets
  - Search for popular domains to harvest email addresses
- Analyze search results
  - Remove duplicates, URLs belonging to search engine
- Infect identified targets
  - Reformat URLs to include the exploit
    - For example, append exploit code instead of username
  - Exploit code downloads the actual infection, joins the infected machine to a botnet, etc.

MyDoom



- Spreads by email
- MyDoom: searches local hard drive for addresses
- MyDoom.O: uses Web search engines
  - Queries split between Google (45%), Lycos (22.5%), Yahoo (20%) and Altavista (12.5%)

Google's view of MyDoom



Number of IP addresses generating queries (60,000 hosts infected in 8 hours)

Peak scan rate: 30,000 queries per second

Number of served queries drops as Google's anomaly detection kicks in

Santy

- Written in Perl, exploits a bug in phpBB bulletin board system (prior to version 2.0.11)
  - Allows injection of arbitrary code into Web server running phpBB
- Uses Google to find sites running phpBB
- Once injected, downloads actual worm code from a central site, asks Google for more targets and connects infected machine to an IRC botnet
- Multiple variants of the same worm
  - Polymorphism: actual Perl code changes from infection to infection, so filtering worm traffic is difficult!

Evading Anomaly Detection

- Google will refuse worm-generated queries
- Different Santy variants generate different search terms or take them from an IRC botmaster

```
GET /search?q="View+previous+topic+::+View+next+topic"+8756+-modules&num=50&start=35
GET /search?q="vote+in+polls+in+this+forum"+7875+-modules&num=50&start=10
GET /search?q="reply+to+topics+in+this+forum"+5632+-modules&num=50&start=15
GET /search?q="Post+subject"+phpBB+6578+-modules&num=50&start=10
GET /search?q="delete+your+posts+in+this+forum"+9805+-modules&num=50&start=35
GET /search?q="post+new+topics+in+this+forum"+1906+-modules&num=100&start=30
```

- Google's solution: if an IP address generates a lot of "rare" queries, ask it to solve a CAPTCHA
- Do <u>not</u> return the result of a query if it contains (a) pages from many hosts, and (b) high percentage of them are tagged as vulnerable

Worm Detection and Defense

- Worm propagation modeling
- Automatic signature generation
  - E.g., Earlybird, Autograph
- Detection
  - Honeypot-based (e.g., HoneyStat)
  - Local information based (e.g., DSC)
  - Global information based (e.g., Kalman Filter)
- Mitigation and response

CONFICKER WORM/BOTNET

Conficker 2008-2010

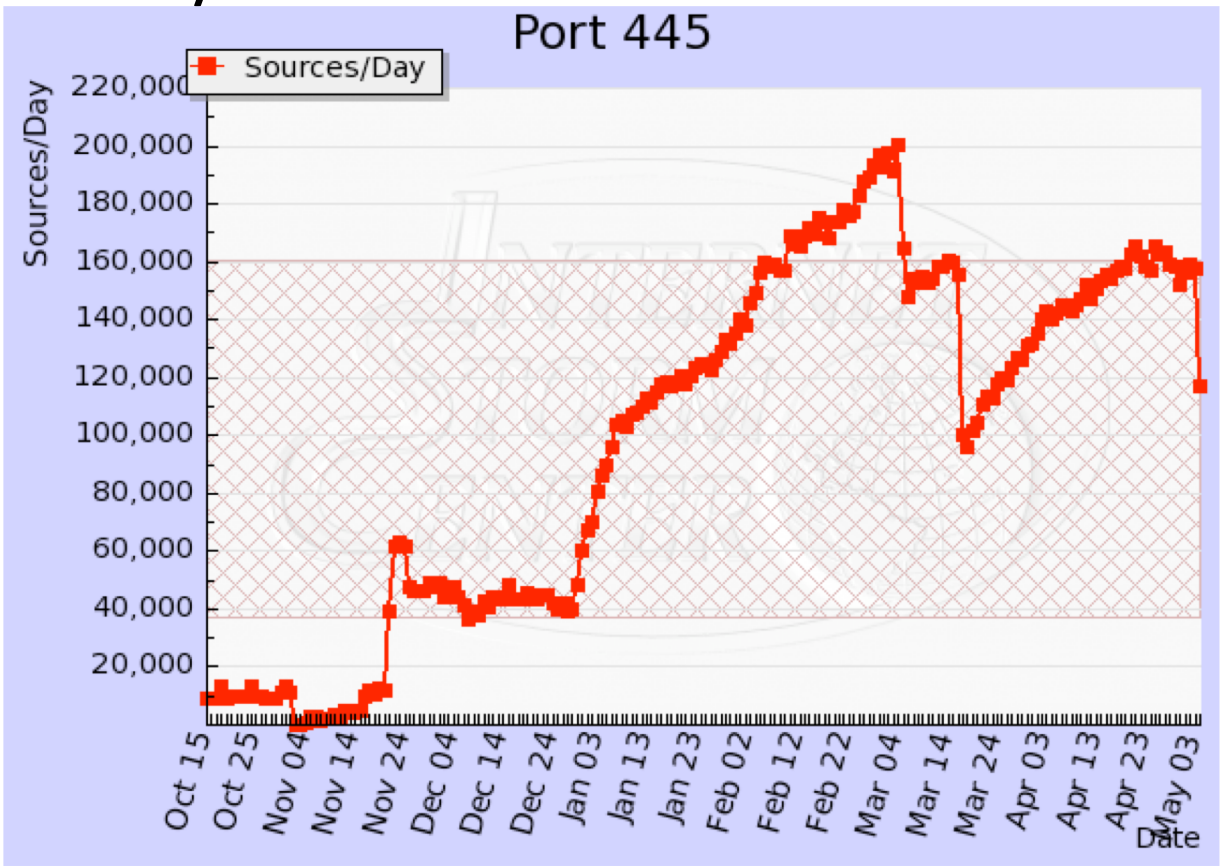
- Most important Worm since Slammer
- Dictionary attacks against Windows admin passwords
- Adds victims to a botnet
- Many different generations, varying attack and defense methodologies

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

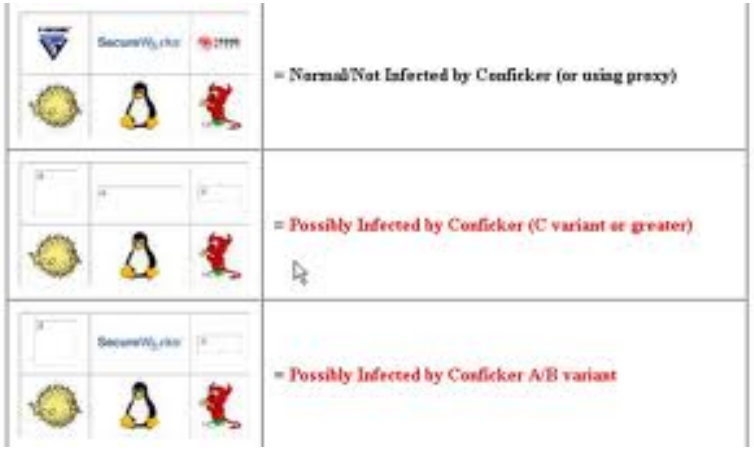


Conficker A 2008-11-21

- **Infection** : Netbios MS08-067
- **Update**: HTTP pull / 250 rand / 8 TLD
- **Self-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
- **Update**
  - HTTP pull / 250 rand / 8 TLD
  - Netbios Push : patch for reinjection
- **Self-defense** :
  - Blocks DNS lookups
  - Disables AutoUpdate
- **End usage** : update to version C or D



Conficker C 2009-03-04

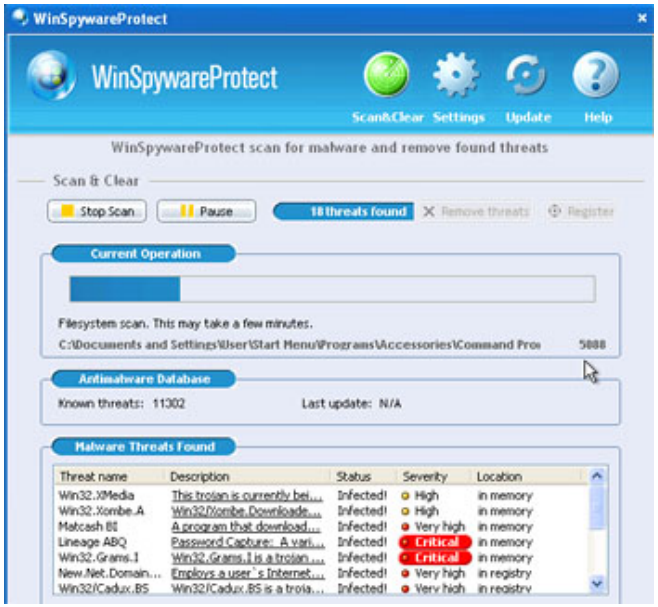
- **Infection** :
  - Netbios MS08-067
  - Removable Media via DLL
  - Dictionary attack on $Admin
- **Update**
  - HTTP pull / 250 rand / 8 TLD
  - Netbios Push : patch for reinjection, Create named pipe
- **Self-defense** :
  - Blocks DNS lookups
  - Disables AutoUpdate

Conficker D 2009-03-04

- **Update**
  - HTTP pull / 50 000 rand / 110 TLD
  - P2P push / pull custom protocol
- **Self-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]**



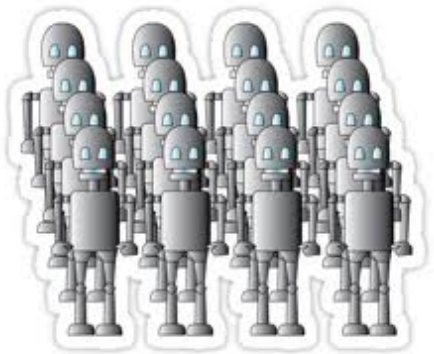
Mirai

- Appeared in 2016
- Targets older versions of Linux, commonly used in Internet-of-Things (IoT) devices
- Created massive botnet — most massive Distributed Denial-of-Service (DDoS) attacks ever seen
- Memory-resident, so rebooting ejects it, but reinfection likely
- IoT devices generally cannot be easily updated or secured

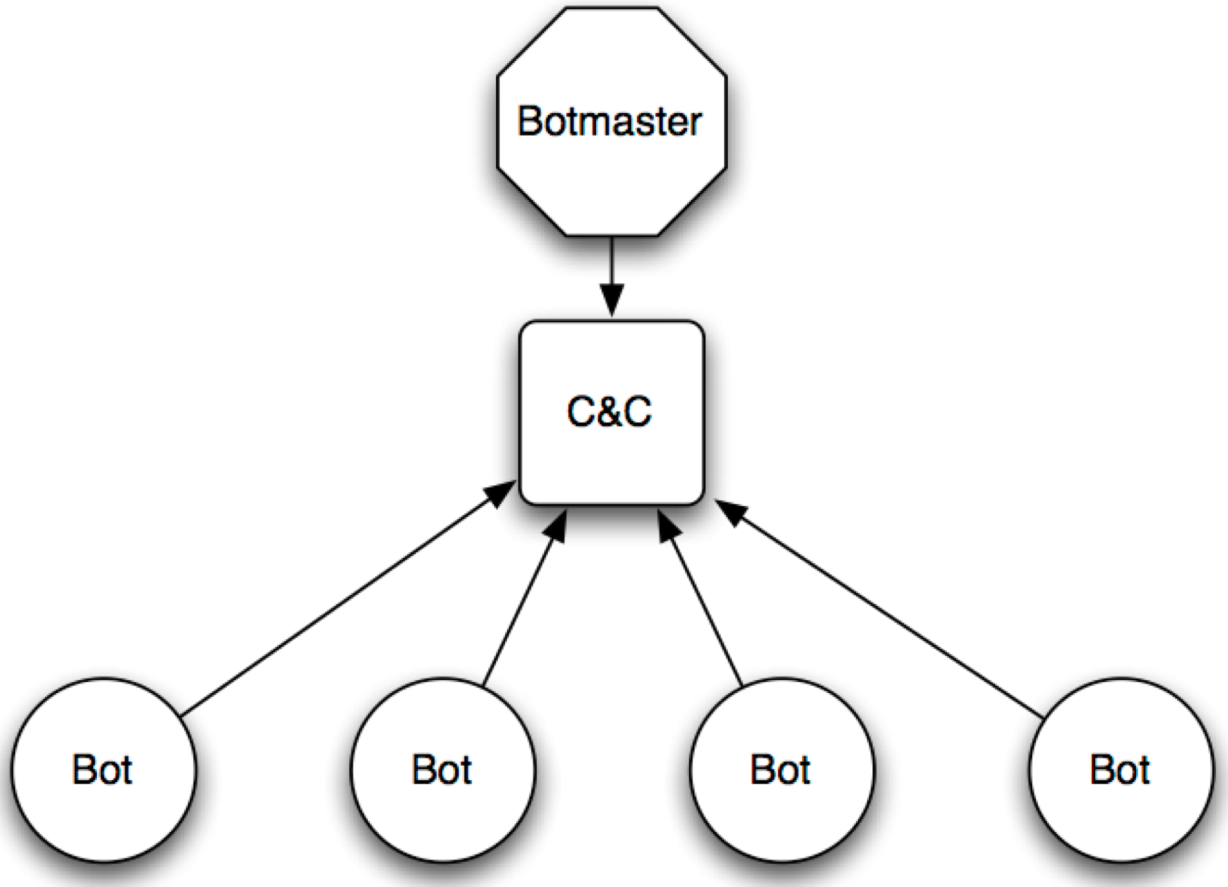
Wannacry Worm

- Appeared in 2017
- Ransomware — encrypts victim's files
- Demands payments in Bitcoin in exchange for the decryption key
- Infected medical, Telecom, and other large-scale businesses
- Contained a killswitch — a domain that did not exist, but was later registered by a security researcher
- There is a simple thing you can do to never have to worry about ransomware: **back up your data!**

BOTNET

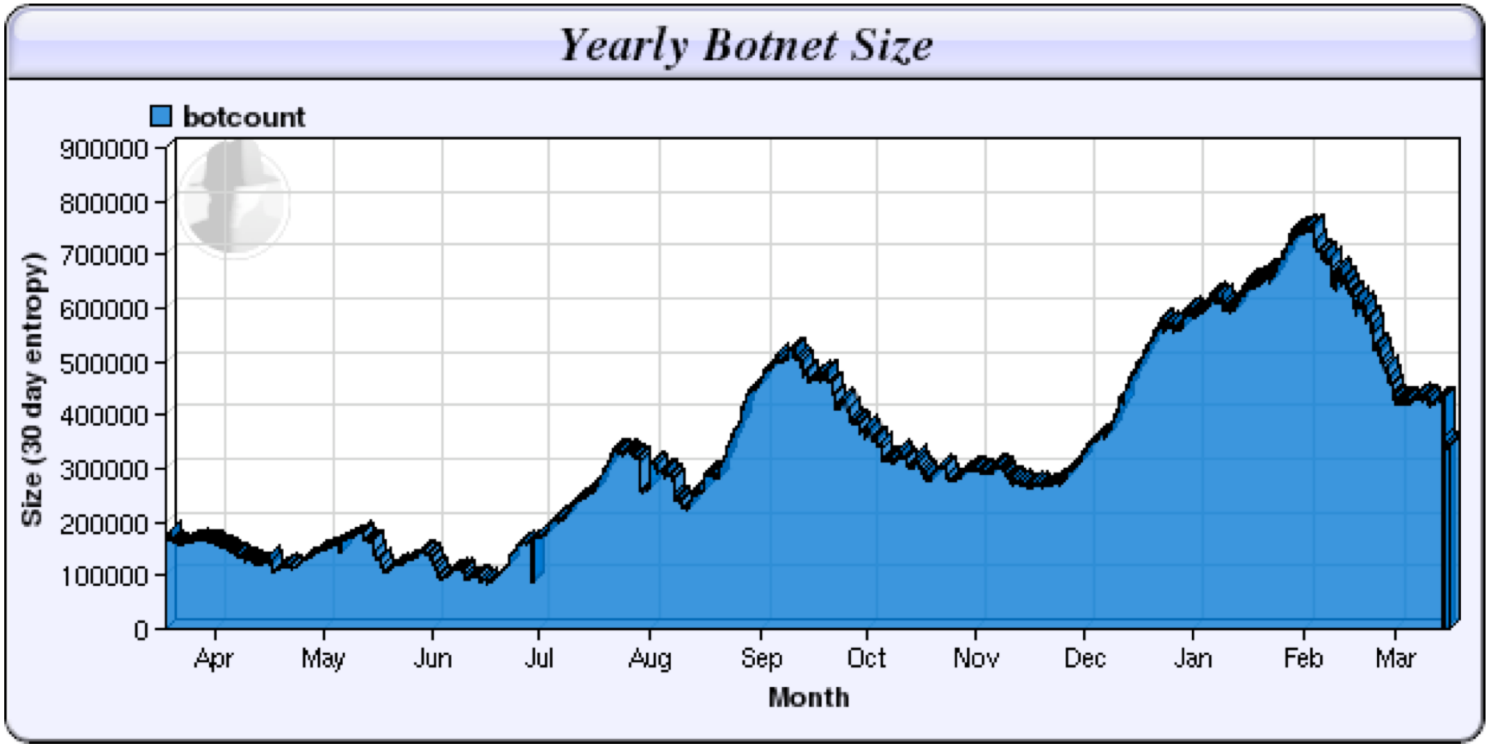


Centralized botnet



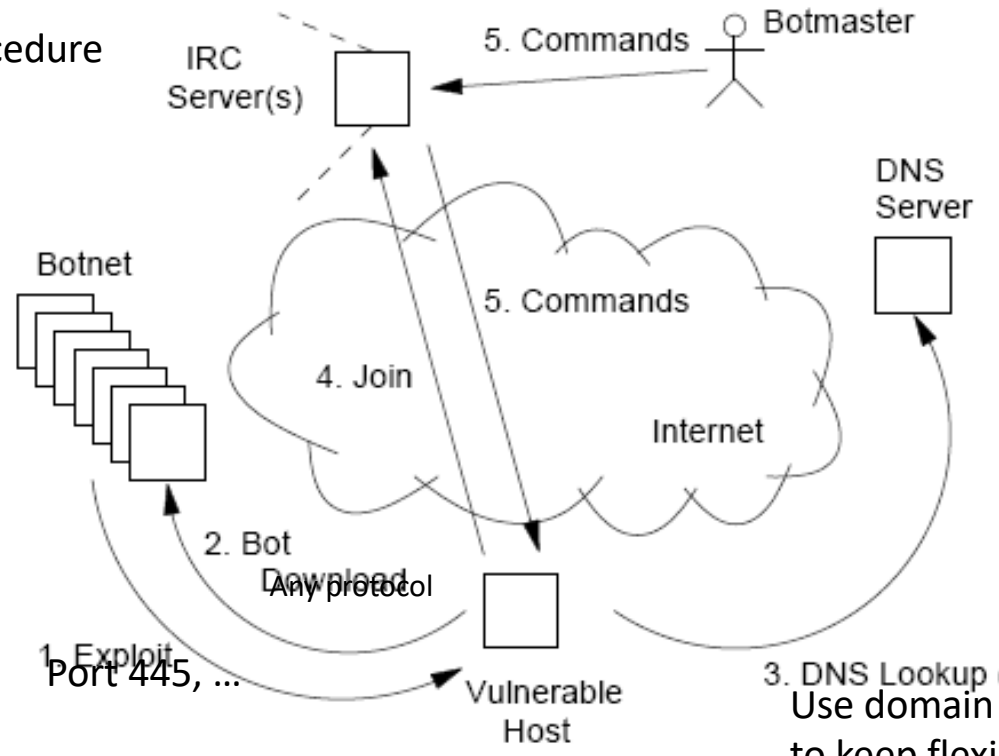
Centralized

C&C centralized Stat



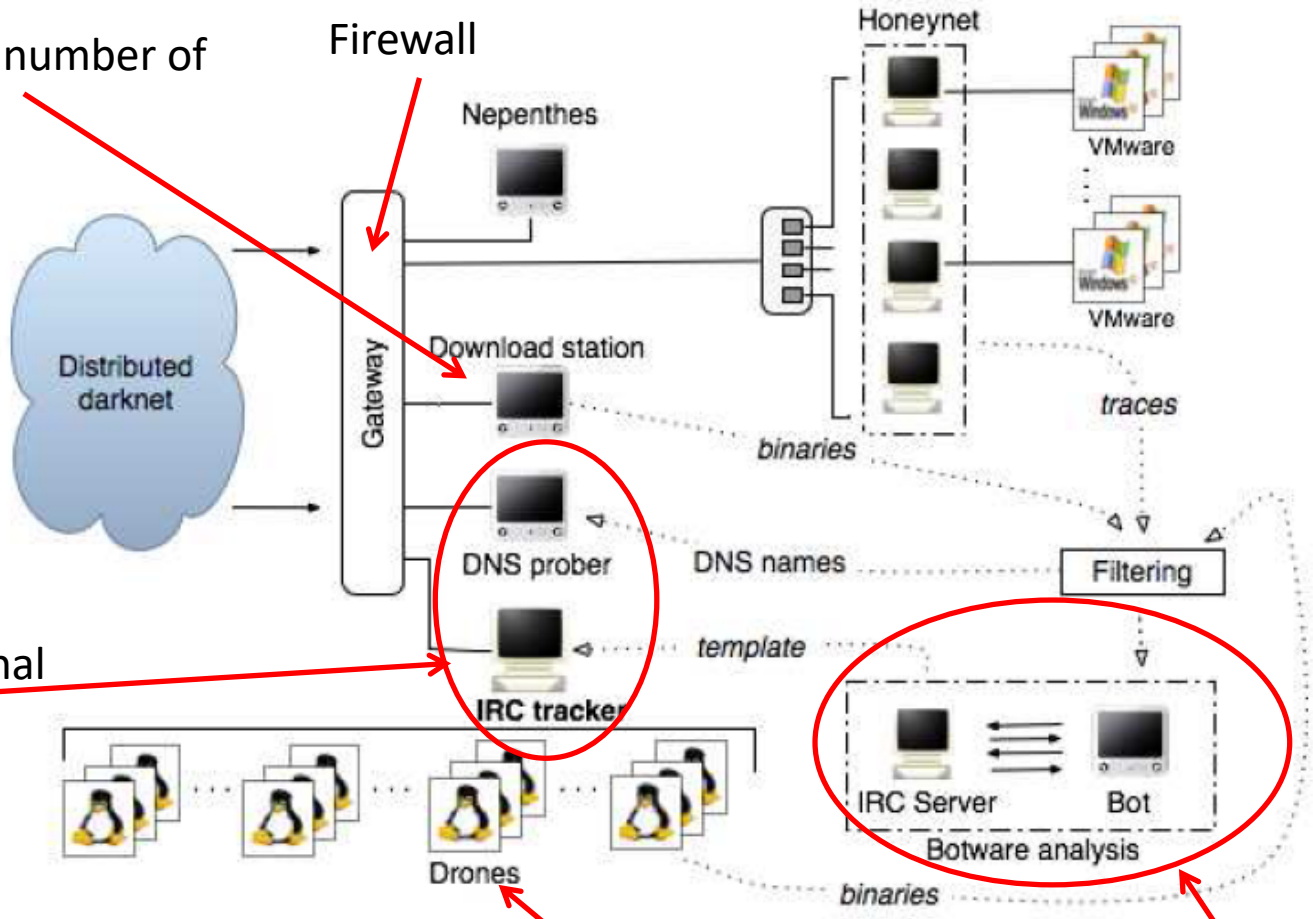
Life-Cycle of an IRC-Based Botnet Infection

4. and 5. include authentication procedure



*A Multifaceted Approach to Understanding the Botnet Phenomenon.

Malware Collection



Reduce the number of downloads

Firewall

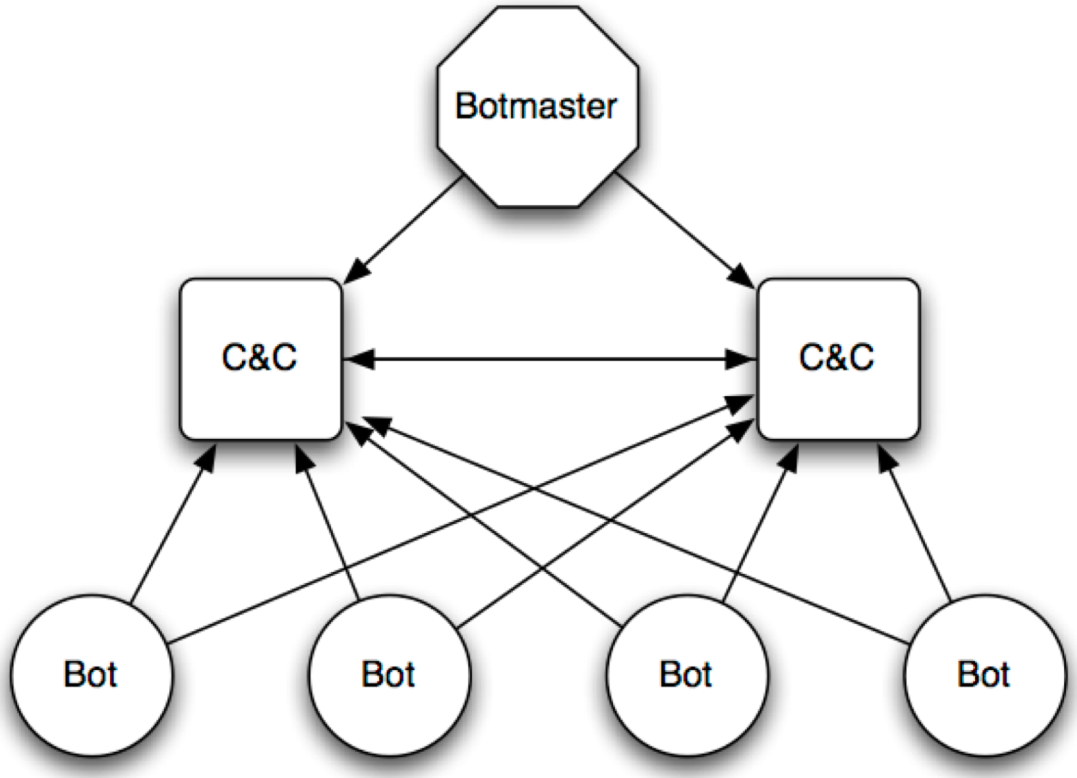
For longitudinal tracking

Its own IRC client

Graybox testing:
Extract network & IRC-related features, and learn the dialect

Data collection architecture.

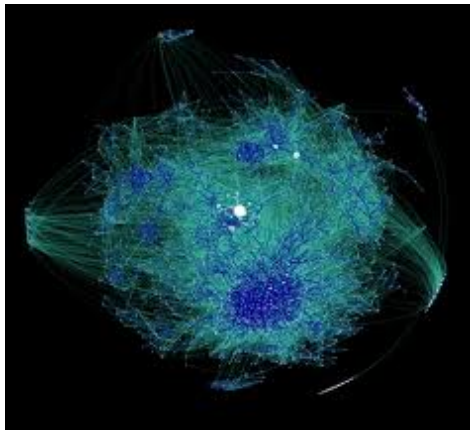
Distributed Botnet



Distributed

Example: Storm

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



Storm Worm / Peacomm (2007)

- Spreads by cleverly designed spam campaign
  - Arrives as an email with catchy subject
    - First instance: “230 dead as storm batters Europe”
    - Other examples: “Condoleeza Rice has kicked German Chancellor”, “Radical Muslim drinking enemies’s blood”, “Saddam Hussein alive!”, “Fidel Castro dead”, etc.
- Attachment or URL with malicious payload
  - FullVideo.exe, MoreHere.exe, ReadMore.exe, etc.
  - Also masquerades as flash postcards
- Once opened, installs Trojan (wincom32) & rootkit

Storm Worm Characteristics

- Infected machine joins botnet
  - Between 1 and 5 million machines infected (Sep 2007)
- Obfuscated peer-to-peer control structure
  - Not like Agobot, which uses simple IRC control channel
  - Interacts with peers via eDonkey protocol
- Obfuscated code, anti-debugging defenses
  - Goes into infinite loop if detects VMware or Virtual PC
  - Large number of spurious probes (evidence of external analysis) triggers distributed DoS attack

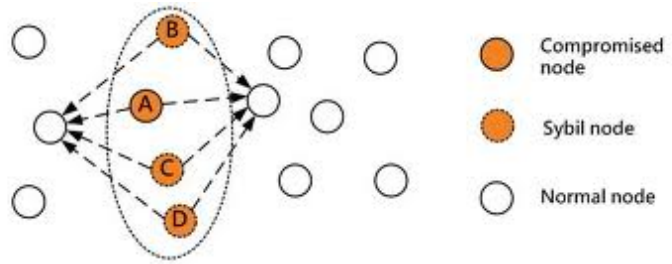
Storm Worm Outbreaks

- Spambot binary used to spread new infections in subsequent campaigns
  - Looks for email addresses and mailing lists in the files on the infected machines

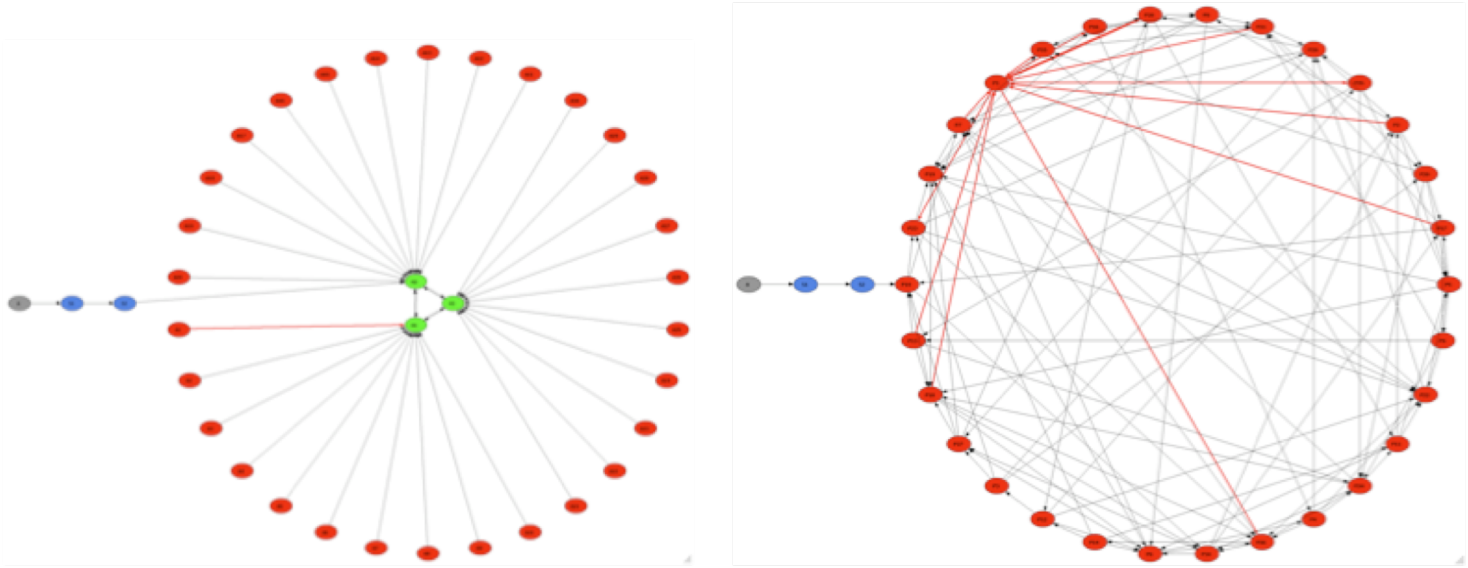
Date	Spam Tactic
Jan 17, 2007	European Storm Spam
April 12, 2007	Worm Alert Spam
June 27, 2007	E-card (applet.exe)
July 4, 2007	231st B-day
Sept 2, 2007	Labor Day (labor.exe)
Sept 5, 2007	Tor Proxy
Sept 10, 2007	NFL Tracker
Sept 17, 2007	Arcade Games

Weakness

- Initial peer list
- sybil attack
- Index poisoning



Network view

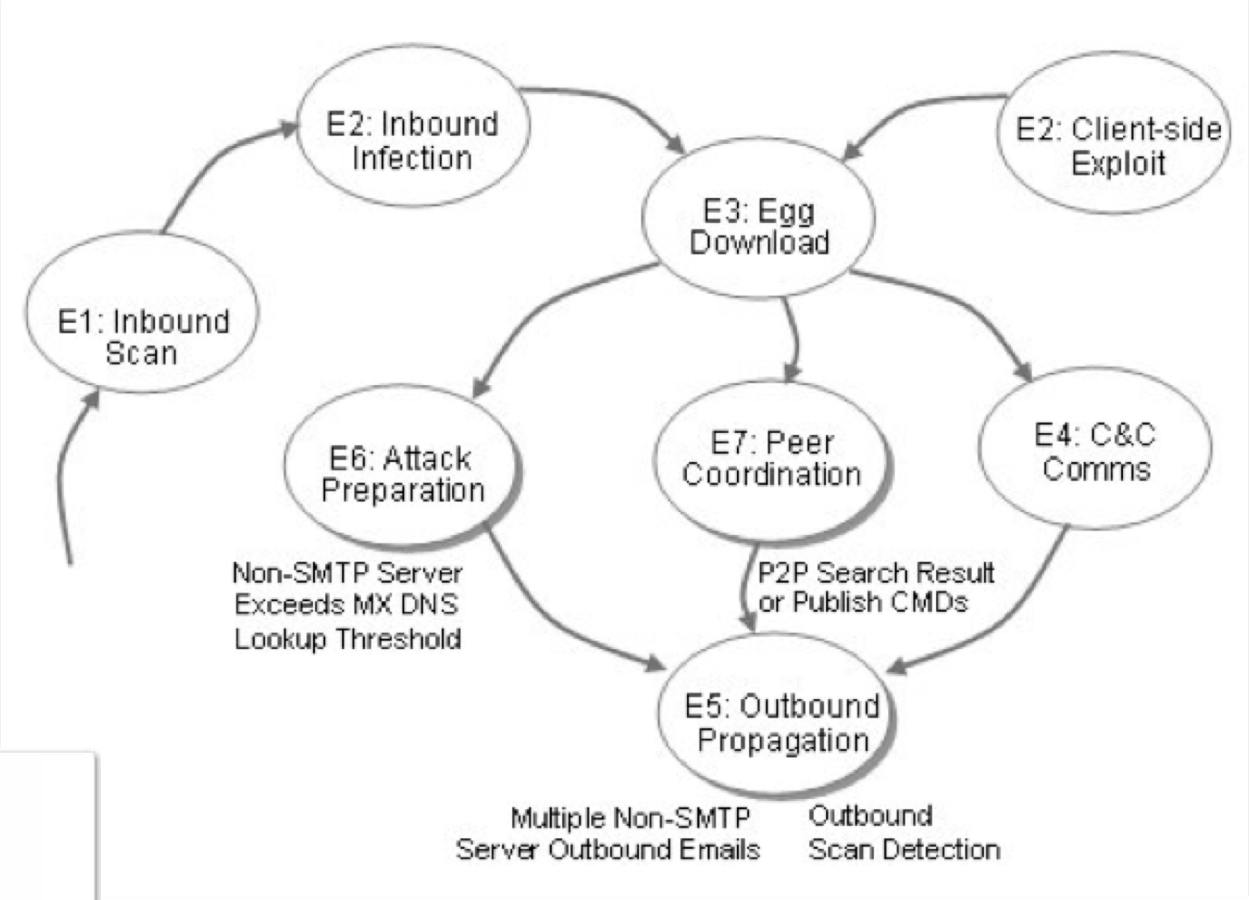


Command and control structures in malware: From Handler/Agent to P2P, by Dave Dittrich and Sven Dietrich, USENIX ;login: vol. 32, no. 6, December 2007, pp. 8-17

Comparison

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

Detecting Botnets Using BotHunter



BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation
Guofei Gu, Phillip Porras, Vinod Yegneswaran, Martin Fong, and Wenke Lee

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

Open Research Questions

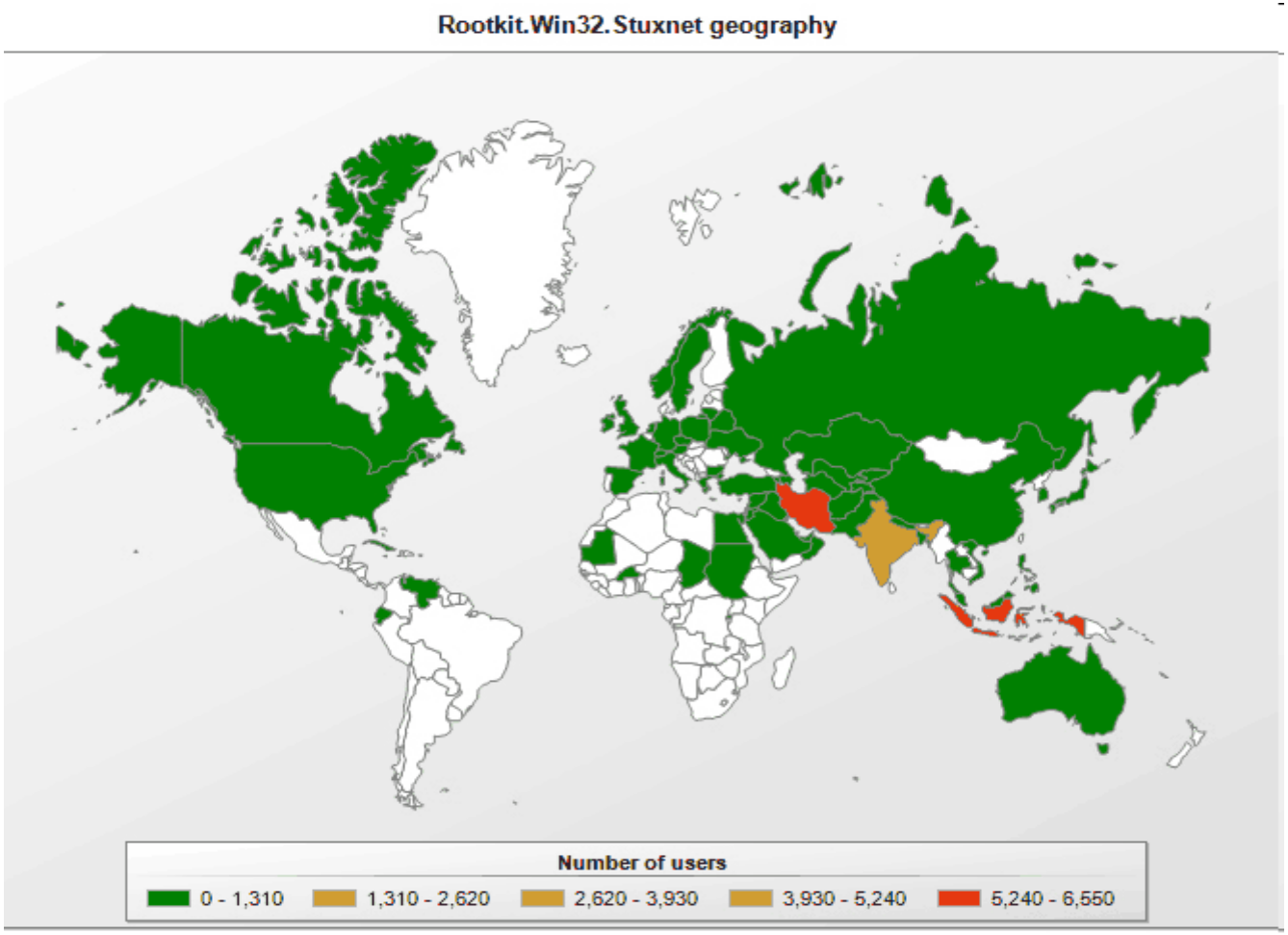
- Botnet detection and defense
  - Under various datasets (packet traces, various numbers of vantage points, etc.)
  - Disable the botnet operation?
- Click fraud detection
- Phishing detection
- …

STUXNET BOTNET: ANOTHER RECENT EXAMPLE

Stuxnet Worm/Botnet Overview

- Primary target: industrial control systems
  - Reprogram Industrial Control Systems (ICS)
  - On Programmable Logic Controllers (PLCs)
    - Specific Siemens Simatic (Step 7) PLC
- Code changes are hidden
- Vast array of components used:
  - Zero-day exploits
  - Windows rootkit
  - PLC rootkit (first ever)
  - Antivirus evasion
  - Peer-to-Peer updates
  - Signed driver with a valid certificate
- Command and control interface

Stuxnet Distribution



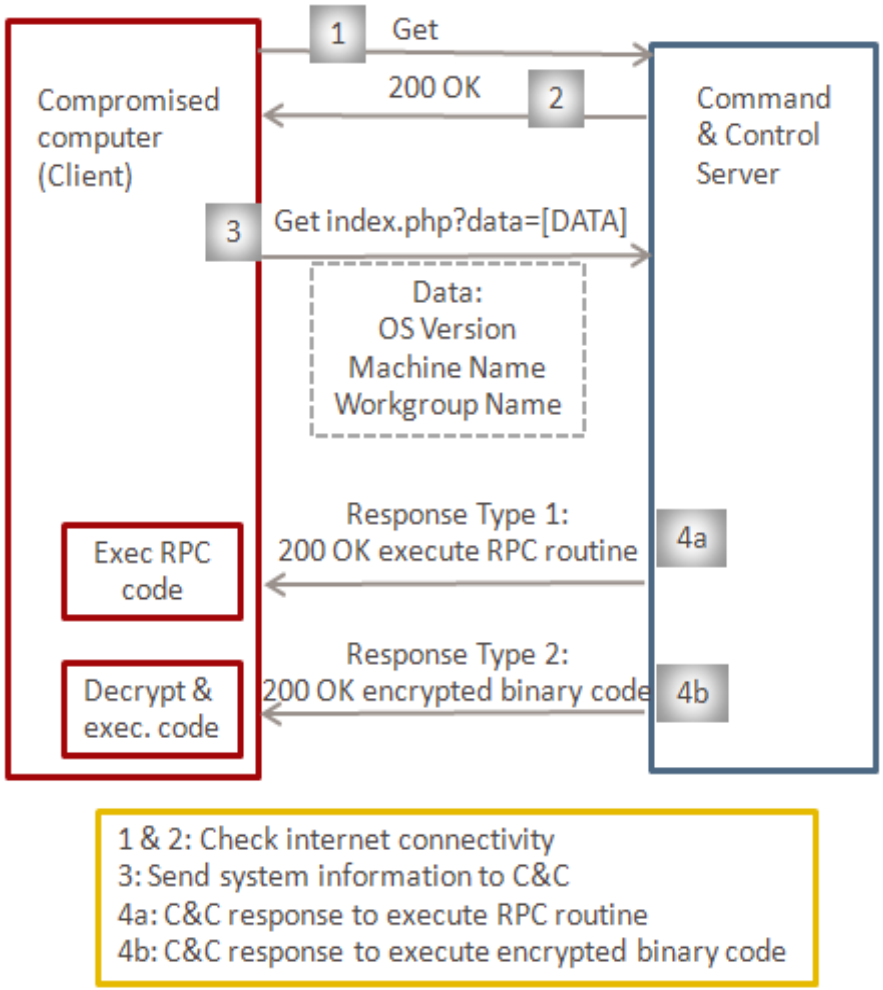
Propagation Methods

- Network
  - Peer-to-peer communication and updates
  - Infecting WinCC machines via a hardcoded database server password
  - Propagating through network shares
  - Propagating through the MS10-061 Print Spooler Zero-Day Vulnerability
  - Propagating through the MS08-067 Windows Server Service Vulnerability
- USB

Command & Control

- Stuxnet contacts the command and control server
  - Test if can connect to:
    - www.windowsupdate.com
    - www.msn.com
  - On port 80
  - Sends some basic information about the compromised computer to the attacker
  - **www.mypremierfutbol.com**
  - **www.todaysfutbol.com**
  - The two URLs above previously pointed to servers in Malaysia and Denmark

Command & Control (2)



Summary

- Stuxnet achieved many things in the malicious code realm
  - First to exploit 4 0-day vulnerabilities
  - Compromised 2 digital certificates
  - Injected code into industrial control systems and hid the code from operators
  - …
- Many experts say it is the most complex malicious software created in the history of cyber security.
- Highlights that it is possible to attack critical infrastructures in places other than Hollywood movies.