CS255: Computer Security

Malware

Chengyu Song 01/05/2022

Lab1: Reverse Engineering

- Goal: understand what the program does and how it works
- Approaches
  - Static: disassembler (objdump, radare2, IDA, Ghidra, Binary Ninja)
  - Dynamic: debugging (gdb, lldb, windbg)
- Why useful?
  - QA: make sure the code is correct
  - Bug fixing: figure out why
  - Malware analysis

Malware

- Malware = **Mal**icious Soft**ware**
  - Virus
  - Worm
  - Botnet
  - Spyware
  - Rootkit
  - Ransomware
  - Crypto miner
  - Keylogger
  - Remote Control
  - etc

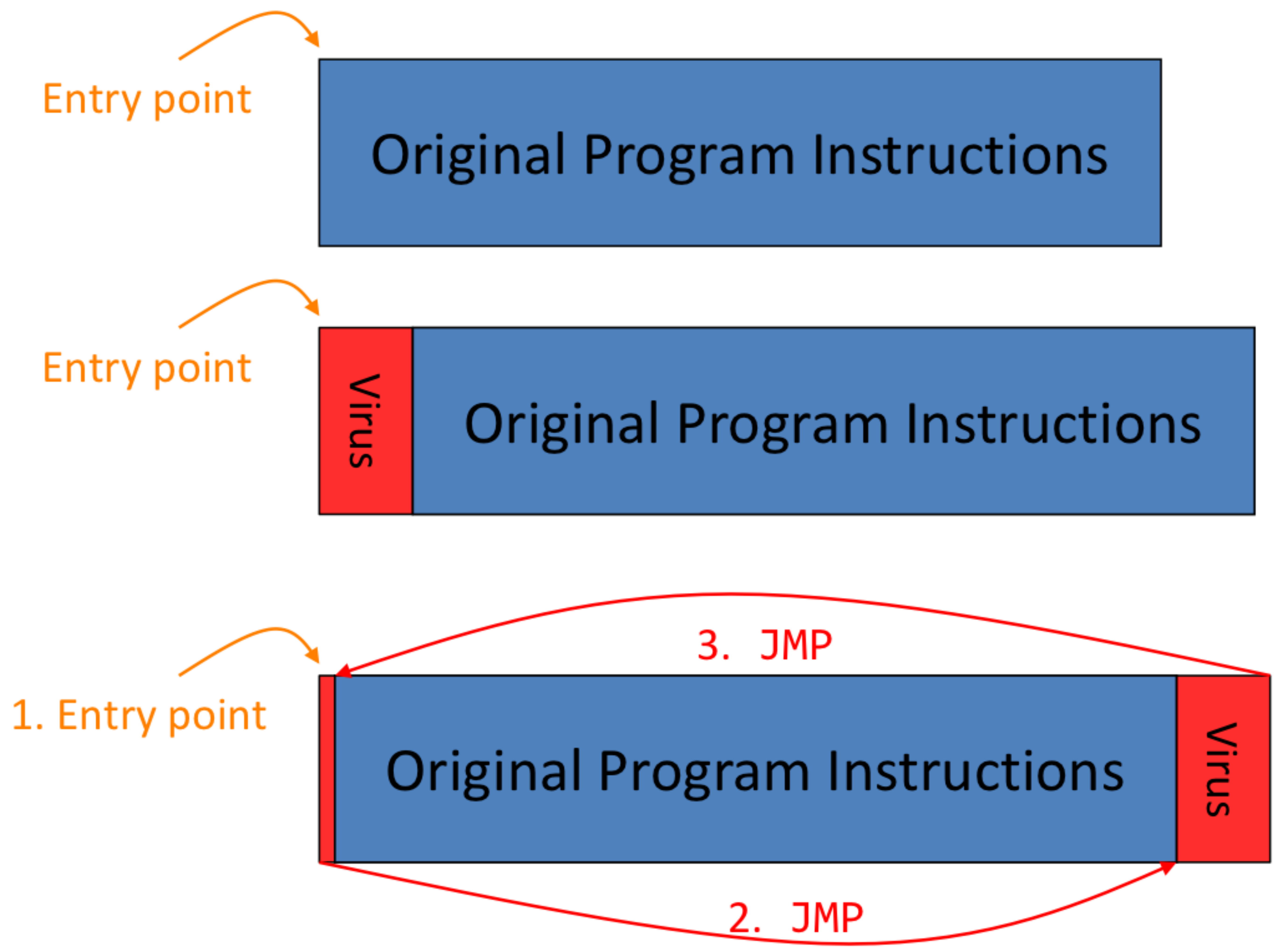
Computer Virus

- Virus = code that replicates
- Originates from a theoretical question
  - **Can a program reproduce itself like organism?**
  - "Theory of self-reproducing automata", *John von Neumann*, 1966
  - Quine: `a=‘a=%r;print(a%%a)';print(a%a)`
- Like real virus, computer virus
  - **Infect** other programs for replication
  - **Hijack** the normal workflow for activation

Propagation of Virus

- General infection strategy: find some code lying around, alter it to include the virus
  - Executables, boot sectors, script (including embedded)
- Example one: attached USB thumb drive
  - Alter executables it holds to include the virus or **autorun** script
  - So once the drive is attached to another machine, boom
- Example two: email attachment
  - Alters attachment to add a copy of itself

Activation of Virus


Entry point
Original Program Instructions
Entry point
Virus
Original Program Instructions
3. JMP
1. Entry point
Original Program Instructions
Virus
2. JMP

Payload

- Besides self-reproducing, what else can the virus do?
    - Pretty much **anything**, payload is decoupled from propagation
        - Only subject to permissions of the infected program
- Examples
    - Brag or exhort (pop up a message)
    - Trash files (just to be nasty) or encrypt them (ransomeware)
    - Damage hardware (e.g., CIH)
    - Keylogging

Computer Worms

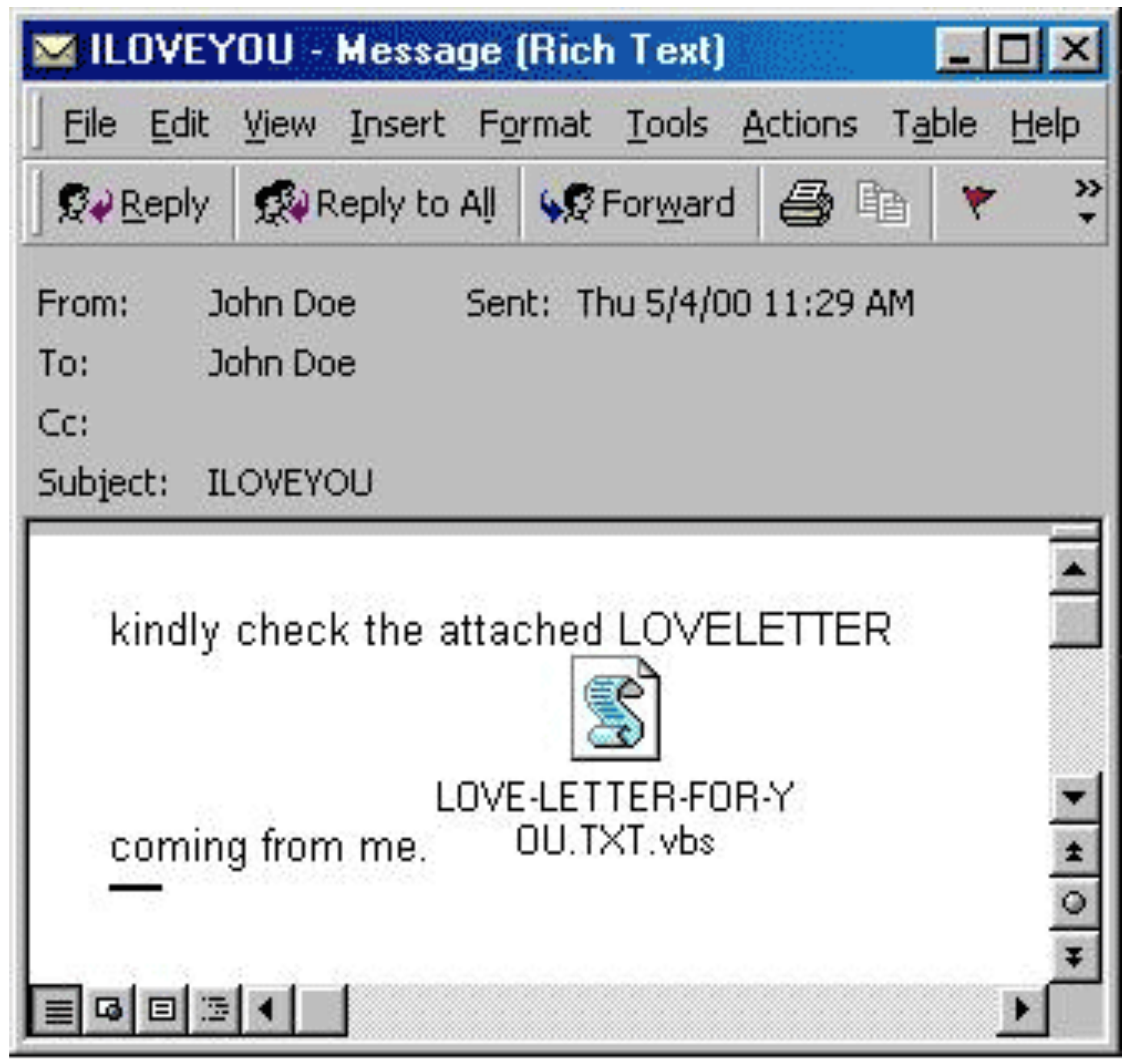
- Worm = malware that **self-propagates**
  - Propagation of virus requires certain type of user interaction
    - Execute program, open file, insert USB disk, etc
  - Worm propagate without user interaction
- How?
  - By exploit **vulnerabilities** of the target system
  - Requires interconnection

Notorious Worms (1)

- Morris (1988): the first worm
  - Scanning the local subnet
  - Exploiting a fingerd buffer overflow
  - Exploiting sendmail's DEBUG mode (not a bug!)
  - Infected approximately 6,000 machine
    - 10% of computers connected to the Internet
  - Cost ~ $10 million in downtime and cleanup

Notorious Worms (2)

- ILOVEYOU (2000): email worm
  - Propagation through email attachment
  - Scans the contacts and sends an email to everyone
  - Estimated to have caused \$5.5–8.7 billion in damages and cost US\$15 billion for removal


ILOVEYOU - Message (Rich Text)
File Edit View Insert Format Tools Actions Table Help
Reply
Reply to All
Forward

From: John Doe Sent: Thu 5/4/00 11:29 AM
To: John Doe
Cc:
Subject: ILOVEYOU

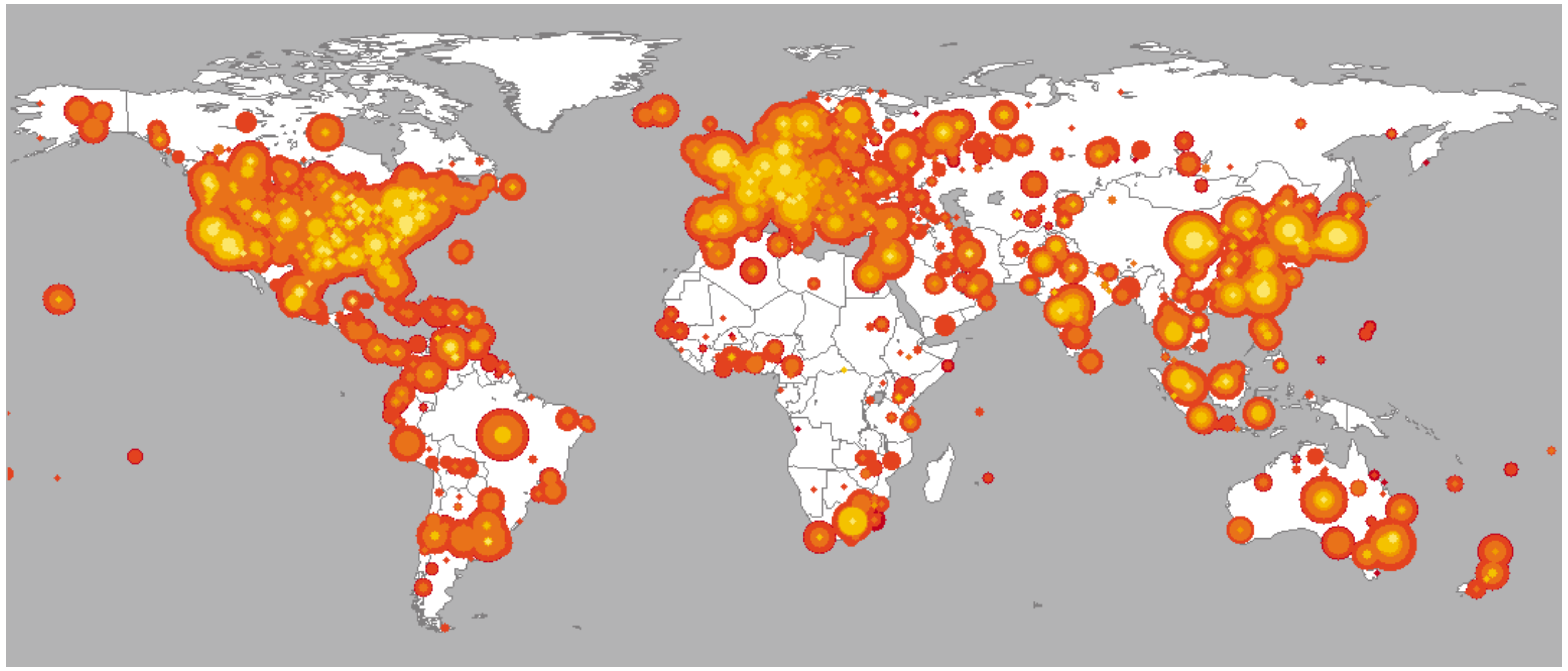
kindly check the attached LOVELETTER coming from me.

LOVE-LETTER-FOR-YOU.TXT.vbs

Un e-mail con el virus ILOVEYOU en todo su esplendor.

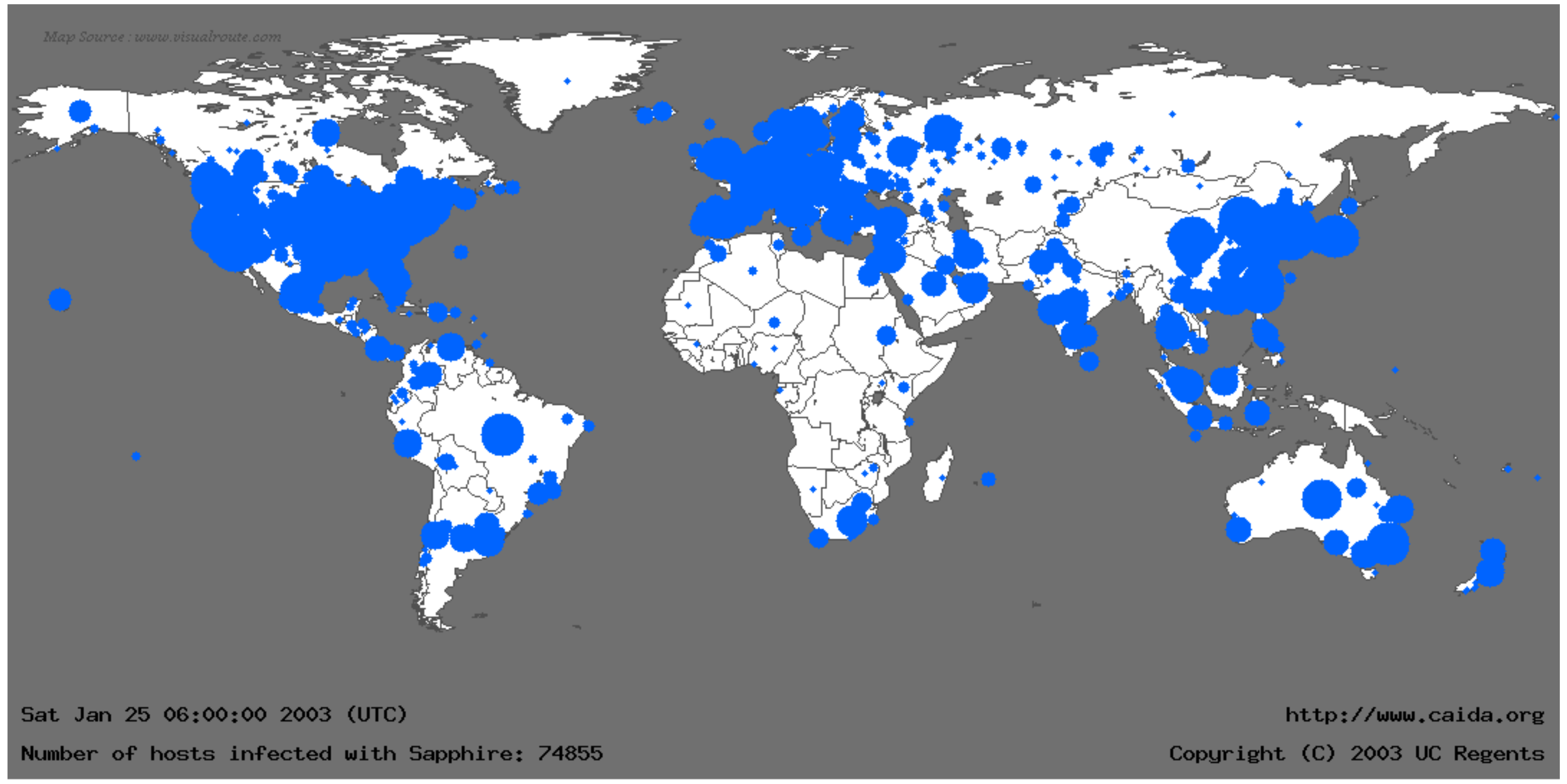
Notorious Worms (3)

- Code Red (2001): fast spreading
  - Exploits buffer overflow vulnerability inside MS IIS
  - Infected more than 359,000 computers in less than 14 hrs



Notorious Worms (4)

- Slammer (2003): fastest ever
  - Exploits buffer overflow vulnerability inside MS SQLServer
  - Infected more than 90 percent of vulnerable hosts within 10 mins

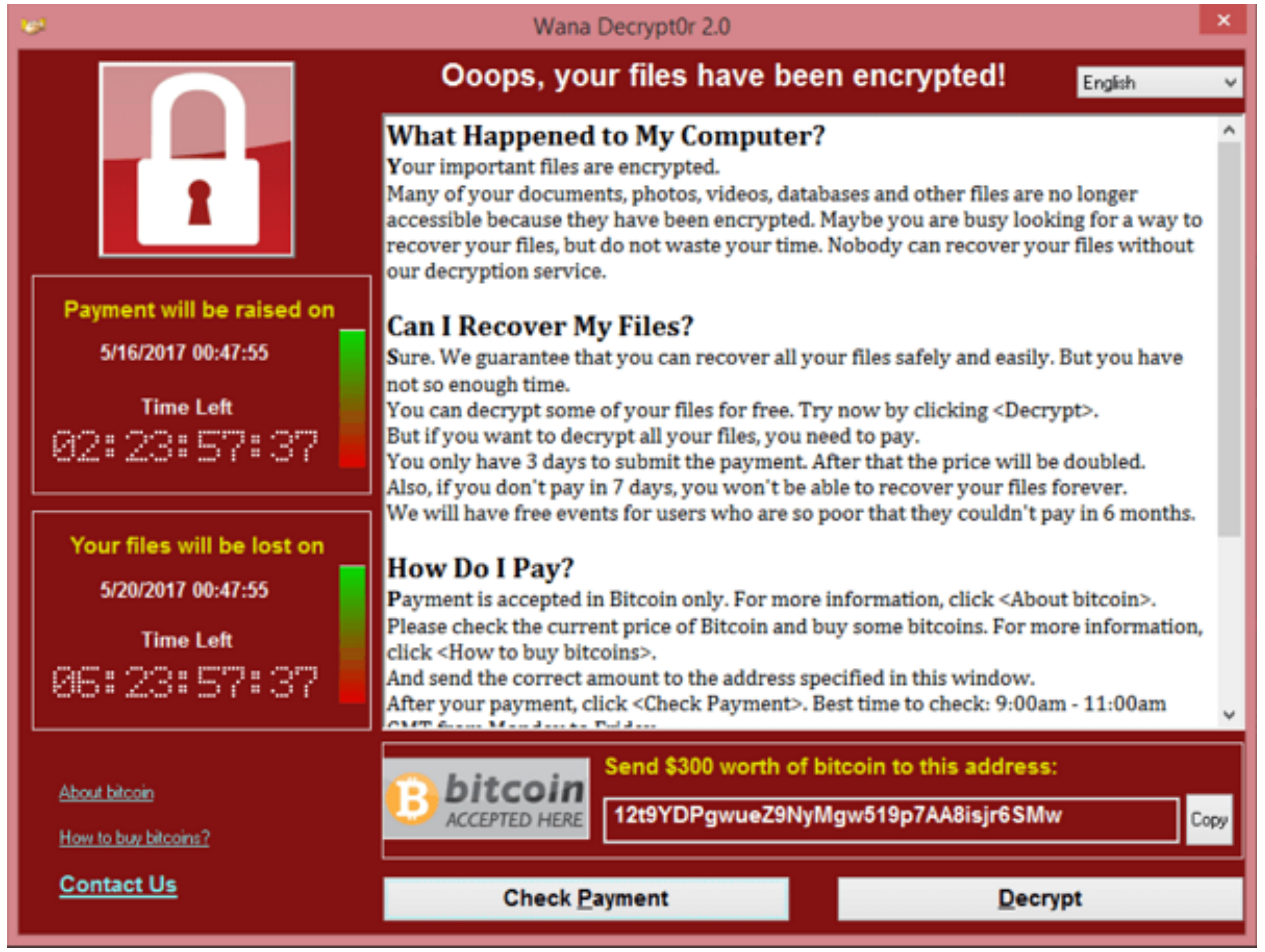

Map Source : www.visualroute.com
Sat Jan 25 06:00:00 2003 (UTC)
Number of hosts infected with Sapphire: 74855
http://www.caida.org
Copyright (C) 2003 UC Regents

Notorious Worms (5)

- Stuxnet (2010): SCADA
  - Multi-mode spreading
    - Initially spreads via USB (virus-like)
    - Once inside a network, quickly spreads internally using Windows RPC
  - Geographically clustered
    - Iran: 59%; Indonesia: 18%; India: 8%

Notorious Worms (6)

- WannaCry (2017): ransomeware
  - Leaked NSA EternalBlue exploit (Windows SMB)



Ooops, your files have been encrypted!

Payment will be raised on

5/16/2017 00:47:55

Time Left

02:23:57:37

Your files will be lost on

5/20/2017 00:47:55

Time Left

06:23:57:37

What Happened to My Computer?

Your important files are encrypted.

Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.

You can decrypt some of your files for free. Try now by clicking <Decrypt>.

But if you want to decrypt all your files, you need to pay.

You only have 3 days to submit the payment. After that the price will be doubled.

Also, if you don't pay in 7 days, you won't be able to recover your files forever.

We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.

Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.

And send the correct amount to the address specified in this window.

After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am

Send $300 worth of bitcoin to this address:

12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw

Notorious Worms (7)

- Mirai Botnet (2016)
    - Infects vulnerable IoT devices (IP cameras and home routers)
        - Common factory default usernames and passwords
    - Used to launch DDoS attacks and mine crypto currency

Botnet

- Botnet = malware that is **remotely controlled** by command and control (C&C) server
  - Collection of compromised hosts (infected in any ways)
  - Platform for many attacks
    - Spam forwarding (70% of all spam)
    - Click fraud / Phishing / Scaware (FakeAV) / Crypto coins
    - Distributed denial-of-service (DDoS)

Spyware

- Spyware = malware that collects your activities
  - Some people don't consider it as real malware (greyware)
    - Google? Facebook?
  - But with advances in machine learning, such activities matters a lot more!

Rootkit/Bootkit

- Rootkit/bootkit = malware that hides other malware
  - Hide the evidence of infection
  - Guarantees persistent
  - Usually executes at very low level (kernel, bootloader, firmware, etc)

Malware Infection

How malware gets into your system?

- Virus: require human interaction
  - **Do not open suspicious files/attachments**
  - **Do not insert unknown USB/Disk**
  - **Do not insert your thumb drive into unknown computer**
- Worm & drive-by: exploit software vulnerabilities
  - **Patch your system as soon as possible**

Malware Infection (cont.)

How malware gets into your system?

- Trojan horse: disguise as something legitimate
  - **Download software from app store or trusted website**
  - **Do not use pirate software**
  - **Check integrity of the software**
- Social engineering: motivate you to do something dangerous
  - **Think twice**

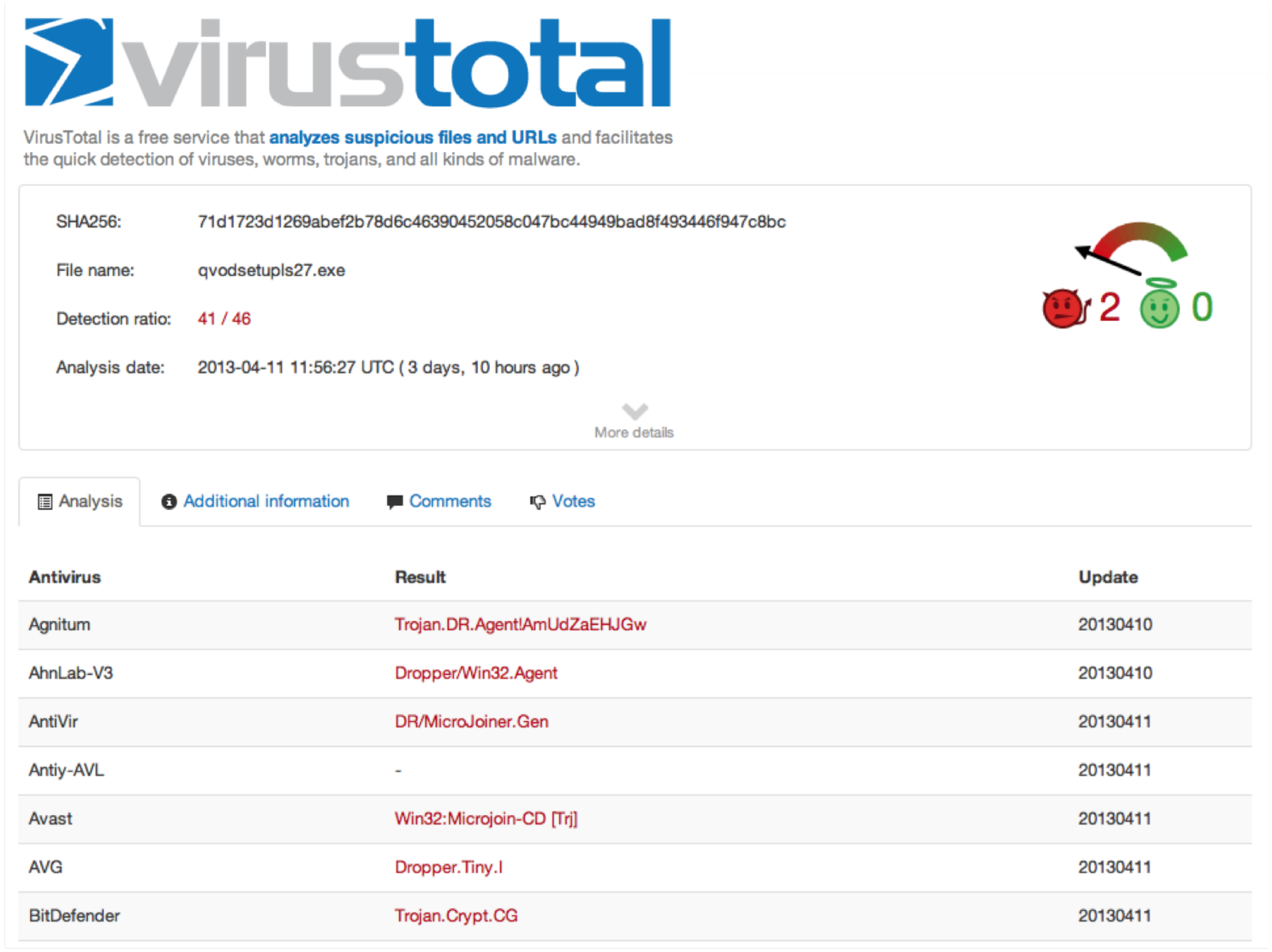
Antivirus

What if you really need/want to open?

- Scan the file for **known** malware => signature-based detection
  - Static signature: look for bytes corresponding to the malware
    - Where to get the samples?
    - How to make sure each signature is unique/good?
  - Why effective? replicating nature of malware
- Drove development of multi-billion $$ AV industry
  - Limited but necessary

Antivirus

An interesting story ...



VirusTotal is a free service that **analyzes suspicious files and URLs** and facilitates the quick detection of viruses, worms, trojans, and all kinds of malware.

SHA256:	71d1723d1269abef2b78d6c46390452058c047bc44949bad8f493446f947c8bc
File name:	qvodsetupls27.exe
Detection ratio:	41 / 46
Analysis date:	2013-04-11 11:56:27 UTC (3 days, 10 hours ago)

Antivirus	Result	Update
Agnitum	Trojan.DR.Agent!AmUdZaEHJGw	20130410
AhnLab-V3	Dropper/Win32.Agent	20130410
AntiVir	DR/MicroJoiner.Gen	20130411
Antiy-AVL	-	20130411
Avast	Win32:Microjoin-CD [Trj]	20130411
AVG	Dropper.Tiny.I	20130411
BitDefender	Trojan.Crypt.CG	20130411

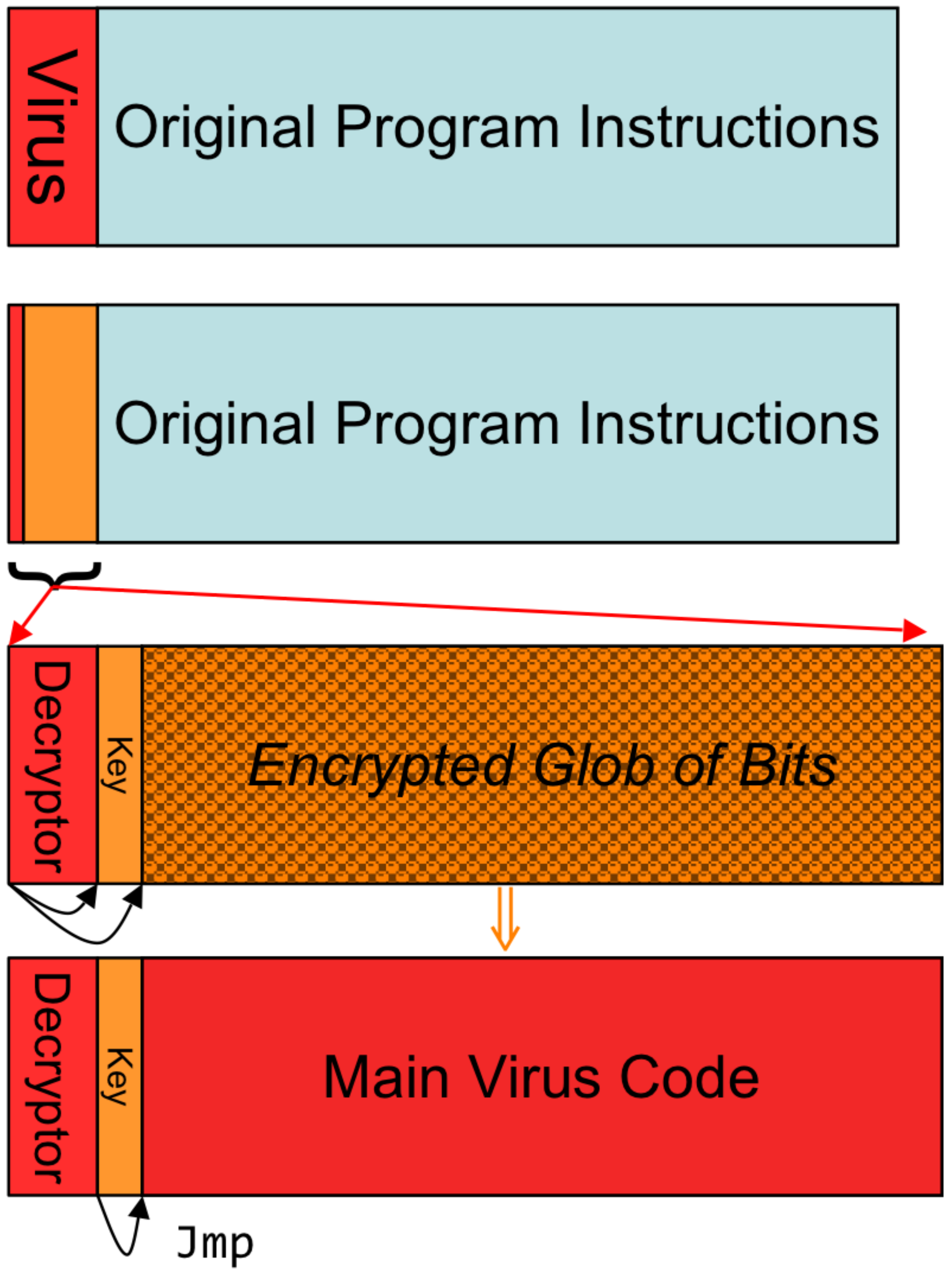
The Arm Race

- If you are a virus writer, what would you do to make sure your effort does not get "wasted" by a signature from the AV industry?
- If you are a AV company, how would you make sure your signature is hard to evade and the database does not explode?

Polymorphic Code

- Idea: change the appearance of the code every time it propagates
- How? **Encryption**!
  - Encodes the message so that the adversary cannot recover its original content without knowing the secret
- Obfuscation (packing)
  - Weak (but simple/fast) crypto algorithm works fine too
  - Strong crypto algorithm: use random key / initial padding

Unpacking



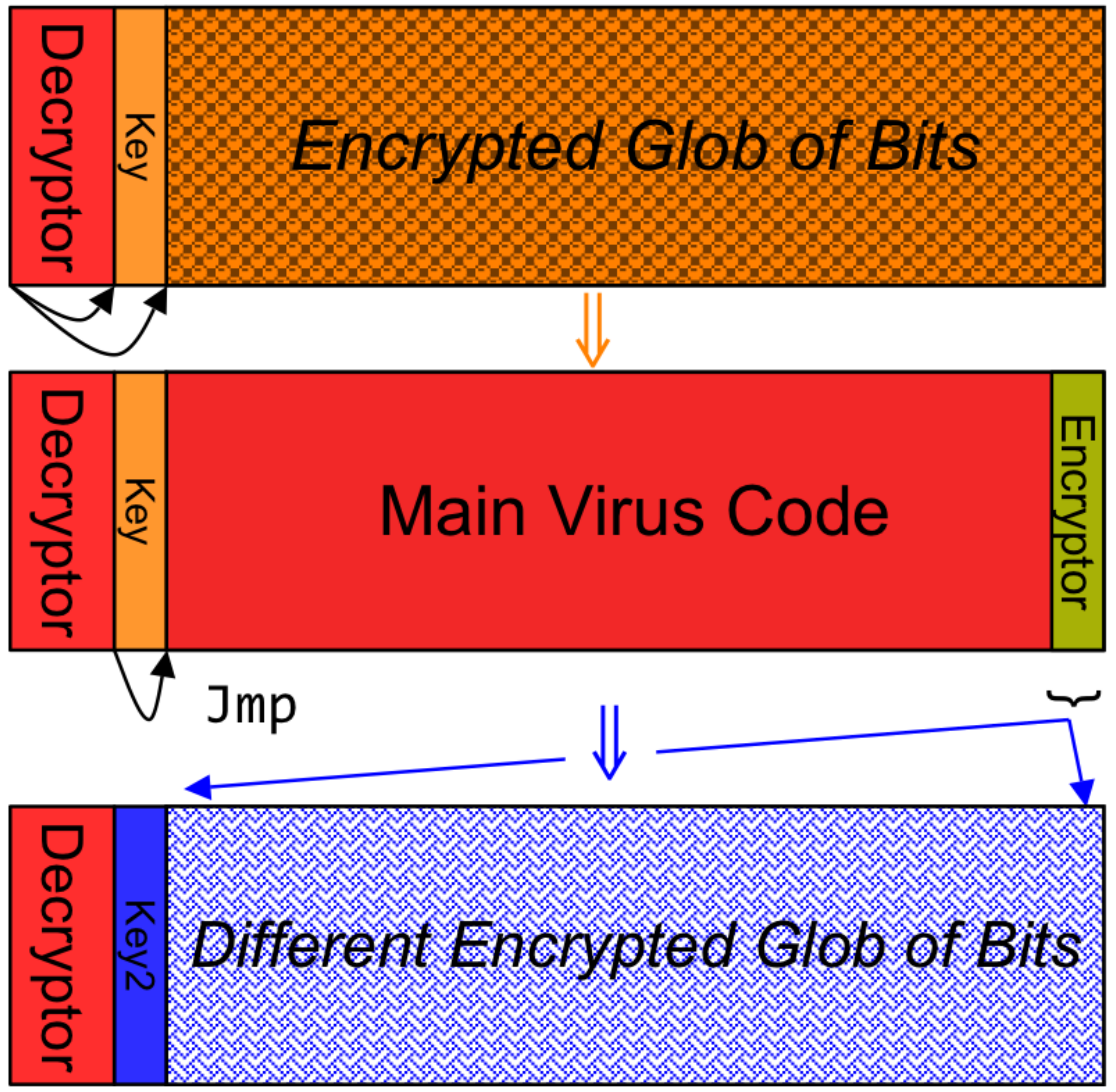
Instead of this …

Virus has *this* initial structure

When executed, decryptor applies key to decrypt the glob …

… and jumps to the decrypted code once stored in memory

Polymorphic Propagation



Once running, virus uses an *encryptor* with a new key to propagate

New virus instance bears little resemblance to original

Arm Race

Detecting polymorphic malware

- How would you detect a polymorphic malware?
- Idea #1: detect the unpacker/decryptor
    - False positives: less code to match, legitimate software also use obfuscation to protect IP
- Idea #2: decrypt and detect
    - Speculative runs the software for a while and scan memory
        - But for how long?
- Virus-writer countermeasures?

Metamorphic Code

- Idea: change the **syntax** of the code every time it propagates
- How? **Code rewriter**
    - Renumber registers
    - Change order of conditional code
    - Reorder operations not dependent on one another
    - Replace one low-level algorithm with another
    - Junk dead code
    - etc

Metamorphic Code

Hunting for Metamorphic, Szor & Ferrie, Symantec Corp., Virus Bulletin Conference, 2001

Arm Race

Detecting metamorphic malware

- How would you detect a metamorphic malware?
- Idea: focus on **semantics** (behaviors) instead of appearance
  - Create signatures for **malicious behaviors** (e.g., syscall-based)
  - Monitor dynamic behaviors of a process and detect malicious ones
- Virus-writer countermeasures?
  - Anti dynamic analysis
    - VM/emulator/debugger detection, triggers, env binding, etc
  - Metamorphic syscalls

Summary

Host side detection

- Deciding whether a software is malicious or not in general, is not decidable
    - With theoretical proof (**the halting problem**)
- In practice, signature/black-list based approach has one big limitation
    - **Only detects known malware**
    - VT as an oracle
- What about white list approach, like on iOS
    - Much better but still limited