SOPHOS

Viruses and virus detection

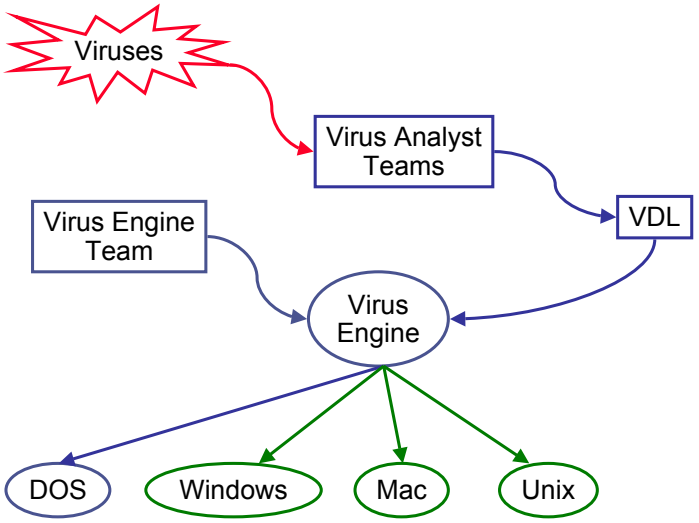
Paul Jarvis

Virus Engine Team Leader

30 October 2002

The Virus lab teams

SOPHOS



Viruses and virus detection

SOPHOS

- Introduction to viruses
- The changing threat
- Virus writers
- Virus types
- Executable file infection
- Virus hiding techniques
- Finding viruses

SOPHOS

Viruses and virus detection

Introduction to viruses

Malicious software attacks

SOPHOS

- Trojans
- Viruses
- Worms

Trojans

SOPHOS

- Executable code
- Do not self-replicate
- Unexpected result
- Usually malicious payload

Viruses

SOPHOS

- Self replication
- Require a host
- Executable path
- Side effects
- Disguise

Worms

SOPHOS

- “Self-contained” viruses
- Usually use the Internet

SOPHOS

Viruses and virus detection

The changing threat

www.sophos.com

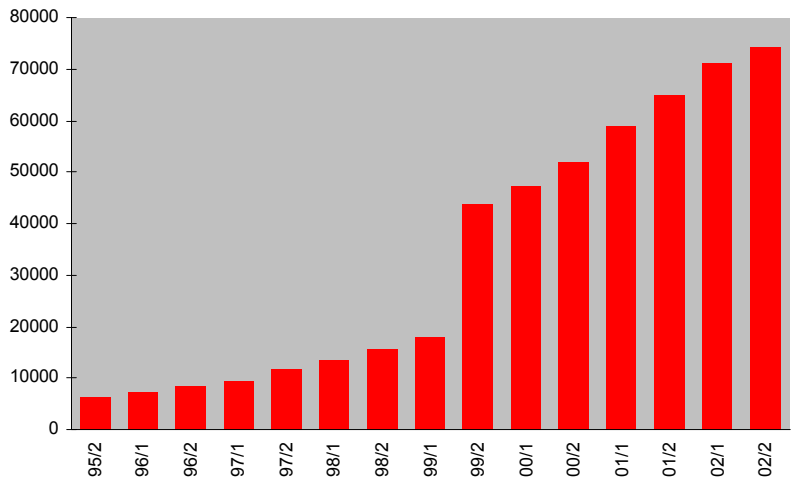
Viruses - a short history

SOPHOS

- 1982 - First self-replicating code (Xerox)
- 1984 - First paper on viruses (Fred Cohen)
- 1986 - First large-scale virus infection (Brain)
- 1987 - First anti-virus software
- 1992 - First virus construction kits
- 1998 - First Access, PowerPoint and Java viruses, CIH
- 1999 - First viruses mass spread by Email, Melissa
- 2001 - First ITW Unix viruses

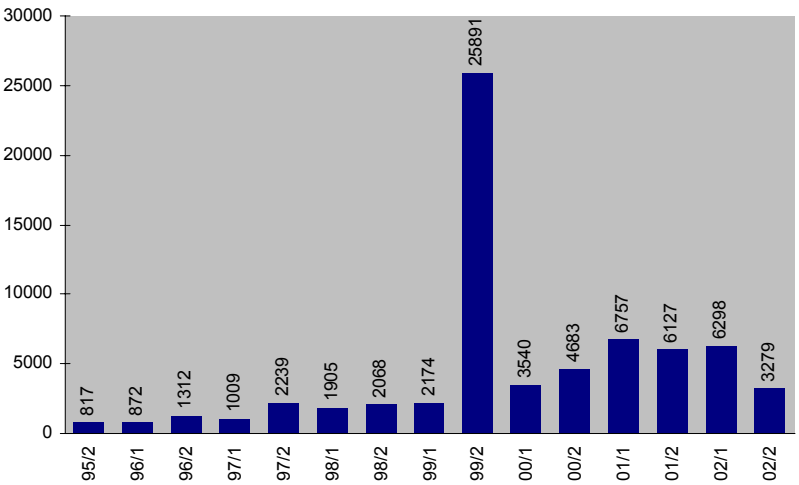
Virus numbers

SOPHOS



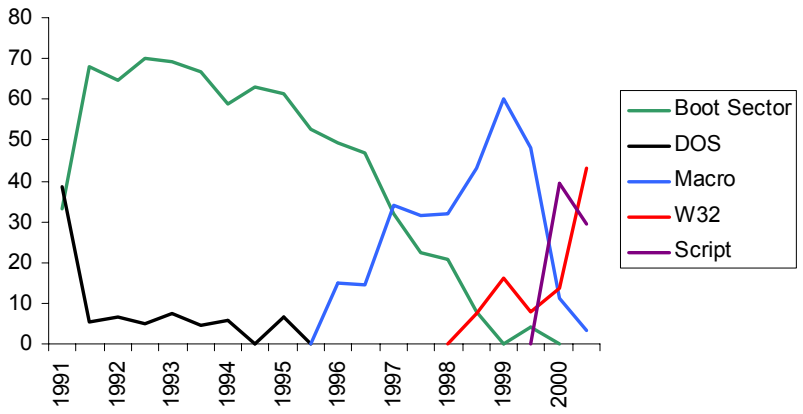
New viruses

SOPHOS



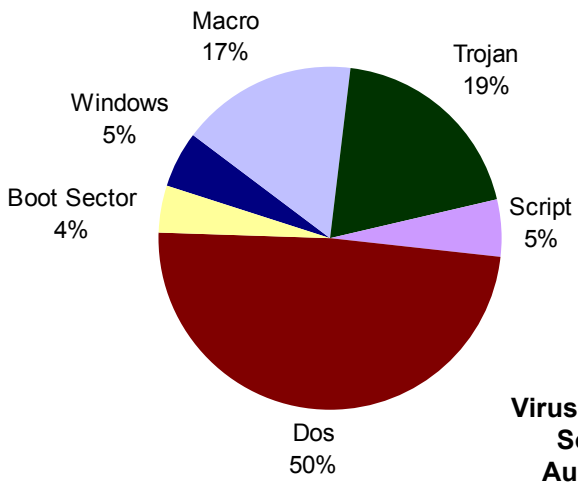
Overview of the Top 10 Reports

SOPHOS



Viruses known to Sophos

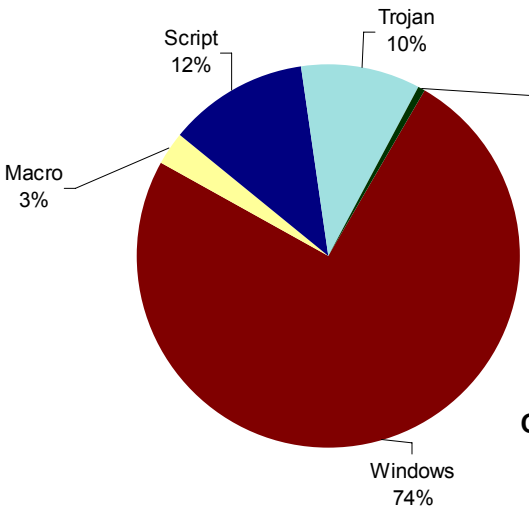
SOPHOS



Viruses known to Sophos in August 2002

Viruses reported to Sophos

SOPHOS



Customer reports to Sophos August 2002

SOPHOS

Viruses and virus detection

Virus writers

Who writes viruses?

SOPHOS

No “standard” virus writer, no “standard” motivation

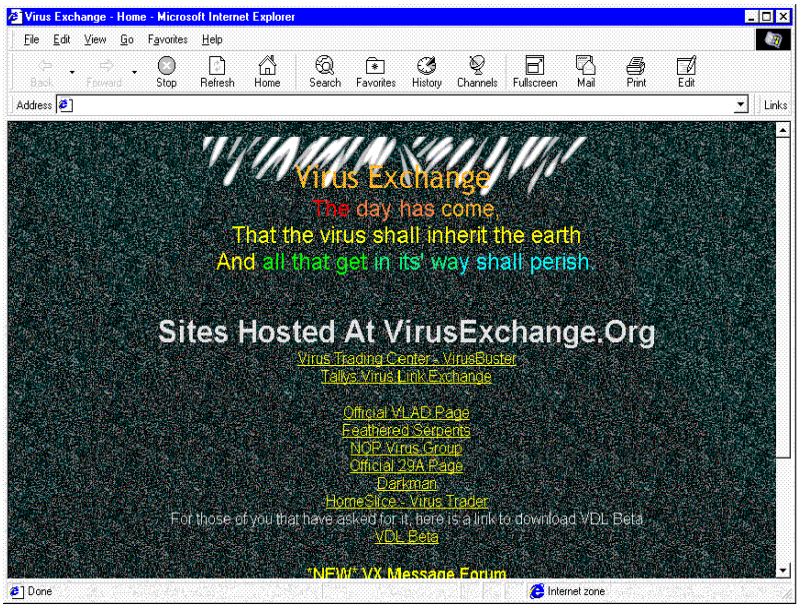
- School children
- Undergraduates
- Post-graduates
- IT Professionals

Virus writers...

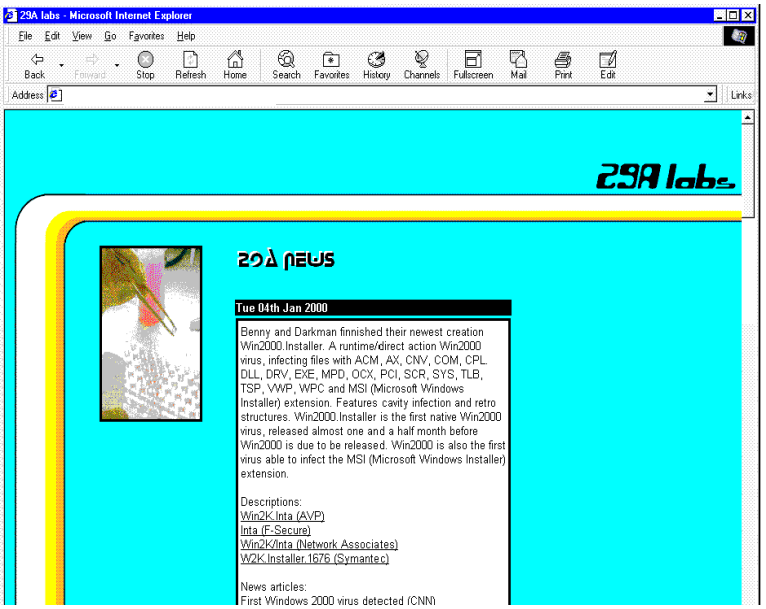
SOPHOS

- Are organised
- Publish books and magazines
- Exchange information
- Operate web sites
- Are (generally) male

SOPHOS



SOPHOS



Virus construction tools

SOPHOS

- VCS (?, 1990)
- VCL (Nowhere Man [Nuke], 1992)
- PS-MPC (Phalcon/Skism, 1992)
- G2 (Phalcon/Skism, 1993)
- CVEX (Golden Cicada, 1995)
- Various WM toolkits (1997+)

Polymorphic toolkits

SOPHOS

- Mutation Engine (Dark Avenger, 1992)
- TPE (Masud Kafir, 1993)
- DSME (Taiwan, 1993)
- SMEG (England, 1995)
- WinLamer (Taiwan, 1995)
- RDA (?, 1995?)

SOPHOS

Viruses and virus detection

Virus types

www.sophos.com

Virus types

SOPHOS

- Hoaxes
- Boot sector viruses
- Macro viruses
- Script viruses
- Executable viruses

Hoaxes

SOPHOS

- Always from a trusted source e.g. Microsoft, IBM
- Will do terrible things, e.g. delete all files without even opening the email
- Usually includes a request to forward on to everyone you know
- Examples:
  - Good Times
  - Irina
  - Penpal greetings
  - JOIN THE CREW

Boot sector viruses

SOPHOS

- Independent of installed operating system
- Cannot handle modern filing systems
- Generally spread by exchanging floppy disks
- Very effective spreading initially
- Currently no significant threat
- Can cause the operating system not to initialise

Macro viruses

SOPHOS

- Fast spread
- Mass infections
- Easy to write
- Easy to modify
- Cross-platform

Script viruses

SOPHOS

- Easy to write
- Easy to modify
- Existing tools
- No programming knowledge required

Executable viruses

SOPHOS

- Available for DOS, Win9x, NT/2000, Unix, Mac
- Are specific to operating system
- More complex to write
- Can use the operating system facilities (e.g. filestore access)

SOPHOS

Viruses and virus detection

Executable file infection

Overview

SOPHOS

- Only covering executable file infections
  (Other viral types are generally much more simple)
- Executable viruses require more knowledge to write and are thus usually more complex
- These often include code to hinder detection
- Recent additions include code to prevent disinfection

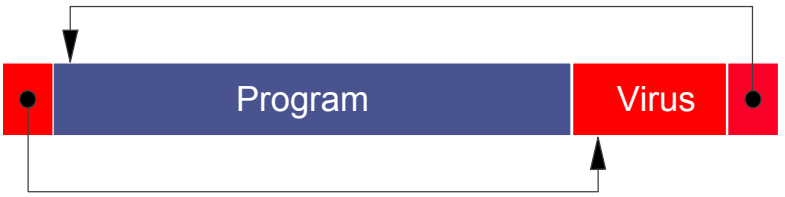
Com files

SOPHOS

- The most basic executable format on DOS
- Are small - maximum 64K
- Contain just the raw program
- Are copied into memory and run
- Very simple to infect

Infecting com files

SOPHOS



Com files

SOPHOS

- Increase in file size
- Care required to not multiply infect a file
- Require ‘stealth’ to hide
- Not straight forward to disinfect

Exe (16 bit files)

SOPHOS

- Have a defined structure
- Can be large
- Can use overlays
- Simple to infect

MZ File format

SOPHOS

MZ header
Image size
Entry point
Relocation table(s)
Code

```
4D 5A 9A 00 29 00 00  00-20 00 C5 00 FF FF A7 05  MZ..)... ....
80 00  00 00 10 00 CF 04-1E 00 00 00 01  00 00 00  ................
00 00  00 00 00 00 00  00-00 00 00 00 00  00 00 00  ................
00 00  00 00 00 00 00  00-00 00 00 00 00  00 00 00  ................
00 00  00 00 00 00 00  00-00 00 00 00 00  00 00 00  ................
00 00  00 00 00 00 00  00-00 00 00 00 00  00 00 00  ................
00 00  00 00 00 00 00  00-00 00 00 00 00  00 00 00  ................
00 00  00 00 00 00 00  00-00 00 00 00 00  00 00 00  ................
```

Exe (16 bit files)

SOPHOS

- Similar infection to com files
- Header requires correct updating
- Stealth to hide size change
- Slightly more difficult to disinfect

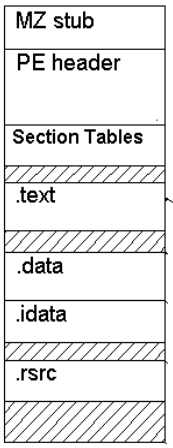
Exe (32 bit files)

SOPHOS

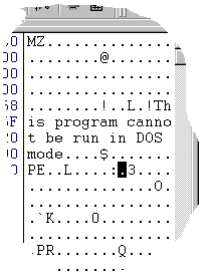
- Have a defined structure
- More complicated than 16 bit exe
- Can be very large
- Start with a 16 bit ‘stub’
- Slightly harder to infect

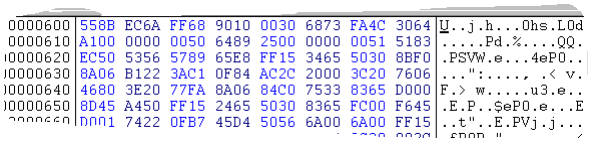
PE File format

SOPHOS



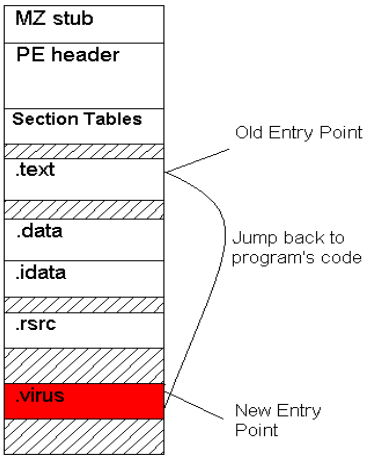
Entry Point





Simple Win32 virus

SOPHOS



Exe (32 bit files)

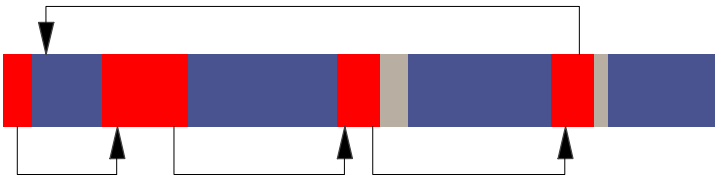
SOPHOS

- Header requires correct updating
- Stealth to hide size change, but ...
- File structure leaves gaps
- Cavity infection leaves no size change

Cavity infection

SOPHOS

Uninfected program (grey = unused)



SOPHOS

Viruses and virus detection

Virus hiding techniques

www.sophos.com

Hiding techniques

SOPHOS

- Originally hiding required to:
  - Ensure further infection before detection
  - To remain undetected until the payload activates - trigger date.
- Currently:
  - Mass emailers don't bother to hide
  - Back doors require to remain undetected

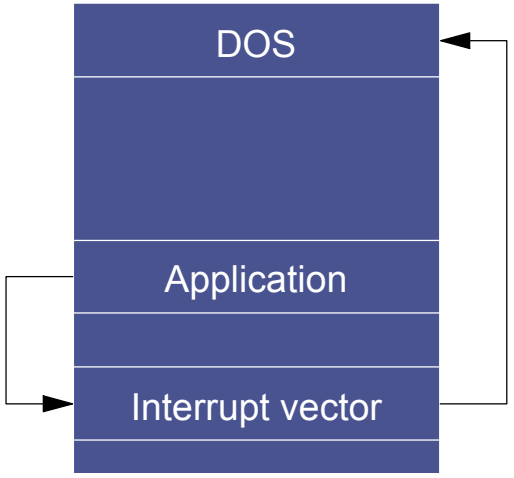
Hiding techniques

SOPHOS

- If active in memory then a virus can:
  - infect other files
  - ‘hack’ requests that would reveal it
- The virus can hide in files using:
  - Encryption
  - Polymorphism
  - Metamorphism

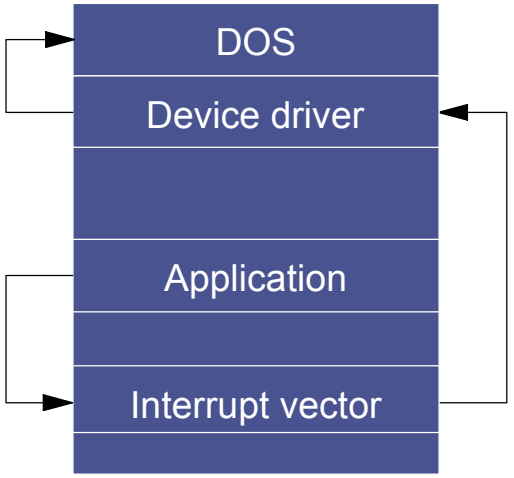
Normal system

SOPHOS



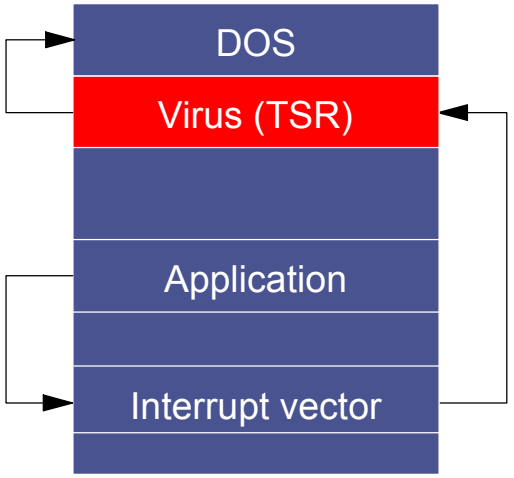
Device driver installed

SOPHOS



Virus installed

SOPHOS



Encrypted virus

SOPHOS

Program 1	DEYu*&81|p[@#
Program 2	DE132{+as$5\%6
Program 3	DE!"334%'dfs6456

Polymorphic virus

SOPHOS

Program 1	DEYu*&81|p[@#
Program 2	4k132{+as$5\%6
Program 3	^x!"334%'dfs6456

Metamorphic viruses

SOPHOS

a. An early generation:

```
C7060F000055        mov    dword ptr [esi],5500000Fh
C746048BEC5151      mov    dword ptr [esi+0004],5151EC8Bh
```

b. And one of its later generations:

```
BF0F000055          mov    edi,5500000Fh
893E                mov    [esi],edi
5F                  pop    edi
52                  push   edx
B640                mov    dh,40
BA8BEC5151          mov    edx,5151EC8Bh
53                  push   ebx
8BDA                mov    ebx,edx
895E04              mov    [esi+0004],ebx
```

c. And yet another generation with recalculated ("encrypted") "constant" data.

```
BB0F000055          mov    ebx,5500000Fh
891E                mov    [esi],ebx
5B                  pop    ebx
51                  push   ecx
B9CB00C05F          mov    ecx,5FC000CBh
81C1C0EB91F1        add    ecx,F191EBC0h ; ecx=5151EC8Bh
894E04              mov    [esi+0004],ecx
```

SOPHOS

Viruses and virus detection

Finding viruses

www.sophos.com

Finding viruses

SOPHOS

- With a simple executable virus this is quite straight forward:
  - Use the header to find the entry point
  - Read the code bytes at the entry point
  - Compare these bytes to the virus identities
- Win32 virus identity

Proprietary information removed

Detection of encrypted viruses

SOPHOS

- The decryption loop is constant
- We can simply search for the encryption loop itself, and/or the virus in it's encrypted form
- Encrypted virus are almost as easy to process as unencrypted ones
- Polymorphic viruses are more of a problem

Detection of polymorphic viruses

SOPHOS

- Originally detection was done using only our ‘Virus Definition Language’ (VDL) to follow the decryption loop
- The identities were very complex and have to be run on every executable file
- Care was required to avoid the identity being too slow

Identity for a polymorphic virus

SOPHOS

Proprietary information removed

The solution is … an emulator

SOPHOS

- The virus engine now includes an emulator to simulate a program running on a Windows PC
- At its heart is a simple loop:
  - Read code bytes
  - Decode code bytes
  - Execute the decoded instruction
  - Track any writes to memory
  - Move to the start of the next code bytes

What does it do?

SOPHOS

- When it is run on a polymorphic virus, the emulator will track the decryption of the virus body as it is written to memory
- In most cases, we only need to decrypt some of the virus body to recognise the virus
- This constant decrypted virus body can then be given to VDL. This makes detection of polymorphic virus much simpler

The benefits of the emulator

SOPHOS

- Creating identities is easier and quicker:
  - The analysts don't have to study the virus code in as much depth
- Scanning files is quicker:
  - The emulator is run on every executable file. Luckily most clean executables stop emulating after only a few instructions and so generally emulation is quick
  - The complicated polymorphic identities are no longer used so the overall scanning is faster

Emulator virus identity

SOPHOS

Proprietary information removed

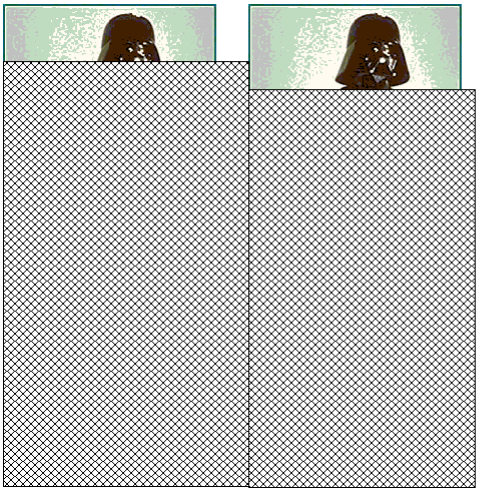
Problems with emulation

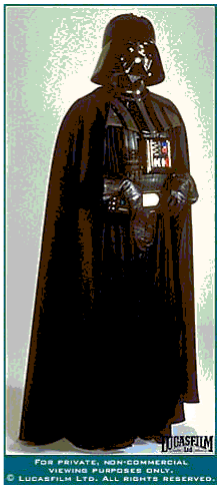
SOPHOS

- The main problem is knowing when to stop
- The emulator needs to be run long enough to decrypt some of the virus body
- But it must not slow down the scanning of clean files
- One solution is to use VDL to control the emulator

Decryption of the virus body

SOPHOS





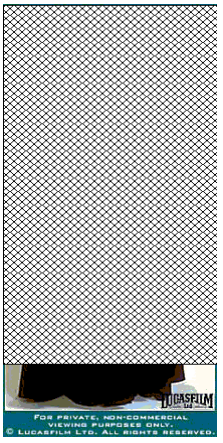
Tracking the virus body

SOPHOS

- The decryption of the virus body can happen in many ways
- The simplest is a forward growing continuous block
- Other methods are used just to make detection harder. For example:

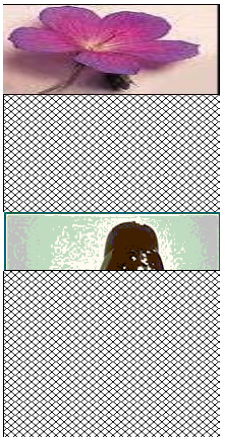
Backwards

SOPHOS



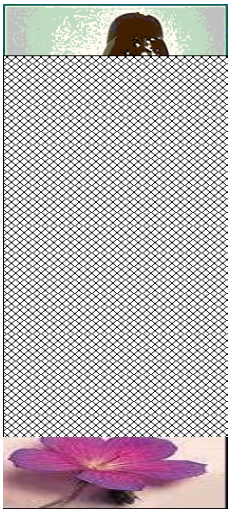
Nested

SOPHOS



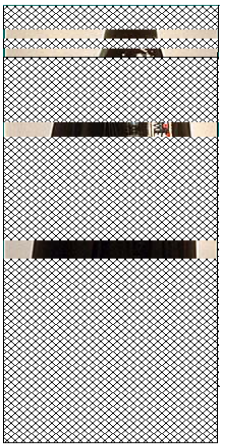
Multiple buffers

SOPHOS



Random order

SOPHOS



Anti-emulation tricks

SOPHOS

- Virus writers put tricks in their virus code to make emulators fail:
  - Calls to dll functions
  - Do nothing loops
  - Junk instructions
  - Using structured exception handling
  - Random virus code execution
  - Checking elapsed time
- An emulator has to keep changing as the virus writers come up with new tricks

New challenges

SOPHOS

- .Net Framework for Windows XP:
  - The executables for XP have an extended PE file format
  - There are no viruses at present
  - They have a new instruction set which is complied when the file is run (JIT compilation)
- Metamorphic Viruses:
  - No constant virus body anywhere
  - The actual virus code changes

W95/Zmist

SOPHOS

- "Undetectable Virus Technology"
- ZMist has a bit of everything:
  - Sometimes it has polymorphic decryptors
  - It is metamorphic
  - It merges itself with the original file's code
- How can we detect ZMist?
  - Look at the file structure. This will give false positives
  - Use the emulator to look for important instructions

SOPHOS

Viruses and virus detection

Paul Jarvis

Virus Engine Team Leader

Sophos plc

www.sophos.com