Seminar Malware
Daniel Loebenberger
B-IT Bonn-Aachen International Center
WS 2007/08

Cohen and the First Computer Virus

From: Wolfgang Apolinarski

What do we want to discuss today?

- Short biography of Fred Cohen
- Virus – The theoretical view
- Between ideas and reality
- Virus – Practical experiments

Short biography of Fred Cohen

- Short biography of Fred Cohen
- Virus – The theoretical view
- Between ideas and reality
- Virus – Practical experiments

Short biography of Fred Cohen

- Professor of Computer Science / Electrical Engineering
  - 1985 – 1988
- One of the first virus researcher
  - Wrote several papers (1987, 1989…)
  - Did several proofs with Turing Machines
- Member of ACM, IACR, IEEE, etc.

Short biography of Fred Cohen
Today‘s activities

- Deception Toolkit (Linux)
  - Honeypot, created ~1998

- Security consulting service
  - Business inspections
  - Employee security training

Short biography of Fred Cohen
Today‘s activities

- Also does
  - Digital forensics
  - Digital crime scene reconstruction

Short biography of Fred Cohen

- Virus research was complicated
  - No “real” virus existed “in the wild”
  - Nobody wants to have
    - “dangerous” experiments in their PC-environment
    - Encourage students to program a virus

- In his theoretical paper, all the helpers are only given by their first names!
  - “sensitive nature”

Virus – The theoretical view

- Short biography of Fred Cohen
- Virus – The theoretical view
- Between ideas and reality
- Virus – Practical experiments

Virus – The theoretical view
Definition of “computer virus”

- “We define a computer ‘virus’ as a program that can ‘infect’ other programs by modifying them to include a possibly evolved copy of itself.”
  - By F. Cohen, “Computer Viruses”, 1987

Virus – The theoretical view

- Infect -> spread through a computer or a network
- Every infected program also acts as virus
  - Exponential growth
    - But infected programs can‘t be infected twice!
- Evolving ~ some kind of polymorphism
  - Virus detection is more complicated

Virus – The theoretical view

- An example virus

```
program virus :=
{1234567;

subroutine infect-executable :=
   {loop: file = random-executable;
   if first-line-of-file = 1234567
         then goto loop;
   prepend virus to file;
   }

subroutine do-damage :=
   {whatever damage is desired}

subroutine trigger-pulled :=
   {return true on desired conditions}

main-program :=
   {infect-executable;
   if trigger-pulled then do-damage;
   goto next;
   }

next:}
```

Virus – The theoretical view

- Is a virus detection possible?
  - The determination of: Given a program P, “Is P a virus?” is undecidable.
  - Say there exists a decision procedure ‘D’, which decides ‘V’ is a virus, if ‘V’ infects another program.
  - So virus-‘V’ is detected by ‘D’.

Virus – The theoretical view

- But now we modify ‘V’ to ’CV’
- ‘CV’ will not infect other programs, if ‘D’ decides, that ‘CV’ is a virus.
- If ‘D’ decides ‘CV’ is not a virus, than ‘CV’ will infect other programs.
- So ‘D’ is not the desired decision function
  - Because ‘D’ was an arbitrary function, this function does not exists.

Virus – The theoretical view

- So we can‘t decide if a program is a virus or not.
- Other proofs about viruses are done by Cohen using a Turing Machine.
  - We will now see one example

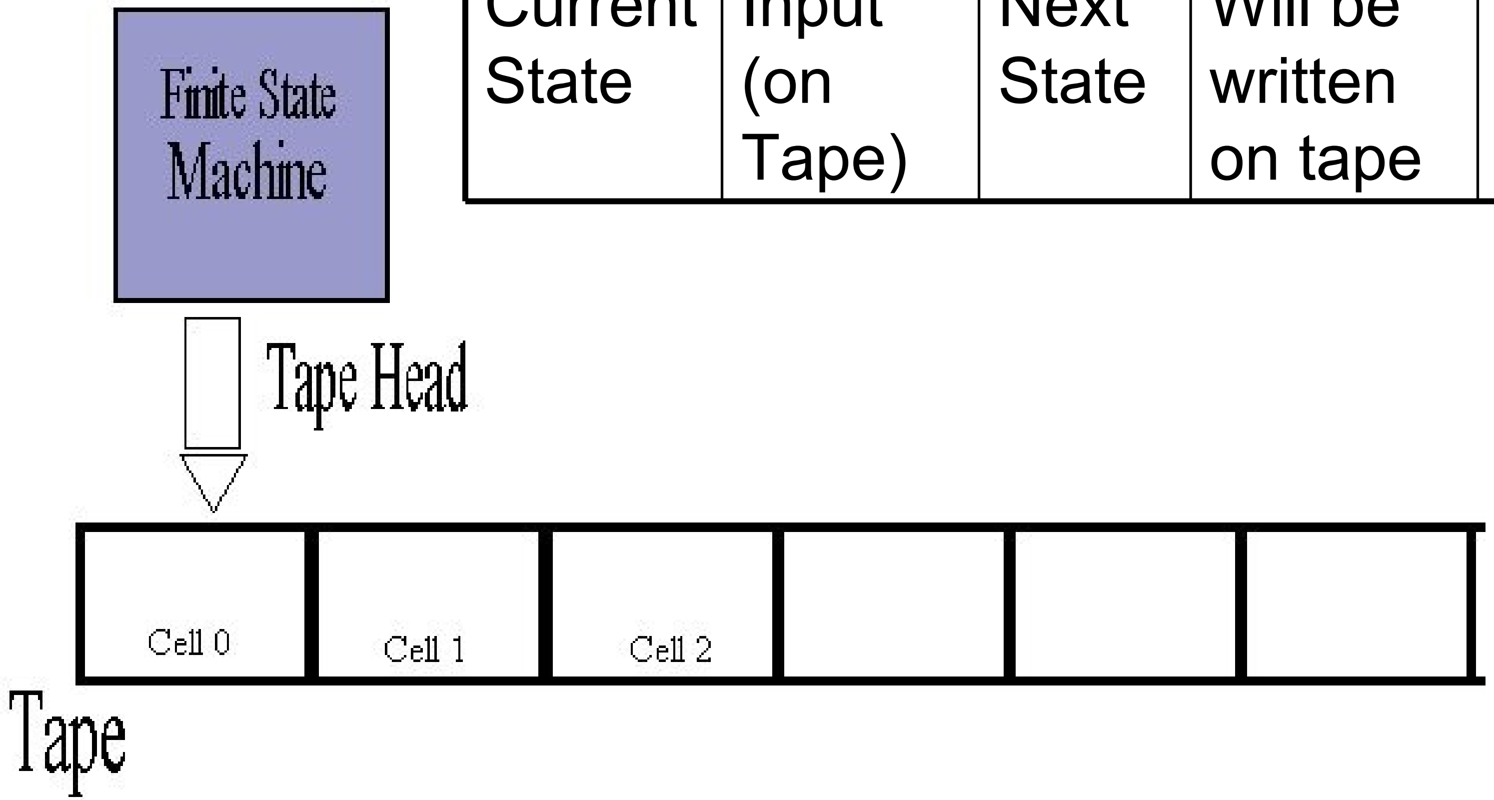
Virus – The theoretical view
Turing Machine

- A Turing Machine has the following characteristics:
  - A finite number of states
  - A tape head
    - Moving is possible in different directions (-1;0;+1).
  - A semi-infinite tape (only in one direction)

Virus – The theoretical view

Turing Machine

S	I	N	O	D
Current State	Input (on Tape)	Next State	Will be written on tape	Tape Head Movement



Virus – The theoretical view

- We also use ‘macros’ here
  - So our turing machine table can be shorter
    - I only show a short description of these ‘macros’.
- C(0,1,2)
  - Changes every occurrence of ‘0’ on the tape to ‘1’ until it reads the ‘2’ on the tape.
    - Moves right while doing this, next state is the state before the current state

Virus – The theoretical view

- L(0)
  - Moves left, until it reads the ‘0’ on the tape
    - Movement (-1), next state after reading ‘0’ is the state after the current state.
- R(0)
  - Moves right, until the ‘0’ occurs in front of the tape head
    - Movement (+1), next state after reading ‘0’ is the state after the current state.

Proof by demonstration…

S	I	N	O	D
S_0	L else	S_1 S_0	L X	+1 0
S_1	0	C(0,x,R)		
S_2	R	S_3	R	+1
S_3		S_4	L	+1
S_4		S_5	X	0
S_5	L(R)			
S_6	L(X or L)			
S_7	L X	S_{11} S_8	L 0	0 +1
S_8	R(X)			
S_9	X	S_{10}	0	+1
S_{10}		S_5	X	0
S_{11}	R(X)			
S_{12}		S_{13}	0	+1
S_{13}		S_{13}	R	0

Virus – The theoretical view

- So what does this Turing Machine do?

L	0	R

Start with L

L	0	R

Change 0 to X till R

L	X	R

Read R, write R, +1

L	X	R	L

Write L, +1 (S_4)

L	X	R	L	X

Write X, L(R) (S_6)

L	X	R	L	X

L(X), S_7

L	0	R	L	X

R(X), S_9

L	0	R	L	0	X

X, L(R), S_6

L	0	R	L	0	X

L(L or X), R(X)

S_{12}

L	0	R	L	0	0	R

Huh? S_{13}

Halt State

Virus – The theoretical view

- So the L0R on the tape changed to L00R
  - So it is not a “simple” virus, it is polymorphic
- And we’ve shown another thing
  - If this virus would not have a halt state, but instead repeat his program, what would happen?

Virus – The theoretical view

- It would write infinite often the L0..0R phrase to the tape
  - Exactly: Countable infinite often
- Conclusion: There exist countable infinite viruses.
- But there also exist countable infinite number of different programs on a TM
  - So there exist as many viruses as programs!

Virus – The theoretical view
Summary

- A computer virus
  - Infects other programs
  - Can evolve (polymorphism)
- There exist as many viruses as programs on a computer

Between ideas and reality

- Short biography of Fred Cohen
- Virus – The theoretical view
- Between ideas and reality
- Virus – Practical experiments

Between ideas and reality

- Are there potential benefits of viruses?
  - Yes!
- A compression virus which compresses binary files after infection
  - Could save over 50% of space normally taken by executables
    - In the eighties hard disk space was expensive!

Between ideas and reality

- This “virus” should ask the user for permission
  - So it is no Trojan horse, but a virus!
- Today many executable are already compressed
  - So no need for a compression virus?

Between ideas and reality

- Benevolent viruses?
  - Cohen did write a paper about this in 1991
- Viruses for everything
  - Maintenance tasks
  - Garbage collection
  - etc.
- If one virus would fail, another would take his place

Between ideas and reality

- Man only needs to write a successor virus for a ‘program update’
- Distributed calculations with viruses?
- Failsafe database with virus support
  - A bill collector virus

Between ideas and reality

- So the whole database is distributed along the network
- No regular “scanning” for a bill is necessary
- The viruses awake by themselves and ‘learn’ when they have to be active

- Artificial Life!

Between ideas and reality
Prevention of viruses?

- If sharing is allowed, a virus can spread to every user who takes part at the sharing
  - Virus paths are transitive!
- If modification of software is allowed, than a virus can reach new programs.

- Disallowing one of these?
  - Unacceptable, especially if teamwork is desired

Between ideas and reality

- “Isolationism”
  - Gameboy, other games consoles
    - Sharing is not allowed, but modification
      - Save games!
  - Non-updatable firmware
    - DVD-Players, etc.
      - But most do have a flashable ROM!

Between ideas and reality

- So “Isolationism” is not a solution
- Some complicated security policies?
  - Unix file systems / NTFS partitions
    - Only slow down virus distribution, because not all users are affected
- New Idea: “Flow distance”

Between ideas and reality
Flow distance

- Special metric, that keeps tracks of the number of sharings, ie. the data flow
  - Max(distance(process),dist(file))+1
    - If it is greater than a threshold, access is denied
  - But if all users have direct connections, this doesn’t help a lot.

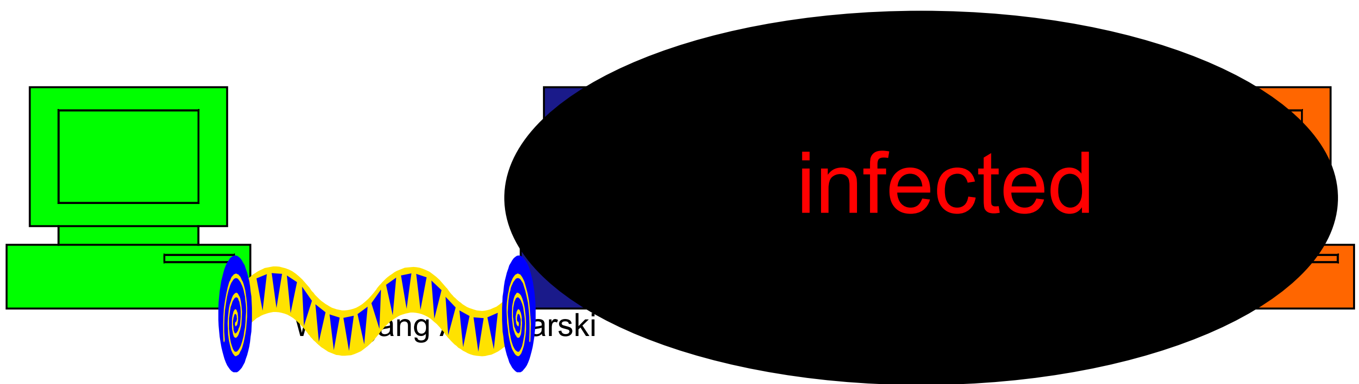
- ‘Flow list’ lists all users that had effect on an object

Between ideas and reality

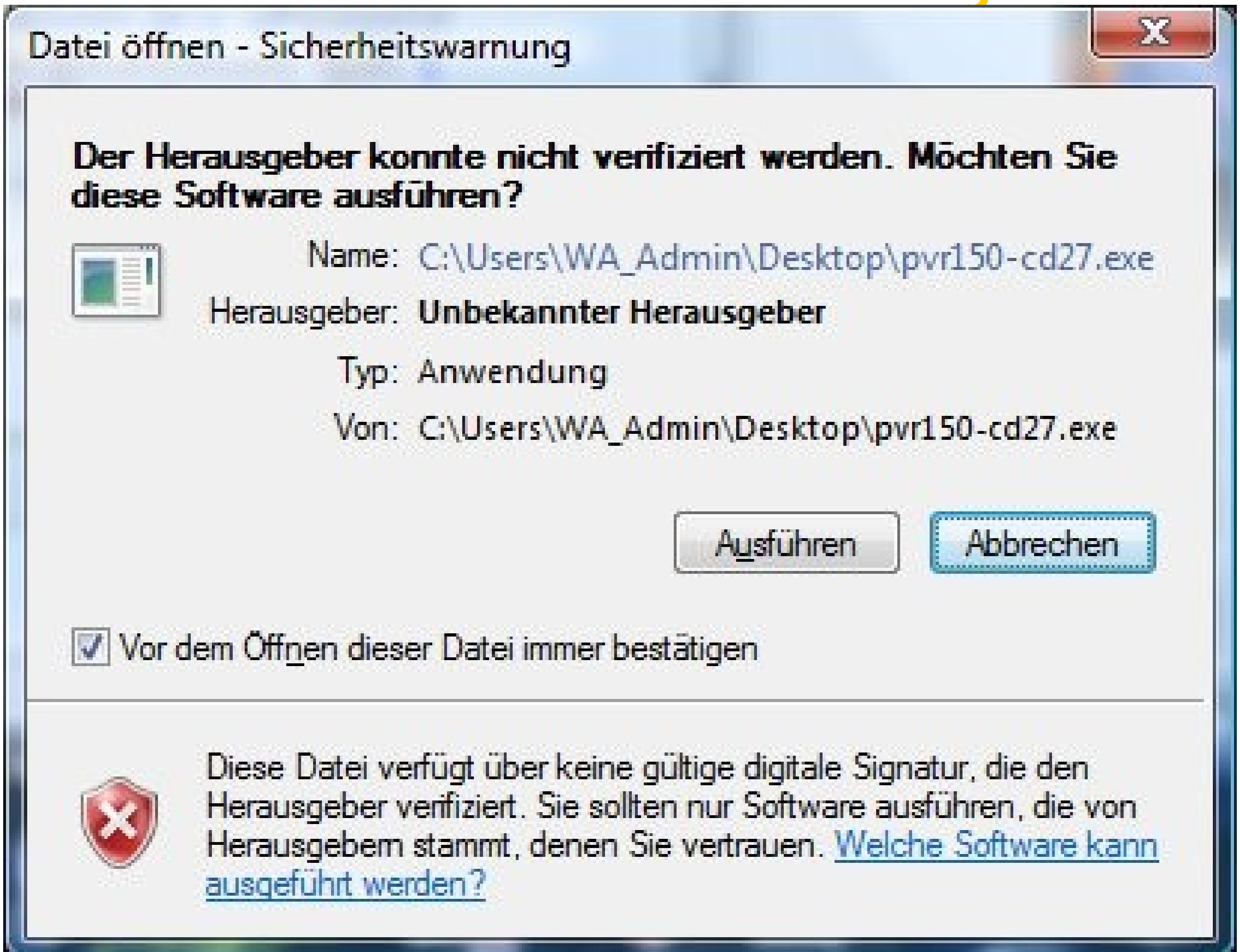
Flow distance

- Access is only granted, if a ‘trusted’ user has touched the object
- A metric is also possible:
  - Only access files where ≤ 2 users were involved
- Files of a distrusted user can be fully ignored

- With this distance metrics a virus spread could be slowed down or stopped



Between ideas and reality



Between ideas and reality

Between ideas and reality
Summary

- Useful viruses
  - Distributed computing, compression virus
  - Artificial Life
- Prevention - “Isolationism”
  - Games console - firmware
  - Flow distance

Virus – Practical experiments

- Short biography of Fred Cohen
- Virus – The theoretical view
- Between ideas and reality
- Virus – Practical experiments

Virus – Practical experiments

- How to study the behaviour of a computer virus?
  - No virus existed in 1983
  - So instead of using an existing virus, a new one was written
- On the 3rd of November 1983 conceived
  - On the 10th presented
    - In a seminar on computer security
  - 8 hours of (expert) work

Virus – Practical experiments

- The virus infected a unix program called “vd” and spread using the system bulletin board
  - No damage routine, only creates reports
  - Traces to detect the virus everywhere
- Five experiments took place

Virus – Practical experiments

- The attacker got all system rights in an average of 30 minutes!
  - Everybody was surprised about the short time, the virus had “success”
- As result the administrators did not allow any other virus experiments to take place

Virus – Practical experiments

- So it was not intended to establish more security, but to “stay” at the current level
  - If no virus exists, no anti-virus actions had to be taken

- Other experiments were planned and viruses for different systems written
  - After several months the administration decided to not allow this experiments
    - The security officer even refused to read the proposals

Virus – Practical experiments

- So it was not allowed to add traces to the system, to discover a potential virus attack

- This reactions were typically for this time
  - Computer system were expensive so buying equipment only for virus testing was quite unrealistic
  - A “real world” scenario can’t take place in a sandbox

Virus – Practical experiments

- In 1984 a virus on a system which used the Bell-LaPadula security policies was developed
  - Bell-LaPadula allows a lower user not to read the higher users file. A higher user is not allowed to write in a lower users file
    - Security of information
    - System was in use by the US Air Force

Virus – Practical experiments

- The virus needed 20 seconds for each infection!
- After 18 hours the first infection was performed
- After 26 hours the virus was shown to administrators and programmers
  - It could cross all security boundaries, write down and read up…

Virus – Practical experiments

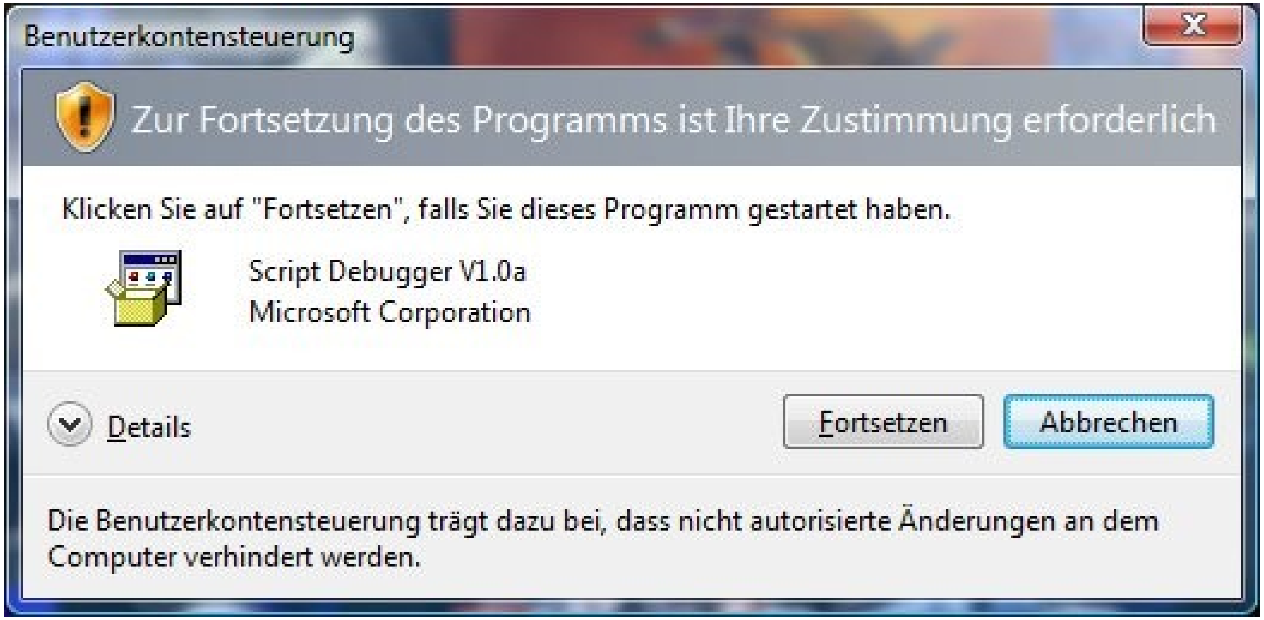
- On an unix system the infection was slowly, until it reaches a system administrator account
  - Especially “root”

Virus – Practical experiments
Results / Countermeasures

- Seperate system administrator accounts and the normal user account
  - This seperation was never really thought of.
  - If a user announces a new program, one of the first users always was a system administrator…
    - Virus spreading is made very easy…

Virus – Practical experiments

- This discussion also applies to today's computers
  - Windows – Vista's new behaviour



Virus – Practical experiments

- He thought of developing an antibody for a virus
  - Which also evolves by itself, in addition to human development
- He never used the term “Anti-Virus”

Virus – Practical experiments Summary

- How to study virus behaviour?
  - Write an own virus
    - Study its behaviour
- Administrators & security personnel might not be helpful
  - Threats are everywhere ;-)
- Viruses spread very fast, if a computer user uses his normal administrator account only