Computer Viruses
Theory and Experiments

By Fred Cohen

Presented by Alexandru – Gheorghe Grigoraș
At Papers We Love Bucharest
January 11th 2016

In the beginning...

- John von Neumann's idea of artificial life
- Cellular automatons
- Conway's game of life (demo)
- Von Neumann's Universal Constructor constraint
  - Leads to huge automations
- Removal of the constraint
  - Much simpler automatons (demo)

What's this got to do with viruses?

- Computer viruses are like reproducing automatons
- They reproduce their code inside other programs
- Being a virus is not bad in itself
- Other IT things can be viruses (can reproduce)
  - Can you guess what they are?
  - Hint: viral (adj.) = to have the properties of a virus

Let's get on with it!

- The paper is a part of Fred Cohen's PhD thesis (1986)
- It is taken further by Eric Filiol's book, which inspired me to write this presentation
- See bibliography

The paper's intro

- "This paper defines a major computer security problem called a virus"
  - So the concept of a virus has been analysed for just 30 years
- "little work has been done in the area of keeping ***information*** entering an area from ***causing damage***"
  - Different perspective on the separation between program and data

Hello 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:}
```

Fig 1 Simple virus 'V'

Some key aspects

- Low detectability = longer life
- Trigger so not to trip any alarms until infection spreads
- Prevent multiple infections of the same file
  - The check for 1234567
  - Files with 2 identical starting blocks are suspicious
- fread and fwrite at the beginning of execution stick out
  - Use system(cp target = source) over fread(source) fwrite(target)

Infection Prevention

- "*(Viruses can spread from user A to B to C)* with the witting or **unwitting** cooperation of user B"
- Paper looks at Bell-LaPadula, Biba and flow models
- These were important isolation models, used in production when the paper came out
- More details about these models in Andrew Tannenbaum's book (see bibliography)
- "information only has meaning in its interpretation"
  - Gmail detects virus source code in PDFs

Bell-LaPadula & Biba

- Bell-LaPadula
  - Users on a security level can't read things with lower security and can't write things with higher security
- Biba
  - Like Bell-LaPadula but the other way around
  - Users can't read things with higher security and can't write things with lower security
- Neither can prevent a virus from spreading

Flow Models

- User's can't send/receive information from more than N hops away
- High complexity
- Do not prevent viruses from spreading either

Generic virus detection

- Detect if a given program is a virus
  - Theoretically impossible (say Cohen, Filiol, Gödel, Turing et. al)

```
program contradictory-virus :=
{...

main-program :=
   {if ~D(contradictory-virus) then
          {infect-executable;
          if trigger-pulled then
                  do-damage;
          }
   goto next;
   }
}
```

Fig 6 Contradiction of the decidability of a virus 'C

How about we make a list

- Can we make a list of all known viruses and use it to protect ourselves?
- No, because some viruses evolve (change shape)
  - When infecting executable insert random useless instructions, which break comparison

Can we detect viral evolution?

- No.
  - We get the same contradiction

```
program undecidable-EV :=
{...
subroutine copy-with-undecidable :=
  {copy undecidable-EV to
        file till line-starts-with zzz;
  if file = P1 then
        print ("if D(P1,P2) print 1;");
  if file = P2 then
        print ("if D(P1,P2) print 0;");
  copy undecidable-EV to
        file till end-of-input-file;
  }
main-program :=
  {if random-bit = 0 then file = P1
        otherwise file = P2;
  copy-with-undecidable;
  zzz;
  infect-executable;
  if trigger-pulled then do-damage;
  goto next;}
next:}
```

Fig. 8 Undecidable equivalence of evolutions of a virus 'UEV'

Detection and prevention conclusions

- Viruses cannot be 100% detected
- Viruses cannot be 100% prevented from spreading
- ...except when every computer is isolated from every other computer
  - Including no USB ports, no typing programs from magazines and compiling them, nothing
- In this day and age it is not generally possible
- If someone puts her mind to it, any antiviral protection can be broken

Theory and **Experiments**

	unixC	B-L	Instr	Shell	VMS	Basic	DOS
time	8hrs	18hrs	N/A	15min	30min	2hrs	1hrs
inf t	.5sec	20sec	N/A	2sec	2sec	15sec	10sec
code	200L	260L	N/A	7L	9L	30L	20L
trials	5	N/A	N/A	N/A	N/A	N/A	N/A
min t	5min	N/A	30sec	N/A	N/A	N/A	N/A
avg t	30min	N/A	30min	N/A	N/A	N/A	N/A
max t	60min	N/A	48hrs	N/A	N/A	N/A	N/A

Fig 11 Experimental results

- time = Time to code
- inf t = time from introduction to first infection
- code = size of code
- min t = minimum time until virus infected highest privilige user

Theory and **Experiments**

- So a 200 line virus made in 8 hours took **half a second** to infect the first program and **5 minutes** to infect a program with superuser permissions
- How about that?

Back in the day

- In '86 there were lots of different OSs and systems
- 200 lines of code took 20s to run

Why bother with viruses?

- For profit
- For fame
- For politics
- For spoilers
- For military stuff

Nuclear power plant strikes Then and Now

- In 1981 Israel wanted an Iraqi nuclear power plant out of commission
  - 6 F15s, 8F16s, 10 dead Iraqis, 1 dead Frenchman
- In 2012 Israel wanted an Iranian nuclear power plant out of commission
  - Stuxnet, a virus, broke the equipment's firmware in an undetectable way
  - Everything seemed ok on the instrument panels, but the uranium wasn't being processed

Why it might all be useless

- In December 2015 a Russian vessel with submersibles hung around an undersea internet cable connecting the US to Europe
- The US freaked out
- All the trouble went through to give hightech security can be nullified by a Russian cutting a cable

Bibliography

- “Computer networks”, Andrew S. Tanenbaum
- “Computer Viruses: from theory to applications”, Eric Filiol
- “Computer viruses”, Fred Cohen, disertation, University of Southern Californaia, January 1986
- Golly, a cellular automaton simulator http://sourceforge.net/projects/golly/files/golly/golly-2.7/