Viruses

- A virus is a self-replicating program which attaches to other files or disc/floppy sectors and spreads in this way.
- A virus may have a payload doing other things, like destroying files, corrupting data etc. Most viruses are not harmful, their payload for instance creates clicks when keyboard is pressed or shows a window in the screen.
- Many harmful effects of viruses are caused by sloppy coding. Typically the virus overwrites some data and does not save it which crashes or jams the operating system or a program.
- Viruses written for one environment may have unintentional harmful effects in another environment. Unintentional harmful effects caused by misfunctioning can often be corrected by antivirus software since the virus has saved the data somewhere. Antivirus program may also use checksums for correction of small errors.

Viruses

- Most of the viruses are written for DOS, so they function in DOS, Windows 3.x, Windows 95 and Windows 97.
- DOS viruses are usually divided into:
  - Master boot sector viruses
  - Partition boot sector /floppy boot sector viruses
  - File viruses
- Macro viruses are another type of viruses. They are platform independent and can spread in MicroSoft applications using macros (Word for Windows, Excel).
- There are few viruses for Unix since access controls on files and directories limit the ability of a virus to infect enough files to survive. Worms are more important threats for Unix.
- Still, there is at least one reported Linux-virus: Bliss. It probably is a true virus though it may be a trojan horse.

Viruses

- DOS-viruses are written in Intel’s assembly language. This used to limit virus writing to people with sufficient skills in assembly, but nowadays there are virus creation kits available from the Internet. Some purported kits from maximal Security book:
  - Virus Creation Laboratories
  - Virus Factory
  - Virus Creation 2000
  - Virus Construction Set
  - The Windows Virus Engine
- There are several how to make viruses books. I can mention two books, though they deal with old style DOS-viruses:
  - Rune Skardhamar: Computer Viruses: Discovery and removal
  - Andrzej Dudek: How to write viruses (in Polish)

Boot viruses

- Boot viruses are very serious viruses. Though there are very few of them (only 5% of the known more than 7000 PC viruses), they represent about 85% of the reported problems. This is because file viruses are much more easily removed by antivirus software.
- A master section boot virus (MSB-virus) is attacking the master boot record (MBR) in the hard disc. MBR is always in the same place in the hard disc: track 0, head 0, sector 1.
- During hard drive bootup, the ROM BIOS boot program loads the MRB from the primary hard disc connected to the computer.
- Then it verifies the signature of MBR at the end of the sector. If the signature matches, it transfers control to the MBR’s boot program.
- This boot program checks which is the active disc partition.

Boot viruses

- If there are more than one or no active disc partitions, the master boot programs gives an error. If there is one active partition, the boot program loads the Partition Boot record and checks its signature.
- It the signature matches, the PBR’s boot program is started and takes over.
- MBR is then not aware of the operating system: there can be several partition boot records for Windows, DOS, Linux etc. No software antivirus program is active when the MRB’s boot program is executed, so it is a very good place for a virus to attack.
- A virus replaces the MBR with its own version. It must save the original MBR as it is vital to run the MBR’s boor program or the computer will not boot. Most viruses save MBR to one of the typically unused sectors after the original place of MBR.

Boot viruses

- Then the viral MBR boot program is run every time the computer boots from a hard disc. The virus tries to set itself as a memory resident program (TSR) and to replace an address in the interrupt vector table (IVT).
- The interrupt vector table contains the addresses of ROM BIOS services and there are software interrupts to DOS and mouse services.
- Hooking (replacing the address, directing it to your routine) to some interrupts, like mouse, can be done by an ordinary program, For ROM BIOS and DOS interrupts you need a TSR program to hook into them.
- By hooking for instance to DOS interrupt to open a file, a virus is run every time a file is opened from DOS.
- In DOS it is usually much easier to use DOS-interrupts for file handling than directly ROM BIOS.

Boot viruses

- Partition boot virus attacks a partition boot record. There are much fewer PBR-viruses than MBR-viruses since it is more difficult to write them. The virus first has to determine the start of the PBR and they depend on the operating system.
- PBR has its own BIOS parameter block, which describes the important attributes of the hard drive. The parts of PRB which are essential are the BIOS parameter block and the signature.
- If a viral PBR has these correct, it can mimic PBR. The virus replaces only the boot program. While PBR-viruses are rare, one of the most common virus Form is a PBR-virus.
- When the viral PBR boot program is run, the virus installs itself as a TSR-program in the same way as in MBR-virus.
- The PBR-virus saves the original PRB at the end of the hard disc and runs it after it has installed itself.

Boot viruses

- The original PBR bootstrap routine then starts the operating system and the user gets the ordinary prompt.
- If the hard disc is very full, the virus may cause damage by overwriting data with the original PBR.
- A floppy boot record (FBR) virus is rather similar to the PBR-virus. It replaces on a floppy floppy boot record (FBR), which contains BIOR Data Area (BDA) corresponding to the BIOS parameter block. FBR contains a signature just like PBR.
- The floppy boot record is checked even in an earlier stage than the master boot record. The ROM BIOS bootstrap routine checks for peripherals and if there is a floppy in the boot floppy station, the computer boots from the floppy.
- It is possible to disable booting from a floppy in most PCs but then recovery requires a new hard disc of the same type.

Boot viruses

- Most FBR-viruses try to install themselves as TSR-programs and hook on some DOS or BIOS services by modifying the address in the interrupt vector table.
- A FBR-virus saves the original FBR in the floppy (often in the last sector in the root directory) and after it has done its installation, the virus runs the original FBR boot program.
- FBR-virus typically modifies the field Total memory in kilobytes in the BIOS Data Area to reserve place in memory for itself. Usually there is 640 kbytes, it can be reduced to, say 638 kbytes for a 2 kbyte virus, then loading the operating system will not overwrite the virus.
- FRB-virus tries to copy itself also on the master boot section, sometimes to the partition boot sector. This is because if the computer is later booted from the hard disc (like if the floppy has no DOS), the virus will be loaded to memory.

Boot viruses

- Removing a partition boot virus can be made by formatting the hard disc.
- Formatting the hard disc does not write over the master boot sector, so MBR-virus is not removed.
- Formatting unconditionally (format a: /U) will remove a floppy boot record virus.
- Boot viruses can do damage in two ways:
- Unintentionally by overwriting some data while saving the boot records or by other programs overwriting the partition tables the virus uses. These cause the system not to boot.
- Intentionally by writing, possibly with random reads and writes, data in the disc.
- Boot viruses infect floppy boot sectors in order to spread.

File viruses

- File viruses are by far the most popular form of DOS-viruses. They infect COM, EXE, overlay and rarely SYS files and can spread fast to a large number of files.
- File viruses do not have a full control of the computer, as the boot viruses have, and therefore an antivirus program has an easier job catching them.
- The different types of executable files in DOS and in Windows have a different structure, so a virus is typically attacking only one type of executable file.
- COM-files are the most simple. they have a starting point and after that comes the code, restricted in length to 64 kbytes.
- A virus can add itself to the beginning of the COM-file after the start and appending the original file after the virus part.
- A more common way for the virus is to append its code to the end.

File viruses

- A COM-virus appending to the end saves the three first bytes of the original program, overwrites the 3 first bytes with a jump instruction to the start of the virus at the end of the program. Then comes the virus code, then the three original bytes of the program and a jump to the start of the original program.
- In this basic version it is easy for a antivirus program to guess, that a jump in the beginning of a COM-file is probably a jump to the virus code.
- There are some ruthless COM-viruses which simply overwrite the original COM-program and then display a message such as no memory to trick the user to think that the program does not work because of memory problems, then the virus has already spread. A virus working in this method is easily noticed.

File viruses

- If the virus does not destroy the original code, but saves it (not for kindness but to stay undetected), an antivirus program can often clean the virus out of the program.
- Removing viruses is best to be left to antivirus programs as the modifications the virus did can be tricky.
- There can be a problem with a COM-virus if the COM-file has a length close to the maximal length. Then if a virus is appended, the file size is too large and DOS will not run the program.
- An EXE-file has a different structure. EXE-file can be much larger than a COM-file (though there are size limits in DOS). An EXE-file has a starting point indicated by two pointers (code segment CS and instruction pointer IP).
- A typical EXE-virus records these pointers so that it can later call the original program.

File viruses

- The virus appends itself to the end of the program.
- It could prepend itself to the beginning of the program, but inserting itself in the middle is likely to interfere with the programs operation and mess up long and short jumps so that the program will not work. Inserting in the middle is rare.
- Then the virus replaces the CS and IP pointers by its own code segment and offset.
- The virus changes certain other fields in the header to reflect the changed size and to inform the virus that the program is already infected.
- In general, a virus must not infect a file more than once, nor can it install itself more than once as a memory resident program hooked in a service. Doing so too many times would stop the routine from working.

File viruses

- This is why most viruses mark infected routines in some way so that they can detect that they are infected.
- Typical ways are looking form the virus fingerprint (i.e. typical sequence of instructions), or by modifying file date of time second field (which DOS never sets).
- Some viruses, notably some memory resident viruses, can avoid this marking by the logic - if they are memory resident, the system is infected, so it needs not to be reinfected.
- An antivirus program can naturally look at the same marks if the virus is known.
- A SYS-file has two entry points strategy and interrupt. They are addressed by pointers like EXE-files. A SYS-file has memory restrictions.
- SYS-viruses are rare, probably since spreading is limited.

File viruses

- DOS-viruses are common, but very few people use DOS directly (I am one still). A special problem for these viruses comes from Windows.
- A windows EXE-file has a DOS-exe header, which writes this program can only be executed from Windows. The size of this DOS-exe-file is very small, while the actual Windows program is quite large. A simple DOS EXE-virus may get confused and overwrite toe actual program.
- There are native Windows viruses, especially for Windows 3.x and 95. There are native Windows NT-viruses, but most PC-viruses cannot cope with 32-bit code and can only be executed as 16-bit Windows 95 applications or in the DOS-window.
- The simulated DOS can restrict the viruses considerably.

File viruses

- While to programs executed in Windows NT can call DOS and BIOS-interrupts, they do not have a direct access to the hard disc, but Windows NT intercepts the calls.
- It should be impossible for instance to change the mode from 16-bit unprotected mode to 32-bit protected mode because the service should not be available to the simulated DOS.
- I have not checked this and cannot say if it is impossible, it is possible to write directly to the screen, write using DOS to all files in the hard disc (unless they are protected which they probably are not in one user computer). It is not possible to overwrite CMOS or flash ROM in the same way as in DOS.
- However, I would not be sure about it. While NT security architecture Discretary Access Control (DAC) is though to be very good and it restricts direct access to the hard disc, mostly Windows security is based on unavailable documentation.

File viruses

- Poor documentation requires reverse engineering which takes time but will be done. The fact that some of my DOS-programs (doing rather low level things) work in NT means that it may allow DOS-programs to do too many things for security. In general, if old programs should work and they do things you should not do, how do you provide security?
- DOS-viruses are written in Intel’s assembler. For the time being it seems that writing macro viruses in Visual Basic would become much more popular. Still good knowledge of assembly language is essential for a virus writer.
- Assembly programs are very small and fast and they are not so much DOS-programs but actually Intel PC programs. The use of DOS interrupts for file access is nice, but apart from that most operations are equally valid for NT or Linux.
- Worms are possible in NT, so viruses and worms written in 32-bit assembler will surely appear.

Virus behavior types

- DOS-viruses are much older and more versatile than macro viruses, so let us look at some virus techniques.
- A slow virus is a TSR program, which tries to avoid antivirus program from detecting the change in program file sizes. A slow virus will not search a file and modify its length, but it will hook into some interrupt, like the DOS service which COMMAND.COM uses to copy files. Antivirus programs hardly ever check these activities.
- When a file is copied, the slow virus appends itself in the memory to the program. Then the infected copy is written to the disc. Check of sizes will not notice the virus as the file is new.
- A retrovirus is a virus, which tries to disable antivirus programs. If can remove the fingerprint file of viruses and the records of file sizes. As a user can remove these files, so can a virus.

Virus behavior types

- A stealth virus tries to hide itself. Stealthing can be used by a boot or a file virus.
- A stealthing file virus must install a memory resident server to such utilities which can be used to detect a virus.
- A simple example is size stealthing: a file virus hooks a memory resident server to open file DOS-service which detects a file being listed by DIR, if the file is infected, the routine reduces the virus size and shows the original size.
- Content stealthing is a way to hide the virus in the file if it is investigated by some low level editor, which shows all bytes, or by a disassembler.
- A memory resident part of the virus hooked in the file open interrupt removes the virus when it is opened and writes it back when the file is closed.

Virus behavior types

- A companion file virus does not infect a file but makes a viral copy of the file and gets it executed instead of the original file.
- For instance, if there is FOO.EXE file, the virus may make to the same directory FOO.COM-file. DOS will execute the COM-file.
- Another way is to put a new file into an earlier stage in the PATH (or even change the path). Try for fun if you can get in DOS the DIR command replaced by your routine by changing the PATH in AUTOEXEC.BAT. Put your DIR.EXE to call the real DIR, so that the system routines would work. This way of replacing DIR should not work (DIR is a service of COMMAND.COM), but some other programs will.
- A polymorphic virus tries to change its appearance so that a virus scanner could not detect its fingerprint.

Virus behavior types

- A polymorphic virus crypting its own code seems very fancy. The techniques are not very complicated, though.
- A simple algorithm a polymorphic virus could use is to insert instructions, like nop (no operation) and correct the jumps accordingly.
- A virus could also replace instructions with instructions having identical content (like xor ax,ax to mov ax,0).
- The way a virus encrypts its code is typically XOR each byte with a key. XOR:in the encrypted byte with the key produces the original byte, so a virus can decode only that part of the code which is at the moment executed.
- The cryptation possibilities depending on the length of the key. With a one byte key there are 255 possibilities, but with a key of two bytes already 65536 possibilities.

Macro Viruses

- In Word for Windows there is a macro language. Documents containing macros can be only of template types *.DOT, not of document types *.DOC, but Word looks at the file type, not at the extension for determining the type, so any suitable extension for Word can actually be a template file.
- If you have never seen a Word macro, look at the templates in Word, for instance INVOICE.DOT in Winword6\Template.
- INVOICE.DOT has a macro Update, which in a simplified form is in the next page. You have to unprotect the document from the tool menu to be able to edit the macro from tools.
- To make a virus out of macros one can create macros with the names AutoOpen and AutoExec.
- There is a global macro pool for all documents and a local macro pool in the document and macros can automatically copy themselves from the local pool to the global pool.

Macro Viruses

```
Sub MAIN
    fieldName$ =  dlg.Name
    If fieldName$ = "Shipping" Or fieldName$ =
    "SalesTax" Then
            Goto TOTAL
    EndIf
‘ ***  I cut out some code from the Update macro
TOTAL:
    SetFormResult "Total"
End Sub
```

Macro Viruses

- If there is a macro with the name AutoExec in the global macro pool, then each time the Word application is started, this macro is executed.
- Being code in VisualBasic, it can do almost anything any middle level code can do. (It cannot do everything you can do in assembly.)
- If there is a macro AutoOpen in the document’s local macro pool, it is run when the document is opened.
- A virus can use this macro to copy itself to the global macro pool when it is opened.
- A macro in the global macro pool can copy itself in the local macro pool of a new document and set there AutoOpen macro. Then we already have a spreading virus.

Macro Viruses

- There are other ways the macro virus could spread. It could take a list of all valid email receivers in some mailing list and send an email where the macro is included in a document attachment.
- It is basically impossible to stop sending attachments in email because then the email system is not of much use.
- It is easy to create messages which look like coming from a respectable sender and contain an attachment, like the agenda for a meeting.
- Macro viruses a better spreading capability than DOS-viruses because they are platform independent and work equally well in Windows 3.11, Windows 95/97, Windows NT, Macintosh or Linux/Unix which supports MS Word or Excel.
- As macros are written in VisualBasic they are easier to write.
- Antivirus programs are not yet so good in detecting macro viruses, but it is changing.

Virus detection

- Antivirus programs use a number of ways to detect viruses and to remove them.
- Virus scanning is maybe the most important technique. Viruses are detected by comparing their fingerprints to a set of known virus fingerprints.
- Virus fingerprint is a piece of virus code, which stays unchanged in the virus.
- A virus scanner must scan in some seconds a large number of files and comparing the whole code of all programs to a large set of known virus fingerprints would last too long.
- Therefore a virus scanner checks only the beginning and the end of the programs, most viruses attach themselves to the end, some to the beginning, hardly any to the middle.

Virus detection

- Viruses are made by modifying other viruses. Therefore a virus scanner can use a set of flexible rules which catch a set of related viruses. These rules are matching rules with wild cards for some bytes.
- Such rules are effective also against simple polymorphic viruses, but not to better polymorphic viruses using encryptation.
- Some virus scanners try to cope with encrypted viruses by following jumps starting at the beginning of an infected file. These jumps must lead to a piece of unencoded virus code, as the virus must be executing to decode its code. This unencoded piece is used as the fingerprint.
- A different mechanism can be tried for complicated polymorphic viruses: they can be tried to be executed in a virtual environment.

Virus Detection

- If a virus in a virtual environment tries to install an TSR routine or tries to write to files or to boot sectors, they can be detected. Slow viruses are a problem to the method, they do not necessarily try anything. (But finally most of them try install a TSR program. A really slow virus would do so only after some trigger time/condition.)
- Virus scanning must be preceded by memory scanning as content stealthing viruses would clean a virus from an inspected file.
- Memory scanning can effectively find TSR routines. Their hiding possibilities are very limited.
- An antivirus program can also bypass DOS and BIOS routines and investigate files with so low level calls, that hookups in higher levels cannot stealth the virus.

Virus Detection

- Other virus detection mechanisms include:
- behavior blocking: this is a sandbox technique, any program doing something suspicious may be a virus. Problems are false alarms as some programs may be doing strange things. If this is the case, virus protection will be turned off.
- heuristic scanning. look for any parts of code which look like possible viruses, for instance look for AutoOpen macro from Word documents. A heuristic scanner does not need to be updated often but may not find all viruses and may cause false alarms.
- integrity checking: file sizes, signatures, checksums and other similar authentication data is stored in a file and checked if it changes. The problem is that users may legally change the data and therefore a virus may change it also by deleting file size records etc. Integrity check requires memory scanning.

Summary

- Viruses are a very serious threat. While the most popular DOS-viruses become less common as people do not so much boot from floppies, reducing boot viruses and spreading of file viruses is more difficult in an environment with access controls, such as Linux and NT, macro viruses are presently very common.
- It is likely, that multipartite viruses, that is, viruses spreading in several ways will become common.
- There are advantages in assembly written viruses, so mixing them to macro viruses, or installing and starting them with a macro virus could make viruses which are more capable and more able to hide themselves.
- Worms should be taken seriously in a multiprocess environment, a virus is largely a virus since running several processes was not possible in DOS PC.