October 1989 ISSN 0956-9979

VIRUS BULLETIN

THE AUTHORITATIVE INTERNATIONAL PUBLICATION
ON COMPUTER VIRUS PREVENTION,
RECOGNITION AND REMOVAL

CONTENTS

EDITORIAL

October 13th

Two imminent PC virus threats are Jerusalem and Datacrime. The former will delete programs run on October 13th and the latter will do a low level format of track zero of the host PC's hard disk on that day and every day thereafter until December 31st.

Jerusalem is one of the most widespread of all computer viruses which affect IBM PCs. It is not unreasonable to predict that significant loss of programs will occur once its trigger mechanism activates.

Datacrime is a less predictable beast. There is no conclusive data available about the extent of infection. It is unlikely to be detected by visual inspection - how many of us, for instance, know the exact length of the programs we run and would notice an increase of 1168 bytes?

Recent efforts to assess the extent of Datacrime's spread have used scanning programs. One such program was recently run on machines used by a number of UK academic institutes. Not a single case of infection was discovered. Similarly, a specific Datacrime scan program currently in use in the States has not revealed any infection by this virus.

This is not to say that the virus is not a major threat. Those of us who have seen Datacrime trigger in a test environment can testify to its destructiveness. We should also remember that only a tiny percentage of PCs have been examined to determine infection. However, indications are that Datacrime will strike at only a few sites.

The 'Itch Syndrome'

Collating data about computer fraud, hacking or virus attacks is a near impossible task. A closing of the ranks and a veil of silence is a natural response to any incident which indicates mismanagement or carelessness.

Computer viruses, in particular, engender extreme bashfulness in infected parties. Businesses which rely on customer and investor confidence are most fearful of details about such infections leaking. The spectre of data loss and corruption has created a siege mentality. Computer viruses, unlike floods, fires and lightning, cannot be defeated with total assurance. This is especially true for network infections and if information is made public it invariably reduces confidence.

There is also an unspoken belief that there is something "dirty" about these programs. It is as if the unfortunate user had contracted an embarrassing social disease. Conversations with Systems Administrators often start "I have a friend who thinks he might have a problem..." This might be termed 'Itch Syndrome'.

This secrecy, understandable as it is, is against the interests of the computer-using community. Non-disclosure agreements and confidentiality clauses prohibit companies active in the field from providing any details about victim companies. Thus the virus sceptics can continue to belittle the problem and in the process promote a false sense of security whilst others are unable to learn from organisations which have endured computer virus attacks. As events unfold over the coming months a modicum of honesty from those affected will prove beneficial to us all.

TECHNICAL EDITORIAL

Joe Hirst

Virus Bulletin has not received permission to reproduce this article on CD from the author. Readers can obtain a paper copy of the original issue directly from *VB*.

COMMENT

Joe Hirst

Virus Bulletin has not received permission to reproduce this article on CD from the author. Readers can obtain a paper copy of the original issue directly from *VB*.

KNOWN IBM PC VIRUSES

The following is a list of the **known** viruses affecting IBM PCs and compatibles, including XTs, ATs and PS/2s. The list consists of two parts. The first part of the list gives aliases and brief descriptions, and this also includes a section on reported viruses (which may be completely inaccurate). The second part includes the infective length (the amount by which the length of an infected file has increased), the hexadecimal pattern to use for detecting the virus, and the offset of this pattern within the virus. To use this information correctly, the virus entry point is required and this has been added. Viruses referred to in other publications by number almost always refer to the infective length. The hexadecimal pattern can be used to detect the presence of the virus by using the "search" routine of disk utility programs such as *The Norton Utilities*.

Virus Bulletin has not received permission to reproduce this article on CD from the author. Readers can obtain a paper copy of the original issue directly from *VB*.

SPECIAL FEATURE

Joe Hirst

Virus Bulletin has not received permission to reproduce this article on CD from the author. Readers can obtain a paper copy of the original issue directly from *VB*.

MANAGEMENT ISSUES

Chris Frost
Price Waterhouse

Implementing a Security Policy

As the number of computer viruses in circulation rises, companies are finding themselves increasingly at risk from the threat of virus attack. In this article, I will examine how the formulation of a security policy can help to tilt the odds against computer viruses.

The formulation and implementation of a corporate data security policy designed to protect data held on PCs is arguably the best method for reducing the risk posed by viruses.

The long term success of any data security policy is always dependent upon a high level of support and commitment from senior management. They must ensure that both the personnel and resources necessary to implement such a policy are made available. It is extremely unlikely that any policy will succeed without this support. Data held on computerised media should be viewed as a valuable resource which must be protected from the various threats to its integrity. A data security policy should minimise the risk of damage to data by addressing those areas at greatest risk first.

Backup and Recovery

Damage to data can prove catastrophic. It is therefore important that critical data is backed up on a regular basis. Procedures should be defined and adhered to. An individual must be responsible for ensuring that scheduled backups are actually performed. Special arrangements must be made for ensuring that backups are taken when staff are sick or on holiday. A written log of all backups should be maintained and these should be reviewed by management on a regular basis.

Incomplete or faulty backups are of no use to anyone. From time-to-time backup procedures should be reviewed and tested, as should recovery procedures. Spare machines are good workhorses for checking that backup/recovery procedures actually work. I know of one company which faithfully copied its PC-based sales ledger to tape every evening for well over two years. It was only when their hard-disk crashed that they discovered that the tape drive write head had been damaged, so that bit 4 had been set on for every byte that had been copied to tape. This rendered their backup copies of the sales ledger useless. They should have made sure that their backup/recovery procedures were working properly.

Floppy Disks

Virus programmers know that floppy disks are the best media for spreading their creations from one system to another. A lack of care over the use of floppy disks by PC users in different systems has been largely responsible for the spread of computer viruses.

A data security policy must, therefore, address the control of movement of floppy disks in and out of an organisation. I know of one software house which sent out customer support engineers with diskettes containing software upgrades for their telecommunications package. One engineer upgraded a customer PC infected with a virus. The virus attached itself to the program used to upgrade the telecomms package. The engineer returned to base with the virus infected program and received instructions to upgrade the communications package at another customer site. He proceeded to do so and inadvertently infected the second customer's PC with the virus.

Had technicians at the first customer site not subsequently discovered the virus in their PCs and informed the software house, the engineer would have infected other customer sites and even PCs at his own workplace - which he shared with other engineers. It is very likely that the software house's entire customer base would have been infected had prompt action not been taken.

This software house now supplies its engineers with notchless diskettes which cannot be written to by standard floppy disk drives and are thus permanently write-protected.

Neither the software house nor the customer sites described above had implemented a data security policy. They had no controls over the transfer of data or software to or from floppy disks. The software house had not considered the possibility that an engineer could pick up a computer virus from a customer installation. Neither customer had considered that engineers working on their systems could either pick up or deposit viruses.

Anti-Virus Software

Anti-virus software can be effective in both preventing and detecting virus attacks. However it should only be regarded as one element of a wider data security policy.

The views of the author are not necessarily those of Price Waterhouse.

VIRUS BULLETIN SUPPLEMENT

Virus Bulletin has not received permission to reproduce this article on CD from the author. Readers can obtain a paper copy of the original issue directly from *VB*.

VIRUS ANALYSIS

David Ferbrache

nVIR and its Clones

First detected in Europe in 1987, the nVIR virus has become widespread in the Macintosh world in a variety of forms. The most common strains are known as nVIR A and B. From the nVIR B strain a number of simple clones have been constructed and released, including Hpat (Autumn 1988), AIDS (March 1989 in Holland), MEV# (April 1989 in Belgium) and nFLU (August 1989 in Minnesota, USA).

The later three clones were constructed by replacing each occurrence of the nVIR string in virus binary with an alternative four letter resource name. This operation can be carried out easily through the use of binary editors making the risk of other clones considerable. In the case of the Hpat virus the alterations are more extensive and include the renumbering of the "CODE" resource in the infected application. The primary purpose of constructing such clones seems to have been to defeat virus detectors such as the public domain "disinfectant" utilities which rely on the detection of resources with a specific name. For example disinfectant exists in three versions:

Ver	**Can detect and remove**				
	nVIR A	nVIR B	Hpat	AIDS	MEV#
1.0	*	*	*	-	-
1.1	*	*	*	*	*
1.2	*	*	*	*	*

Overview

The nVIR virus infects by adding a new code resource to the resource fork of the file being infected. The jump table (in code resource 0) is then patched to point to the new viral code resource, which when executed will then transfer control to the original code resource via a saved copy of the original jump table entry *(See Figure 1).*

The nVIR virus infects a system in two phases:

1) The virus infects the system file when an infected application is executed (either from hard disk or floppy) by adding a new "INIT" initialisation resource.

2) The "INIT" segment is executed when the system is rebooted causing a copy of the virus to become resident in the system heap after which it can infect executed applications.

The virus will infect the majority of applications programs used after the system heap has become infected (including finder). It will not infect other files which have resource forks, nor will it infect the notepad, clipboard or scrapbook files. Although widespread, nVIR is far less infectious than INIT 29 or Anti.

The virus incorporates a counter which commences at 1000 when the system file is first infected and is decremented each time the system is booted (by 1) and each time an application is infected (by 2). This counter ensures that the virus has an incubation period (of variable length) during which few symptoms are apparent although it is actively spreading. When the counter reaches zero the virus will show a number of symptoms dependent on strain.

nVIR A will say "Don't panic" if Macintalk is installed; this occurs one in sixteen system boots. When an infected application is launched there is a one in eight chance of the "Don't panic" message and a one in 256 chance of the virus repeating the message twice. If Macintalk is not installed the virus will simply beep (once or twice).

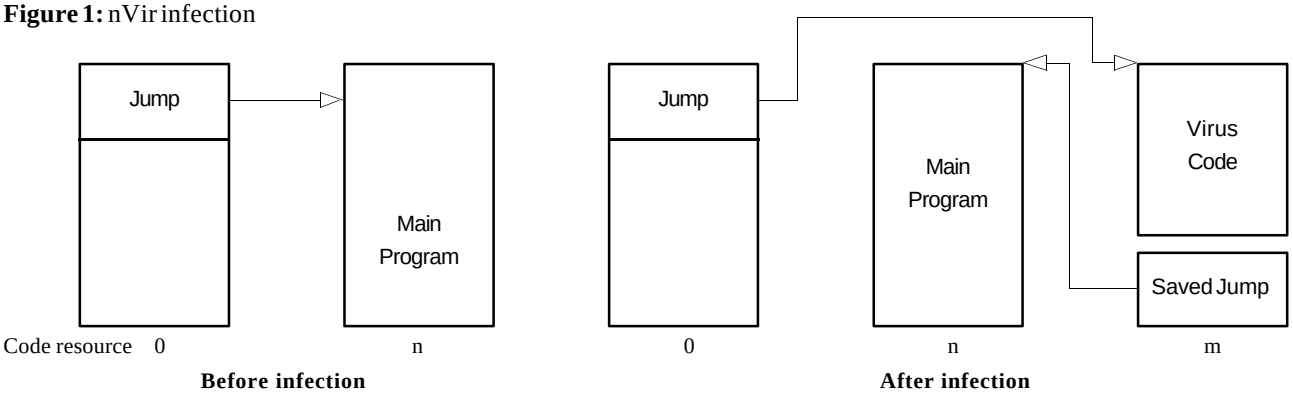
nVIR B (and clones) will beep one in eight system boots. When an infected application is launched there is a one in four chance of a single beep and a one in 64 chance of beeping twice. This strain does not call Macintalk.

The nVIR viruses will breed with other viruses such as INIT 29 to produce a cross-breed which most anti-virus tools will fail to disinfect.

Detailed Analysis

The nVIR B strain consists of six new resources added to an infected application, these are:

Figure 1: nVir infection



Resource name	Number	Size	Function
CODE	256	422	Virus main code
nVIR	1	428	Inserted as tail patch to Telnit system trap
nVIR	2	8	Original jump table entry
nVIR	3	416	Inserted as INIT 32 in infected system file
nVIR	6	66	Virus manipulation routine
nVIR	7	2106	Virus infection routine

Stage 1 Infect System File

CODE 256 in infected application executed

* Infected application is executed which (through the modified jump table) calls the CODE 256 resource.
* The code resource checks if an nVIR 10 resource is present in the system file
* If not present then nVIR 7 is called to infect the system file
* Transfer control to the original file code segment via the saved jump table entry in nVIR2

nVIR 7 in infected application

* Create (if not present) an nVIR 0 in system file (this is used as a counter starting at 1000, decremented on infection and boot, when 0 virus demonstrates beep symptoms)
* Decrement nVIR 0.
* If nVIR 0 is zero then call nVIR 6 damage routine
* Infect system file
  * If INIT 32 is present then
    * Compare nVIR 6 and 7 in system file with application file, if they vary then upgrade application resources from system file, else skip
    * else copy nVIR resources to system file (nVIR 6 as INIT 32)

Stage 2 - Infect system heap

INIT 32

* Call infection routine to install a tail patch into the Telnit system trap consisting of the nVIR 1 code resource

Stage 3 - Infect application

nVIR 1 Patch

* Call original routine for Telnit.
* If nVIR 10 is not present in system file
  * If nVIR 7 not in application file then
    * load and call nVIR 7
    * Update system counter.
    * If zero then call damage routine
    * If CODE 256 is present in application then compare with nVIR 6 and 7 resources in system file, if they differ or CODE 256 is absent then install virus resources
    * Patch jump table and store old entry in nVIR 2 resource

Use of Resource Editor to Detect and Remove Infection

Apple provide a useful utility in examining and disinfecting virus infections called resource editor (ResEdit). This utility allows the user to examine each of the resources which make up an application on the Mac (including menus, text strings, code blocks, windows, alerts etc). This editor allows you to look for known virus resources and to disinfect an application by removing these resources. **ResEdit is not a substitute for a viral disinfection utility and its use is definitely not recommended to the casual user.**

nVIR infections can be detected in a number of ways

* Alteration times on the infected system file and application code file changes
* Delay in application startup and unexpected disk activity
* Reduction in available memory
* Beeps on infection and reboot
* Use of ResEdit and virus disinfection utilities

To use ResEdit open the suspect application or system file, look for the presence of resource types nVIR, MEV#, nFLU, AIDS, and Hpat in the list of resources in the application.

To remove infection:

* Boot from a clean system file (on floppy)
* Copy a new system file from floppy to hard disk
* Reboot providing a clean system environment
* Execute a clean copy of ResEdit (from a floppy disk)
* Open the application
  * Open the CODE resource type
    * Open the CODE 0 resource, look for the string “0000 3F3C 0100 A9F0” in line 3. This jump table entry for the virus code must be replaced by the original jump table entry stored in nVIR 2.
  * Open the nVIR resource type
    * Open the nVIR 2 resource, transfer the 8 byte string to replace line 3 of the CODE 0 entry
  * Remove all nVIR resources
* Close the application resource fork

It is possible to remove nVIR infections from a system file by removing the INIT 32 resource from the file. In general this is not recommended as it is far easier to boot from a clean (write protected) system diskette and transfer the system file from diskette to disk.

Finally, it is worth noting that nVIR infections can be inhibited through the use of an nVIR 10 resource in the system file. As you will note from the algorithm such a resource will prevent spread of the virus. Unfortunately many simpler virus detectors will identify this inhibiter resource as an active nVIR infection.

Acknowledgements: *My thanks to Lawrence Brown and John Norstad for their work in analysing the nVIR virus strains.*

PRODUCT REVIEW

Phil Crewe
Fingerprint Graphics

Symantec Antivirus for Macintosh (SAM)

The Symantec Corporation has a good reputation for its programs on both Apple and IBM platforms. Symantec Utilities for Macintosh (SUM) is one of the most useful disk toolkits available, and both the More II outliner and the Think programming environments are extremely popular. The release of SAM, therefore, was greeted as a welcome addition especially by the larger corporations which maintain a large number of Macintosh-based workstations.

Macintosh Viruses

Protection against Macintosh viral penetration includes use of the varied shareware available like Vaccine, Disinfectant, Interferon and Gatekeeper. These are excellent packages and for a competent individual they will supply the protection required. I would encourage a single user or system with only two or three Macintoshes to continue to use these packages if they are happy with them. However, SAM is aimed at the medium to large business users which require constant protection from viral infection. This should remain invisible to the user unless a potential problem is uncovered, at which time it should stop the Macintosh with a large, clear error message. It should also be capable of 'learning' its environment, and therefore not flag potential problems which it has been told to ignore.

SAM - What You Get

SAM is designed to protect against viral infection continually. It can thus only be effective if it is running all the time on every machine. Since it comes in two parts (an INIT called SAM Intercept and an application called SAM Virus Clinic) the software is effectively running whenever a machine with the INIT installed is switched on. It works on all machines from a 512k upwards, and requires System 4.2 or higher. It is network (TOPS and AppleShare) compatible, and is also MultiFinder friendly. In the documentation it claims to understand the viruses nVIR, Scores, INIT29, HPAT and ANTI, and be able to repair them. Although not in the documentation, the SAM Intercept welcome screen also states understanding of AIDS and MEV#.

Installing SAM is easy. Copy the SAM Intercept file into the system folder on either hard disk or start-up floppy, and copy SAM Virus Clinic to a convenient applications area. Then restart the machine. A welcome screen is displayed when the Macintosh is powered-up after installation. The text tells you how to configure Intercept.

Configuring SAM Intercept

Opening up the Control Panel and selecting the SAM Intercept icon allows you to configure the operation of the INIT. There is constant help when you are doing this configuration, and it is all context sensitive, so merely clicking on the heading of the feature gives further information. Useful items for configuration are: the ability to scan the system folder or the whole startup volume when the Macintosh is started or shut down, the ability to automatically scan inserted floppies, and the ability to tune the level of required protection. This protection level tuning has four settings. Basic prevents launch of an infected application. Standard does Basic and also prevents changes to certain resources and watches for changes in startup documents. Advanced is the most comprehensive, doing Standard, watching for additional modifications and also checking for illegal activity such as bypassing the resource handler. Custom allows the user to switch all checks covered by Advanced globally on and off.

Any configuration chosen can be locked and password protected so that no-one else can modify SAM intercept. This password is stored in encrypted form. Also there is an exception list so that SAM learns what to ignore whilst you are working.

Testing Environment

Three different situations were chosen with the Macintoshes in constant use. The first was an SE with 2 MBytes RAM and a 40 MByte hard disk, used for copy generation, software testing, remote technical support, and remote access to an AppleTalk network for printing and file sharing. The user is highly technical. The second was two machines in a Technical Support section. Both were full specification Macintoshes, one a Mac II and the other a IIcx. Both run MultiFinder, constantly access a central database and run many different software packages both for evaluation and support uses. They are used by highly technical people. The third was three machines in a Desktop Publishing bureau. All were full specification, two being Mac IIs, and the third a Mac IIx. They run Finder only, all DTP packages, and constantly work on floppy disks supplied by customers. They are in use 24 hours per day by specialist but non-technical staff, and they are the most vulnerable point in terms of potential

virus risk.

SAM intercept was configured on all machines to different levels of protection. The SE and Technical Support used Advanced mode, whereas the bureau used Standard mode. They were all set to scan floppies automatically when inserted, and to scan the system folder when started and shutdown.

SAM Intercept in Use

Inserting a floppy in a protected machine generates an automatic scan, and a dialogue showing you the progress. Scanning the SAM floppy itself takes 10 seconds, and then you get the message 'no viruses found during this scan' at which point the normal desktop appears. If you do a scan whilst in an application then the scan proceeds exactly the same, returning to the application when completed. The only problem I found with this was that on some applications the screen didn't re-draw properly after the scan dialogue was removed. However this is a fault of the application rather than SAM.

Attempting to scan a totally unreadable disk gives the standard 'do you wish to initialise' dialogue without triggering SAM. When given a disk with a fault (ie one file is unreadable) SAM completes the scan but on the dialogue there is the message (for example) '5 of 6 executable files scanned'. I would have preferred a better error message than this, but at least it gives a clue that something is wrong. Note that it counts system files, desktop and the like as executables. It doesn't scan documents.

Scanning a disk infected with nVIR B brought up a Grim Reaper dialogue box (nice touch!) and correctly informed me that the disk contained nVIR, and to check it with SAM Virus Clinic. Intercept did NOT prevent the disk mounting on the desktop. Attempt to run the infected program, and again up pops the Grim Reaper telling me that nVIR has been detected.

Attempt to move the SAM Intercept file from the System folder, and up comes a flashing dialogue warning of an attempt to modify SAM Intercept. The options presented are Allow, Deny, and Learn. Clicking Learn will allow the process unhindered and will not prompt if the same thing happens again at a later stage. It is at this point that I found a potential problem with this procedure. Once an action is marked as Learned, it is supposedly entered into a table for possible modification later. However this table isn't updated until the machine is shut down, and is held in memory up to that point. If the Macintosh bombs in the interim, and the machine is not shut down normally, then you have lost all the learned information since the last correct shutdown. There is no alternative but to re-teach Intercept. This is an oversight, as Macintoshes do have a tendency to bomb when pushed hard in some circumstances, and you shouldn't need to go through a restart procedure every time you teach Intercept that something is allowable, simply to write the learning table to disk!

Problems in a Normal Environment

There were a few clashes between SAM Intercept and other INITs. The most serious of these was with StartupScreen. This allows a different screen during power-up instead of the normal 'Welcome to Macintosh'. Unfortunately it prevents feedback during the power-up scan of the Macintosh. It proved fatal when power-up started modifying QuickTimer INIT (part of QuicKeys) which SAM then dutifully flagged. The screen prevented the dialogue box showing, and it appeared as if the Macintosh had hung. By hitting return (which by default allows the progress of the modification, but doesn't teach SAM to cope with it next time) the power-up procedure recommenced.

Another clash was with 'LUC AIS'. This is an INIT which is part of the Al-Nashir Arabic RSG4! package. The INIT had to be removed, and if Arabic is required then the system has to be booted with SAM suppressed. The final clash was with Virtual INIT (for virtual memory on PMMU-equipped Mac IIs). It was necessary to load Virtual before SAM Intercept which necessitated file renaming.

SAM Virus Clinic

This application is run after SAM Intercept has located a problem. Virus Clinic gives exact details of the virus located and in which application it has been found. You can then Disinfect. I noticed that when Intercept found one virus on a disk, it didn't look any further. Thus a disk containing ANTI, nVIR and Hpat showed up as only ANTI within Intercept.

Virus Clinic provides the option to remove the virus. I would caution against this. If you have the original program on its distribution disk, then restore it from there. Only attempt to remove the virus if you have no alternative, and for some time later treat the restored application with caution. It may not be infected, but it could be corrupted.

I generally found no problem running Virus Clinic. I ran it on TOPS, AppleShare and also on a 386 (as a Novell server running Netware for Macintosh!). It seemed well-behaved, causing no more problems with MultiFinder

than to be expected, and the user interface is easy to grasp. Drives can be selectively scanned and individual files and folders can be checked and/or repaired.

Virus Clinic can also be extensively configured. It can check for 'irregularities' which gives the application some chance of locating a future virus. This does, however, cause false alarms. A technical user ought to interpret these reports. Also Virus Clinic can 'fingerprint' the code resource of a file using checksums which are compared against the originals, and the user is warned of alterations.

Operation Against Virus Attacks and Infections

A test bed selection of viruses shown in Table 1 was used to check out Intercept and Virus Clinic for the ability to detect, disinfect (ie remove viral code) and repair (ie return the application to its pre-infection state) infected items. SAM was compared against Disinfectant (1.0, 1.1, 1.2), Virus Detective (2.3, 3.0), Interferon (3.1) and VirusRx (1.4a2).

Table 2 (below) summarises the results. F indicates a successfully located virus, D a disinfected application, and R a repaired application. The sample codes refer to the virus types indicated in the previous listing.

SAM appears to fail in testing in four main areas.

1. It detects nVIR 10 inhibitor resources (as placed by inoculation software) as an nVIR virus. This is an oversight which will have more serious consequences in a non-technical environment than in a technical one, where awareness of the inoculation routine will probably be higher.

2. It fails to detect simple binary edited versions of nVIR and Scores. The changes were trivial ones which could have been made by an unskilled programmer. It should be said though that in Standard mode and above, SAM Intercept will detect the operation of the virus, even if it doesn't recognise the virus itself. No protection is however offered in Basic mode. For this reason I would advise not running Intercept in Basic mode, only in either Standard or Advanced. Unfortunately it comes with Basic as a default.

3. It detects but does not attempt disinfection of multiple virus infection.

Table 1

A Scores (infected Scores file, System file, Scrapbook file, Note Pad file, Desktop and application).
B nVIR A (infected System file and application)
C nVIR B (infected System file and application)
D Hpat (infected System file and application)
E AIDS (infected System file and application)
F MEV# (infected System file and application)
G nFLU (infected System file and application)
H INIT 29 (infected System file and application)
I Anti (infected application)
J INIT 29/nVIR B cross infection (infected System file)
K Peace 1832 (infected System file)
L Peace 1908 (infected System file)
M renamed nVIR B resource (infected application)
N renamed Scores resource (infected application)

Note that Peace 1908 was not available locally for testing, but a dummy virus with similar characteristics was fabricated.

Table 2

Virus code:	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Disinfectant 1.0	R	R	R	R	-	-	-	R	R	R	-	R	-	-
Disinfectant 1.1	R	R	R	R	R	R	-	R	R	R	-	R	-	-
Disinfectant 1.2	R	R	R	R	R	R	R	R	R	R	-	R	-	-
Virus Detective 2.3	D	D	D	D	D	-	-	D	-	D	D	D	D	-
Virus Detective 3.0	D	D	D	D	D	D	D	D	D	D	D	D	D	-
Virus Detective 3.1	D	D	D	D	D	D	D	D	D	D	D	D	D	D
SAM 1.0	R	R	R	R	R	R	-	R	F	F	-	R	-	-
Interferon 3.1	D	D	D	D	D	D	D	-	-	F	-	-	D	D
VirusRx 1.4a2	F	F	F	F	F	F	F	F	-	F	-	-	F	F

4. It fails to disinfect Anti infected files correctly.

The last two points may or may not be important. Provided that a safe backup copy of all applications and system files are kept (the original locked floppies, for example) then the situation should never arise where a disinfection is necessary. However this should be borne in mind if the situation is different from the ideal.

SAM does not attempt to remove viruses from the system heap. Thus disinfection still leaves the virus in memory, which if SAM Intercept is not present will attempt re-infection. Disinfectant detects memory resident viral code, and allows the user to reboot to the now clean system file after the disinfection run. Users should reboot immediately after disinfection to prevent a virus re-attack.

Basic Scores is trapped in Advanced or Standard mode, and no problems were encountered in blocking the infection. Re-engineered Scores will not be detected by Intercept in any mode except for the alteration of the CODE 0 jump table, which will always be detected. Basic nVIR is trapped in Advanced or Standard mode, and also the change to the system file nVIR 0 counter is detected during subsequent infections. Re- engineered nVIR is partially trapped, the alteration of the CODE 0 jump table and installation of CODE 256 resources to applications are detected, as well as the INIT 32 code in the system file. However auxiliary nVIR resources are not detected if they are renamed. Infection by Anti is recognised and prevented, but disinfection is flawed.

INIT 29 in Particular

This failure to remove memory resident code causes problems when encountering INIT 29 (a particularly virulent virus). After disinfection of INIT 29, quitting SAM virus clinic immediately caused the memory resident copy of the virus to attempt re-infection of Finder. Although SAM Intercept prevents incursion by the virus onto a clean system (assuming floppy scanning is done automatically), installation of SAM onto an already infected Macintosh will not directly combat the virus.

After installation, SAM Intercept correctly detects the virus on startup. However even if the DENY button is selected, the Intercept file is still infected by INIT 29. It does this via the INIT 31 mechanism.

The only way to combat this virus is to boot from a known clean system disk and attack from there. One addition in this regard is a REBOOT button on the Grim Reaper dialogue so that a clean reboot can be performed. Additionally the Proceed button on the infection detected dialogue box should not be aligned with the Learn button on the resource modification alert box. An inadvertent double-click causes SAM Intercept to Learn about virus infection, which is, on balance, not a good idea!

Documentation

Overall I found this very good. Everything is included, except for the search algorithms, which possibly some people would like to see. All the reports and message types appear, so if a problem occurs a solution can be devised. There is an appendix on different virus types, which is invaluable for a systems manager just starting to implement threat procedures. The manual is also illustrated with screen views which I found very helpful. I can foresee problems for someone installing it with only rudimentary knowledge of the Macintosh interface, but once again I must reiterate that I don't think this is the package for that sort of person anyway.

Conclusions

My own feeling is that SAM is the best all round commercial package for a Corporate environment. In a production and bureau environment it received unqualified praise. The automatic scanning of floppies proved time-saving over manual scanning using a package like Disinfectant, and the scanning seemed to be effective. In fact it already has one notch in its bow - during testing it successfully picked up nVIR on a floppy disk delivered to us direct from the manufacturer.

If you are a system manager in a large systems environment then SAM is probably for you. However it is definitely NOT for the non-technical user. Although it is easy to install and run, tuning your particular setup, and ensuring that you know what SAM is doing means you have to spend considerable time learning it. Not difficult, but if you already have a system and software to deal with the potential problem, then stick with it.

Acknowledgement: *I should like to thank David Ferbrache for his technical help with this review.*

SAM is available from The Symantec Corp., Product Support Dept., 10201 Torre Ave., Cupertino, CA 95014, USA. Tel 408 253 2167.

Symantec UK, NKA House, 36 King St., Maidenhead SL6 1ES. Tel 0628 776343.

BOOK REVIEW

Jim Bates

Computer Viruses- A High Tech Disease - Abacus, 282pp

Author: Ralf Burger

This is a translation of a German publication and it would appear that no effort has been made to correct the various cross references within the text. This makes the book extremely difficult to use when trying to follow the references to other sections.

The book devotes a lot of its space to self-justification, self-praise and denigration of similar works by other people. The first page for example, refers to the German Chaos Computer Club (a loose collection of hackers and enthusiasts who meet to swap information about virus production and security penetration techniques) as "known for discovering security breaches in data processing systems". The general approach is a little childish and it tries to adopt a pseudo-scientific approach to viruses and Trojans. Frequent attempts are made to justify the production and distribution of such programs and techniques on the grounds of "discovering security breaches" in major computer installations. Most of the book's observations concern computers operating under MS-DOS but there are occasional references to other systems.

The inclusion of some printed listings of actual virus source code and hex dumps created something of a furore when this book was first published but the actual examples given are quite innocuous and often a little silly. Source listings are in a variety of languages available under MS-DOS but attempts to enter and run these programs are almost invariably thwarted by errors either in coding or printing. Such programs that do operate are crude and ineffective. In several instances of a so-called "overwriting virus", virus code is written indiscriminately over existing .COM files, destroying their original function in the process. Since the first .COM file on an MS-DOS system will almost always be COMMAND.COM, this means that "infection" by such a virus immediately renders the machine incapable of booting or handling the simplest of DOS commands. It need hardly be noted that a virus that destroys its host will not survive for very long nor spread very far. Another example of a "virus" consists in replacing program files with a batch file of the same name. Quite what this is supposed to achieve is not made clear.

The only recognisable listing is of the Vienna Virus, which is given in Assembler source form. If the code is entered as printed, the program will not run. However, it is possible to modify the code so that the program WILL operate. This immediately provokes questions concerning the wisdom of publishing such listings in this way and thereby allowing any irresponsible individual to produce his own virus code without thought for the consequences. Drawing a parallel, I wonder what the general reaction would have been to a book listing details of how to make explosives or assemble bombs, however amateurish.

The non-technical sections of the book do little more than restate the obvious ideas behind viruses. They relate little, if any, useful information to the reader. The technical sections display a distinct lack of general expertise and there is only passing reference to the serious potential problems posed by boot sector viruses. There is also no reference to the ANSI type of Trojan which can cause such mischief on unprotected bulletin board systems. In fact Burger says, "As long as documents or programs are only written into a BBS, there is no danger of infection". Quite a lot of space is devoted to the supposed potential for developing beneficial virus techniques, and there is even a page which quite seriously discusses the odds of a random "virus mutation" being beneficial (to the virus of course).

This book seems to be an obvious and cynical attempt to cash-in on the current fear and uncertainty about viruses; the included virus source code listings (excepting perhaps the section on the Vienna Virus) contribute little of worth to an aspiring virus writer and even less to genuine anti-virus research.

There is no doubt that some damage will result from attempted copies of the Vienna virus listing; and the pseudo-official status afforded to hackers and software terrorists may tempt some readers to try such activities. This is to be deplored and my own conclusion is that books of this type can only bring the genuine computer enthusiast into disrepute.

Available from: Abacus, 5370 52nd Street SE, Grand Rapids, MI 49508, USA.

EVENTS

Compsec '89 in conjunction with the EDP Auditors Annual Conference includes a three hour special presentation on computer viruses. The event takes place at the QE II Centre, London, from 11-13 October, 1989. Details from Penny Moon, Elsevier Seminars, UK, Tel 0865 512242.

CW Communications are holding a one day conference entitled **Computer Viruses: Combat and Cure.** The event takes place on 26 October 1989 in London. Details from Clare Peiser, Quadrilect, UK. Tel 01 242 4141.

The Fifteenth Annual **Computer Security Conference of the Computer Security Institute** takes place in Atlanta, Georgia, USA from 13-16 November. Tel 508 393 2600

S&S Consulting Group is holding a one-day 'strategic' **seminar on the Virus Threat**. It takes place on 16 November 1989 at Rickmansworth, Herts, UK. Details from S&S Enterprises, Tel 0494 791900.

Sophos Ltd continue a series of **Virus Workshops**. The next available workshops are on 21 November 1989 and January 25/26 and will be held in Oxford and London respectively. Both technical and general management streams are available. Details from Karen Richardson at Sophos, UK, Tel 0844 292392.

The **Annual Brief on Secure Systems**. This update on computer security developments worldwide takes place on 28-30 November, 1989 at the Hague, The Netherlands. Tel +31 3403 79597.

Corporate Computer Security '90, 13-15 February 1990, Novotel, London, UK. Details from PLF Ltd, UK. Tel 0733 558571.

IFIP/SEC '90. The sixth international conference and exhibition on information security, Espoo, Finland, 23-25 May 1990. For details contact Congrex, Finland, Tel +358 0 175355, or Jugani Saari, Finland, Tel +358 0 177901.

SECURICOM '90. Computer and communications security conference, La Defense, Paris, France. Details from SEDEP, 8 rue de la Michodiere, 75002 Paris France, Tel +33 1 4742 4100.

VIRUS BULLETIN

Subscription price for 1 year (12 issues) including delivery:

US$ for USA (first class airmail) $350, Rest of the World (first class airmail) £195

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, Haddenham, Aylesbury, HP17 8JD, England

Tel (0844) 290396, International Tel (+44) 844 290396
Fax (0844) 291409, International Fax (+44) 844 291409

US subscriptions only:

June Jordan, Virus Bulletin, PO Box 875, 454 Main Street, Ridgefield, CT 06877

Tel 203 431 8720, Fax 203 431 8165