COMPARATIVE REVIEW

Making an Entrance: SuSE Linux

Matt Ham

As *Virus Bulletin*'s first *Linux* comparative, this review was embarked upon with some trepidation. In general, *VB* comparative tests have become easier to carry out as time has progressed and the obscure foibles of the various products have made themselves known. Without the benefit of this background knowledge, it was anyone's guess as to what the products in this test would present by way of pitfalls.

In addition, there is an array of testing tools available to ease the process of comparative tests, as well as various scripts and utilities, all of which are *Windows*-based and of no use in a *Linux* test.

With such a show of anxiety at the start, I shall break with tradition and state that all products proved testable for on-demand detection, though the methods used to produce these results differed slightly from those usually employed in *VB* comparative tests.

Of the eleven products submitted for testing, only three offered on-access scanning located entirely upon the *Linux* server, and the results from the testing of these modules were less than impressive. This being the case, the results of on-access scanning tests are bundled together after the main body of the review.

Several of the product lines that are regular contenders in *VB*'s comparatives are absent from this review – either due to their being at beta stage or because this platform is not supported by their manufacturers. Furthermore, a sizeable proportion of the products reviewed are scheduled for major upgrades in the near future – the *Linux* anti-virus market is still young and subject to change.

Test Sets

The test sets for this comparative review were based upon the standard *Virus Bulletin* comparative test sets. The In the Wild (ItW) set was aligned to the *WildList Organization*'s February 2002 WildList.

In addition to the usual contents of the test sets a number of *Linux* worms and viruses were added. These fall into two categories: worms transferred as archives after an initial exploit has given local access rights and ELF file infecting viruses. As yet, the number of these is not great, but more files will be added with future test set updates.

Other additions to the test sets included two viruses in the polymorphic test set, W32/CTX and W32/Fosforo. Again, the polymorphic test sets can be expected to have several further additions in the near future.

Of the additions to the ItW test set one is more noteworthy than its impact in the real world might have suggested. W32/Heidia.A is a .ZIP file infector which relies upon manual running to insert itself into existing .ZIP archives. The main code for this process was the file included in the ItW set – though a pair of infected .ZIP files were added to the standard set. The addition of archives such as these will instantly strike a detection rate rift between those scanners which look inside archives by default and those which do not.

This difference will be made all the more apparent by the presence of the *Linux* worms. *Linux* worms are commonly transferred as archives of files – and, clearly, these will not be scanned (at all) on-demand by products which do not consider archives worth scanning.

In the cases of Lion.A, Ramen and Adore, the files placed into the test set consisted of the contents of the archive as well as the archive itself. This left Lion.B and Lion.C which were represented only by their archived form. Another likely problem file is Cheese, which is UUE encoded.

For speed testing the standard clean sets were used – though with another *Linux*-specific addition. In order to test the rate of scanning for native *Linux* files the contents of /bin, /opt and /sbin were selected as a further test set. Since these files may be subject to replacements or additions when software is installed, a copy was made of these two directories. Testing was performed on this copy so as to ensure that each product was scanning an identical test set.

Test Procedure

All test sets were stored in RAR archives or compressed machine images and restored between tests.

On-demand tests were performed locally, with the bulk of the test sets being scanned while located on FAT partition. The exception to this was the *Linux*-specific malware which was scanned while located in a directory in the root of the *Linux* installation. This was so that the files would be scanned on their native partition format.

A cursory inspection of the first few products to arrive suggested that the most reliable method of detection in this test would be to standardize on detection by deletion of infected files. Primarily, this was because this is a far quicker process than sifting through the results generated by programs which do not support logging except by redirection of STDOUT to a file. Deletion proved to be a good solution (except in those cases noted in the individual product comments).

On-demand tests	ItW File		Macro		Polymorphic		Standard		Linux	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%	Number missed	%
CA Vet Rescue	0	100.00%	0	100.00%	460	97.00%	1	99.94%	64	51.31%
Command AntiVirus	0	100.00%	0	100.00%	486	93.01%	4	99.79%	78	52.20%
DialogueScience DrWeb	0	100.00%	0	100.00%	399	99.14%	1	99.98%	35	81.91%
Eset NOD32	0	100.00%	0	100.00%	454	97.75%	0	100.00%	77	39.81%
FRISK F-Prot	1	99.92%	0	100.00%	399	99.14%	1	99.98%	35	81.91%
GeCAD RAV	1	99.92%	0	100.00%	411	96.79%	19	99.22%	42	68.43%
Kaspersky KAV	0	100.00%	0	100.00%	399	99.08%	0	100.00%	10	92.41%
NAI VirusScan	0	100.00%	0	100.00%	413	98.78%	2	99.87%	24	77.80%
Norman Virus Control	3	99.70%	18	99.68%	473	93.76%	13	99.49%	19	84.72%
Sophos SWEEP	0	100.00%	5	99.87%	476	93.31%	18	99.43%	54	58.90%
VirusBuster VirusBuster	1	99.95%	0	100.00%	493	91.01%	14	99.55%	7	90.00%

In order to test on-access scanning, the *Linux* server was connected by *SAMBA* to a *Windows 2000 Professional* workstation. From here, the standard *VB* test tools were used to move recursively through the test set, opening each file in turn so as to trigger on-access scanners.

Finally, the matter of testing the speed of scanning was addressed. Again, the standard *VB* clean sets were selected for scanning on a *Windows* partition situated locally, while a *Linux* test set was constructed – consisting, in this preliminary incarnation, of the contents of the /sbin, /bin and /opt directory trees of the test *Linux* machine. Since several of the products install within the /opt tree, these files were copied into a dedicated test directory rather than being scanned *in situ*.

Computer Associates Vet Rescue 10.5.0.0

ItW	100.00%	Macro	100.00%
Polymorphic	97.00%	Standard	99.94%

CA Vet Rescue displayed several odd quirks, not all of which were unique to this product, but since it is first alphabetically this seems an appropriate place to discuss these oddities.

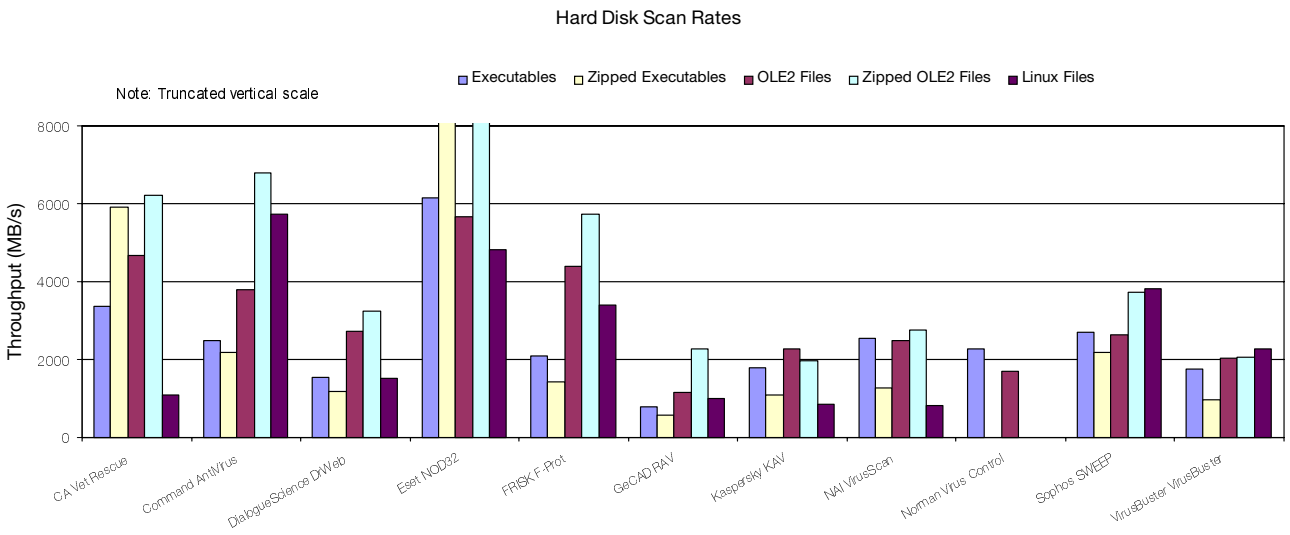
The most commonly encountered problem was that of accepted command line arguments. Using the –? argument for help produces a brief list of arguments followed by a more detailed description of what each of these does. This is all well and good, except that in many of the products, *Vet* included, the two lists of acceptable arguments do not tally. In other products the two lists tally, yet do not agree with the usable options – an even more confusing situation.

Vet Rescue hints as to the source of these unhelpful proceedings, since it announces itself as 'rescue.exe'. This leads to the conclusion that the command line argument handling code and other associated routines have been considered as being machine-portable from the DOS command line scanner. This may be true from a purely code-based point of view, but it would have been preferable for the text to have been taken into consideration when this portability issue was decided upon.

The lack of available command line options in *Vet Rescue* is quite marked, but certainly not unique to *Vet*. *Vet* was unusual, however, in requiring the target directory to be included before any options in the command line – which is opposite to the *de facto* standard. The lack of functionality may well explain *Vet Rescue*'s impressive scanning speeds.

A full tally of ItW and macro detections bode well for *Vet*'s fortunes, but results slipped slightly away from perfection on the polymorphic and standard sets, while the *Linux* set saw a detection rate of only a little over 50 per cent. However, *Vet*'s *Linux* detection rate proved to be not far below the average detection rate managed by products in this set.

It should be noted in relation to the *Linux* sample detection rates that, with such a small sample set as that used for these tests, there is great scope for errors in estimating the detection ability of a product. Until the number of samples in the set has increased significantly, no great messages



should be inferred from these figures – which are provided here for interest.

Command Software AntiVirus 4.64.0

ItW	100.00%	Macro	100.00%
Polymorphic	93.01%	Standard	99.79%

Following in the footsteps of *Vet*, *Command AntiVirus* demonstrated some odd behaviour. In this case it was a point blank refusal to delete any file which potentially could contain useful data – notably archives and OLE files. Since quarantining of these files was not permitted either, another method of deletion was selected.

The product was permitted to disinfect the samples which it refused to delete, and those files with changed checksums were deleted as having been declared dirty. As a sanity check the checksumming was performed without disinfection – to guard against the remote possibility that the scanner would alter checksums in some arcane manner. The scan with no disinfection showed no change in checksum – as would be hoped.

After obtaining results in this way the detection rates were certainly not disappointing at first glance although, admittedly, the *Linux* samples were discovered with only 50 per cent regularity and there were a number of misses in the polymorphic test set.

In the polymorphic set, the newly-added Win32/Fosforo samples caused problems – and the slightly older W32/Zmist.D samples evaded detection completely. However, a more concerning set of missed files was hidden behind the façade of full detection In the Wild.

The newly In the Wild virus W32/CTX is represented by ten samples in the ItW test set and, by reason of its polymorphic nature, is represented in the polymorphic sample set too, with 84 further samples. All ItW samples were detected, but 15 of the samples in the polymorphic set evaded detection. Such imperfect detection is not uncommon with complex polymorphics – but is undesirable nevertheless.

DialogueScience DrWeb 4.27a

ItW	100.00%	Macro	100.00%
Polymorphic	99.14%	Standard	99.98%

DrWeb registered its standard tally of suspicious files during the tests on the clean set, though there were surprises in store elsewhere.

I will admit that the detection of a virus In the Wild (in this case W97M/Pecas.B) using heuristics is not shocking. *DrWeb*, however, has a good record of identifying infected files accurately and exactly, so it was mildly surprising that on this occasion it detected only heuristically.

Other than this unexpected change, detection rates were good, though lowered by the influx of *Linux* and polymorphic viruses, which have added a significant new challenge to the companies submitting to this comparative. However, *DrWeb* was less affected by the new samples than many of the other products on test.

As with some of the other products there was no obvious method of determining a version number for the product, other than using the version number provided as the name and description of the installation RPM. There was also a slight difficulty in persuading *DrWeb* to delete what it considered to be archive files – though here these were only *PowerPoint* and some VBS files.

Eset NOD32 1.990

ItW	100.00%	Macro	100.00%
Polymorphic	97.75%	Standard	100.00%

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables		Zipped OLE Files		Linux Files		
	Time (s)	Throughput (MB/s)	FPs [susp]	Time(s)	Throughput (MB/s)	FPs [susp]	Time (s)	Throughput (MB/s)	Time(s)	Throughput (MB/s)	Time(s)	Throughput (MB/s)	FPs [susp]
CA Vet Rescue	162.0	3376.1		17.0	4666.7		27.0	5904.3	12.0	6217.3	169.0	1082.5	
Command AntiVirus	221.0	2474.8		21.0	3777.8		73.0	2183.8	11.0	6782.5	32.0	5717.0	
DialogueScience DrWeb	354.0	1545.0	[16]	29.0	2735.6		136.0	1172.2	23.0	3243.8	120.0	1524.5	
Eset NOD32	89.0	6145.3		14.0	5666.7		17.0	32172.5	3.0	26444.6	38.0	4814.3	
FRISK F-Prot	263.0	2079.6		18.0	4407.4		111.0	1436.2	13.0	5739.0	54.0	3387.9	
GeCAD RAV	690.0	792.7	[1]	69.0	1149.8		277.0	575.5	33.0	2260.8	181.0	1010.7	
Kaspersky KAV	307.0	1781.5	[18]	35.0	2266.7		147.0	1084.5	38.0	1963.4	219.0	835.4	
NAI VirusScan	216.0	2532.1		32.0	2479.2		124.0	1285.6	27.0	2763.2	224.0	816.7	
Norman Virus Control	241.0	2269.4		47.0	1688.0	[77]	n/a	n/a	n/a	n/a	n/a	n/a	
Sophos SWEEP	202.0	2707.6		30.0	2644.5		73.0	2183.8	20.0	3730.4	48.0	3811.3	
VirusBuster VirusBuster	310.0	1764.3		39.0	2034.2	[1]	166.0	960.3	36.0	2072.4	80.0	2286.8	

Once again, *NOD32* was significantly speedier than any of the other products on test, and it maintained its excellent detection rate on the old favourites in the *VB* test sets.

However, there proved a good deal more to challenge *NOD32* than usual, partially on account of the additions in the *Linux* test set. Scoring the lowest percentage of any scanner when faced by ELF format viruses, there is room for improvement for *Eset* here. Similarly, a number of the newly-added W32/Fosforo samples were missed by *NOD32*, resulting in the largest number of misses for *Eset*'s product for many comparatives.

Frisk F-Prot Antivirus 3.11

ItW	99.92%	Macro	100.00%
Polymorphic	99.14%	Standard	99.98%

The *F-Prot* product suffers from similar command line argument oddities to its close relative *Command AntiVirus*. At least in this case the problem is noted in the documentation. The documentation is also quite clear in stating that this is a product which is still under development, with several possible new avenues opening up to it in the near future.

As befits a product using the same engine, the results of the detection tests for *Command AntiVirus* and *Frisk F-Prot* were very similar. This similarity went as far as identical results in all but the ItW test set. Here, *Frisk* missed the .EML-extensioned sample of W32/Nimda.A – presumably .EML format files are excluded from scanning in order to reduce scan time.

This raises an intriguing problem as far as scanning from or upon a *Linux* machine is concerned. Many products still employ extension lists as a first filter when determining which files are to be scanned. It is not uncommon for scanners to check for executable content disguised by extension, but this is by no means universal.

On a *Linux* machine, however, extensions are essentially meaningless in many cases, and are more likely to be descriptive than any guide as to whether the file in question is an executable.

This is not so much a problem for products which can perform intelligent file-typing – but it may be irritating to developers who have traditionally relied upon extensions as an easy way of avoiding processor usage.

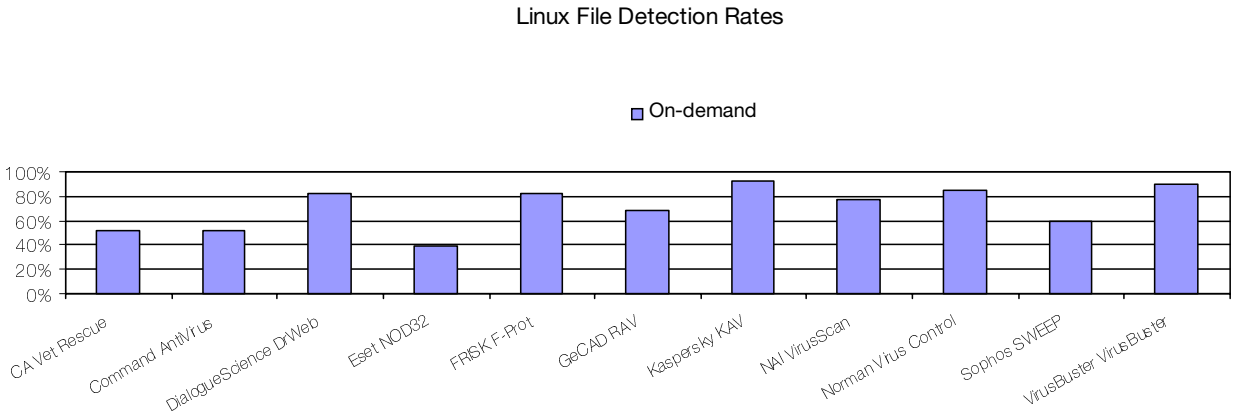
GeCAD RAV AntiVirus 8.5

ItW	99.92%	Macro	100.00%
Polymorphic	96.79%	Standard	99.22%

RAV was the only product in the test to boast a graphical interface – though the command line version was used for testing.

As befits a product which has seen more development than most, the command line options within *RAV* were numerous and bore more resemblance to the feature set usually seen on a DOS scanner.

Since the *RAV* scanner has been the subject of a recent standalone review, discussion of features here can be skimmed past speedily. However, this feature set did not protect against accidents, and after creditable detection rates in most categories, a miss due to the .EML version of W32/Nimda.A In the Wild will, no doubt, be galling for *GeCAD*.



Kaspersky AntiVirus 4.0.0.1

ItW	100.00%	Macro	100.00%
Polymorphic	99.08%	Standard	100.00%

One of the important differences to keep in mind when returning to a Unix-based operating system from *Windows* is the need for correct capitalization, a feature which led to some problems with the *Kaspersky* product.

Irritatingly, the archives provided for updating the product were all fully capitalized, whereas the program expects file naming in lower case lettering. This led to the somewhat tedious need to rename all of the definition files supplied, of which there were a large number, each dedicated to a certain type of threat.

In a rather idiosyncratic display, *KAV* defaulted to disinfecting files within archives on several occasions – despite being explicitly configured to perform deletions.

When this had been worked around, however, *KAV*'s performance was very much a return to form after some unlucky outings in recent *VB* comparative reviews. Certainly at the top of the detection range as far as the *Linux* files were concerned, *KAV* showed good detection all round.

When scanning the clean test set there was a moment of interest, as several possible false alarms appeared where none have been seen recently. The question was raised as to whether these should be classified as false alarms or merely as suspicious files. The announcement of some feeble joke program as 'VIRUS-noseless-dog-joke' has been a constant irritation to testers and end users alike – and in this case *KAV*'s alerts proved to be false alarms triggered by the detection of some form of greetings card.

Gratifyingly, however, the messages produced were as clear as might be hoped in the circumstances – declaring that what had been found was 'not-a-virus;GreetingCard.SLR'. With such a label, what had initially been considered a possible false alarm was speedily downgraded to merely a suspicious file.

NAI VirusScan 4.16 4188

ItW	100.00%	Macro	100.00%
Polymorphic	98.78%	Standard	99.87%

It was, perhaps, a little surprising that *NAI*'s product did not arrive as a fire-and-forget RPM package, but in the more humble guise of a gzipped tarball. Far from being typical from a company which has in the recent past indulged in the home user feature race with large competitors, this return to simplicity was reminiscent of the earlier days of *NAI*'s ancestral companies.

An irritating if not fatal niggle was that the default settings were not listed when command line switches were displayed, which left a large number of possibly irrelevant selections being used to avoid unwanted disinfection and the like.

Similarly, as noted for other products, the treatment of documents as archives makes it difficult to delete these directly. The fact that this proved to be a constant problem in this *Linux* comparative, while not having been an issue when dealing with any other platform, does seem odd.

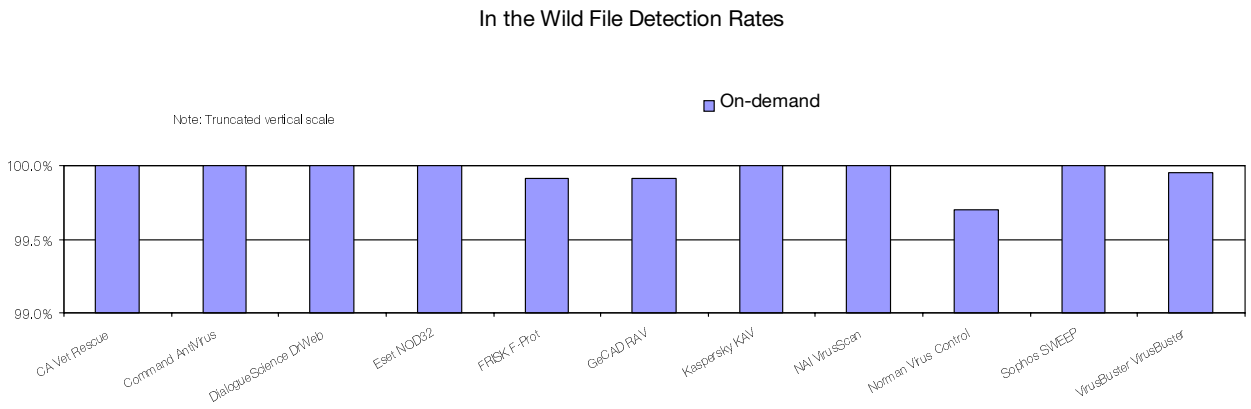
As far as misses in the detection tests are concerned, *VirusScan* was another product where a streak of bad luck seems finally to have come to an end. Detection was certainly at a better level than has been the case lately – and only in the *Linux* set can any weaknesses be identified.

After the good news there remains one fly in the anti-viral ointment, this being *VirusScan*'s speed of scanning. This was in the slower half of the field where the age-old *VB* clean sets were concerned, and the slowest of all those tested when *Linux* clean files were scanned.

Norman Virus Control 5.3-1

ItW	99.70%	Macro	99.68%
Polymorphic	93.76%	Standard	99.49%

Norman's product scored highly where the provision of reports was concerned – which is odd indeed, since this is



not a feature that is supported directly in those versions tested on other platforms. Especially appreciated was the list of clean files – this may be of somewhat limited use to the end user, but is excellent for a reviewer.

The version of *NVC* supplied seemed to encounter numerous difficulties when faced with the *VB* clean test sets. On non-archived Win32 and OLE2 files all was well, but on scanning the .ZIP test sets and the *Linux* test set, which includes some archives, the program ground to a halt – not before producing some cryptic error messages and a few random characters on the screen.

Equally disturbing was the program's behaviour when scanning the OLE2 files. The increase to 77 suspicious files detected in this test set must be indicative of an error somewhere.

Again, when pure detection was inspected *NVC* demonstrated some unexpected behaviour. This manifested itself in the missing of files ItW which have been detected by *NVC* on other platforms since time immemorial.

Whether the problems encountered here are specific to the flavour of *Linux* on test, or they are more general in nature, it can only be hoped that they will be banished in short order.

Sophos SWEEP 3.55

ItW	100.00%	Macro	99.87%
Polymorphic	93.31%	Standard	99.43%

Like roughly half the packages submitted, *SWEEP* arrived as an archive rather than as an RPM package – though an installation shell script was supplied to ease matters. The script requires that a *SWEEP* user and group are set up, though it seems that these are not used unless the machine is to become an *InterCheck* server.

In terms of detection, like *Norman*, *SWEEP* was hit fairly hard by the addition of W32/Fosforo to the polymorphic test set, as well as the numerous new archive files which were added to the test sets this month.

Two other features of note came to light in this review. The first was that the IDE files used to add virus detection to the product must be placed manually in a directory which is not the main program directory – slightly counter-intuitive.

Perhaps of greater note is that the detection for W32/CTX was added on the day of the review deadline – though the virus had been declared to be In the Wild for some time by that point.

VirusBuster VirusBuster 1.06

ItW	99.95%	Macro	100.00%
Polymorphic	91.01%	Standard	99.55%

The main frustration with *VirusBuster* came when trying to determine a version number for the product – this seemed impossible to determine from within the software. In the end the package version number was selected – though quite how a user will be able to tell which virus definitions are loaded remains a mystery.

On the detection front *VirusBuster*'s behaviour is best described as variable. Detection is good in all areas, with the *Linux* detection rates being in the top of the field, but the polymorphic detection rate is distinctly weak.

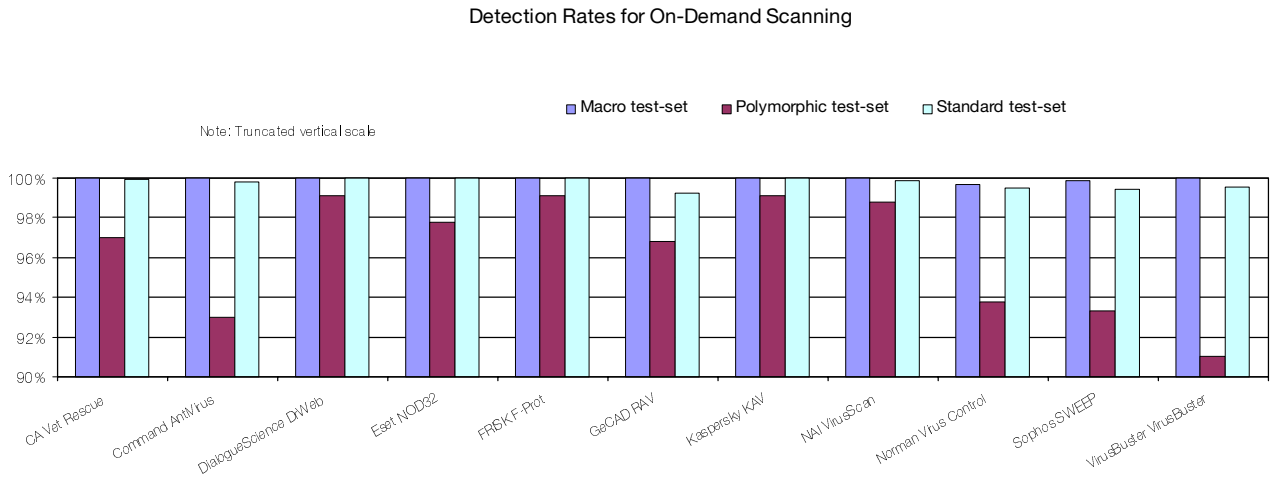
In the past there have been complaints that too many products detect almost all files in the test sets. Frequent additions to the polymorphic set should mean this will edge well away from a collection which can be detected fully.

On-Access Scanning

Of the products reviewed four had some form of on-access component.

The first to be considered outside the scope of the review was *Sophos SWEEP*. Although *Linux* can be used to support an *InterCheck Server*, which supports on-access scanning, there is no *InterCheck Client* for *Linux*.

In effect, this means that on-access scanning can be done by a *Linux* machine – but can only be performed on behalf of a



machine which offers support for the *InterCheck* client, thus making the machine incapable of scanning itself.

The three remaining products to offer on-access scanning offer this feature as a kernel module. This allows for fully native file access interception – but such modules are kernel-dependent, which leads to problems in that a standardized module cannot be supplied.

Kaspersky Lab circumvents this problem by supplying make files and source for the module, which is compiled by the user. Unfortunately, on the default installation of *SuSE Linux* used for testing, compilation failed to complete.

The situation for *DialogueScience*'s *Dr Web* was somewhat different, in that *DialogueScience* supplied a pre-constructed module which was tailored to the kernel versions under test. Again, there was one fatal problem with this, in that the version of *SAMBA* used in this test was not compatible with *Spider*'s requirements.

The most hopeful performance was offered by *ESET*'s *Amon* module – which loaded and performed interception as advertised when test accesses were performed on individual files. Admittedly, the behaviour was not particularly informative to the user, since access was denied to infected objects without any explanation.

With such a promising start it came as something of a disappointment when the on-access tests were commenced.

On numerous occasions during the on-access scanning tests the *Linux* machine simply locked up – accepting no input whatsoever other than the power switch. Again, testing was left for standalones, where experimentation is a luxury not possible in the time available for a comparative.

VB 100% Awards

All this talk of on-access scanning steers the course of discussion to that old favourite, the VB 100% awards.

The expectation that a product should be able to detect both on access and on demand remains a primary feature in the awarding of the VB 100% logo.

As indicated, there were no products that were able to install upon the stated default test machine network and thus none in this comparative was eligible for the VB 100% award.

There is no denying that there are great problems for the developers in achieving portable code for a multiplicity of kernels, and these may well prove insurmountable for those users who make use of particularly mephistophelean kernel configurations.

It is equally clear, however, that the on-access components have worked on those kernels that are in more common usage. The challenge for obtaining a VB 100% award in future tests will be partially in providing such a component – but more in providing one which will work on a wide range of platforms.

More than ever this means that the *Linux* comparatives cannot be seen as a representation of anything other than how the selected test configuration is supported. Making the assumption that these results would be identical on other kernels or configurations would be foolhardy.

Conclusion

The addition of *Linux* as a platform for comparative review has certainly brought some new and challenging problems to the testing process, due simply to the smaller number of features that can be taken for granted on this platform.

Anti-virus products are still, by and large, quite young in the *Linux* market, with those features such as quarantining, which are taken for granted elsewhere, being a rarity in the products reviewed.

It does look as if a certain degree of market impetus is present, if the rapid changes in the products available and the features on existing products are anything to judge by.

One disappointment, however, was the generally poor level of detection for the *Linux* files which were added into the test sets.

Of course, some of these *Linux* files are certain to be missed without the use of archive scanning (though this could prove a good reason for enabling the scanning of archives by default, at least on this platform).

There is something of a potential problem involving circular reasoning with this lack of detection. The nature of *Linux* is such that the need for virus protection on this platform is somewhat lower than it is on other platforms – providing the correct procedures are followed. For this reason, the development of anti-virus products for *Linux* has been slow historically.

However, if the rate of detection of *Linux* files is low, few customers are likely to come forward, there will be no impetus for development and detection rates are unlikely to increase.

Whether this cycle is realized or boom ensues only time will tell.

Technical Details

As this is the first in a potentially long series of *Linux* comparative reviews, the technical details come with what amounts to an explanatory note.

The version of *Linux* chosen was selected deliberately so as not to be one of the most commonly installed, while still being sufficiently large to have relevance to developers. *SuSE* version 7.2 was chosen over version 7.3 as this was considered to be the more stable of the two.

In effect, the ideal platform for the test would provide a slight challenge to a product's cross-platform abilities, yet at the same time avoiding any unnecessary obstacles from known bugs.

The test environment, as noted above, was designed to mimic at least a possible real-world situation. In this case a *Linux* machine using *SAMBA* was considered a 'normal' application, while not the simplest for an on-access scanner to negotiate.

Technical Details

Linux machine: 750 MHz AMD Duron workstation with 128MB RAM, 8 GB and 4 GB dual hard disks, CD-ROM, LS120 and 3.5-inch floppy, running *SuSE Linux 7.2* (Glibc 2.2, *Linux* kernel 2.4.4)

Client machine: 750 MHz AMD Duron workstation with 128MB RAM, 8 GB and 4 GB dual hard disks, CD-ROM, LS120 and 3.5-inch floppy, running *Microsoft Windows 2000 Professional*.

Connected by *Samba 2.2.0 -15*.

Virus test sets: Complete listings of the test sets used are at http://www.virusbtn.com/Comparatives/Linux/2002/02test_sets.html. A full description of the results calculations protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

On-demand tests	Polymorphic	
	Number missed	%
CA Vet Rescue	106	97.00%
Command AntiVirus	164	93.01%
DialogueScience DrWeb	34	99.14%
Eset NOD32	89	97.75%
FRISK F-Prot	164	93.01%
GeCAD RAV	87	96.79%
Kaspersky KAV	35	99.08%
NAI VirusScan	48	98.78%
Norman Virus Control	151	93.76%
Sophos SWEEP	154	93.31%
VirusBuster VirusBuster	171	91.01%

Erratum

VB regrets that a number of errors appeared in *Virus Bulletin*'s recent comparative review on *Linux* (see *VB*, April 2002, p.17). The errors occurred in the table displaying the results of on-demand scanning tests and relate to the number of samples missed in the polymorphic test set. The correct figures are printed here. *VB* apologises for any confusion ▮