This report aims to compare the effectiveness of Symantec Endpoint Protection 12.1 with Trend Micro's Deep Security 8 virtual appliance.

The tests were conducted between 16/03/2012 and 05/04/2012 using the most up to date versions of the software available.

The products were exposed to genuine internet threats that real customers could have encountered during the test period. Crucially, this exposure was carried out in a realistic way, reflecting a customer's experience as closely as possible.

For example, each test system visited real, infected websites that significant numbers of internet users were encountering at the time of the test. These results reflect what would have happened if those users were using one of the products tested.

**EXECUTIVE SUMMARY**

- **Which was the best product?**
  In terms of protection Symantec Endpoint Protection was the best performing product. (The scope of this test does not cover performance or virtual desktop densities.)

- **Malware designed to attack physical systems works just as well in virtual environments**
  All of the malware used in this test was verified on physical systems, but was also observed to attack virtual desktops.

- **Blocking websites is a very effective approach to protection**
  The product that performed best in this test did not allow any malware to run on the target systems.

- **The best-performing product in this test had a technical advantage**
  Symantec Endpoint Protection performed best. It is a fully-featured security product that must be installed on each virtual desktop system. Trend Micro Deep Security is a virtual appliance that is unable to gain as much control over each desktop. This is a limitation imposed by virtual environment.

Simon Edwards, Dennis Technology Labs
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive summary</td>
<td>1</td>
</tr>
<tr>
<td>Contents</td>
<td>2</td>
</tr>
<tr>
<td>1. Total Accuracy Ratings</td>
<td>3</td>
</tr>
<tr>
<td>2. Protection Ratings</td>
<td>4</td>
</tr>
<tr>
<td>3. Protection Scores</td>
<td>5</td>
</tr>
<tr>
<td>4. Protection Details</td>
<td>6</td>
</tr>
<tr>
<td>5. False Positives</td>
<td>7</td>
</tr>
<tr>
<td>6. The Tests</td>
<td>10</td>
</tr>
<tr>
<td>7. Test Details</td>
<td>12</td>
</tr>
<tr>
<td>8. Conclusions</td>
<td>15</td>
</tr>
<tr>
<td>Appendix A: Terms and Definitions</td>
<td>16</td>
</tr>
<tr>
<td>Appendix B: Tools</td>
<td>17</td>
</tr>
<tr>
<td>Appendix C: Terms of the test</td>
<td>18</td>
</tr>
</tbody>
</table>
1. **Total Accuracy Ratings**

The total accuracy ratings provide a way to judge how effectively the security programs work by looking at a single graph. It takes into account how accurately the programs treated threats and handled legitimate software.

We believe that anti-malware software should not just detect threats. They should allow legitimate software to run unhindered as well.

We ran two distinct tests: one that measured how the products handled internet threats and one that measured how they handled legitimate programs.

When a product fails to protect the system against a threat it is compromised. When it warns against, or even blocks, legitimate software then it generates a ‘false positive’ result.

Products gain points for stopping threats successfully and for allowing users to install and run legitimate software. Products lose points for failing to stop threats and when they handle legitimate files incorrectly.

Each product then receives a final rating based on its performance in each of the ‘threat’ and ‘legitimate software’ tests.

The following results show a combined accuracy rating, taking into account each product’s performance with both threats and non-malicious software. There is a maximum possible score of 150 and a minimum of -350.

See 5. False Positives for detailed results and an explanation on how the false positive ratings are calculated.

<table>
<thead>
<tr>
<th>Product</th>
<th>Total Accuracy Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symantec Endpoint Protection 12</td>
<td>148</td>
</tr>
<tr>
<td>Trend Micro Deep Security Appliance (no agent, no recommendations)</td>
<td>6.85</td>
</tr>
</tbody>
</table>

The total accuracy ratings take into account successes and failures with both malware and legitimate applications.
The following results show how each product has been scored for its accuracy in detecting and handling malware only. They do not take into account false positives.

We awarded two points for defending against a threat, one for neutralizing it and deducted two points every time a product allowed the system to be compromised. The best possible score is 100 and the worst is -100.

The reason behind this score weighting is to give credit to products that deny malware an opportunity to tamper with the system and to penalize those that allow malware to damage it. It is quite possible that a compromised system will be made unstable, or even unusable without expert knowledge. Even if active malware was removed, we considered such damaged systems to count as being compromised.

The Symantec product defended against 50 out of the 50 threats and scored 100. It gains double points for each defense (2x50), totaling 100. Trend’s product defended eight times, neutralized 10 times (1x10) but was compromised 32 times. Its score is calculated like this: (2x8) + (1x10) - (2x32) = -38.

**PROTECTION RATINGS**

<table>
<thead>
<tr>
<th>Product</th>
<th>Protection Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symantec Endpoint Protection 12</td>
<td>100</td>
</tr>
<tr>
<td>Trend Micro Deep Security Appliance (no agent, no recommendations)</td>
<td>-38</td>
</tr>
</tbody>
</table>
3. Protection Scores

The following illustrates the general level of protection provided by each of the security products, combining the defended and neutralized incidents into an overall figure.

This set of figures illustrates how many times the systems were protected, either via a defense or a neutralization.

This figure is not weighted with an arbitrary scoring system as it was in 1. Total Accuracy Ratings and 2. Protection Ratings.

The average protection levels afforded by the tested products, when exposed to the threats used in this test, was 68 per cent.

---

The protection scores simply indicate how many times each product prevented a threat from compromising the system.

PROTECTION SCORES

<table>
<thead>
<tr>
<th>Product</th>
<th>Protected Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symantec Endpoint Protection 12</td>
<td>50</td>
</tr>
<tr>
<td>Trend Micro Deep Security Appliance (no agent, no recommendations)</td>
<td>18</td>
</tr>
</tbody>
</table>

(Average: 68 per cent)
4. PROTECTION DETAILS

The security products provided different levels of protection. When a product defended against a threat, it prevented the malware from gaining a foothold on the target system. A threat might have been able to exploit or infect the system and, in some cases, the product neutralized it either after the exploit ran or later. When it couldn’t the system was compromised.

The products are ordered according to how many ‘Defended’ results they generated. For overall protection scores see Protection Scores on page 5.

PROTECTION DETAILS

<table>
<thead>
<tr>
<th>Product</th>
<th>Target Defended</th>
<th>Target Neutralized</th>
<th>Target Compromised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symantec Endpoint Protection 12</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Trend Micro Deep Security Appliance (no agent, no recommendations)</td>
<td>8</td>
<td>10</td>
<td>32</td>
</tr>
</tbody>
</table>

The graph represents the number of ‘Defended’, ‘Neutralized’, and ‘Compromised’ targets for both products.
5. False Positives

5.1 False positive scores

A security product needs to be able to protect the system from threats, while allowing legitimate software to work properly. When legitimate software is misclassified a false positive is generated. We split the results into two main groups because most products we test take one of two basic approaches when attempting to protect the system from the legitimate programs. They either warn that the software was suspicious or take the more decisive step of blocking it. Blocking a legitimate application is more serious than issuing a warning because it directly hampers the user.

![False Positive Scores Graph](image)

When generating a false positive the products were more likely to warn against installing or running a program than blocking it completely.

### False Positive Scores

<table>
<thead>
<tr>
<th>False Positive Type</th>
<th>Product</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warnings</td>
<td>Trend Micro Deep Security Appliance (no agent, no recommendations)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Symantec Endpoint Protection 12</td>
<td>0</td>
</tr>
<tr>
<td>Blockings</td>
<td>Trend Micro Deep Security Appliance (no agent, no recommendations)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Symantec Endpoint Protection 12</td>
<td>1</td>
</tr>
</tbody>
</table>
5.2 Taking file prevalence into account
The prevalence of each file is significant. If a product misclassified a common file then the situation would be more serious than if it failed to detect a less common one. That said, it is usually expected that anti-malware programs should not misclassify any legitimate software.

The files selected for the false positive testing were organized into five groups: Very High Impact, High Impact, Medium Impact, Low Impact and Very Low Impact.

These categories were based on download numbers as reported by sites including Download.com at the time of testing. The ranges for these categories are recorded in the table below:

<table>
<thead>
<tr>
<th>Impact category</th>
<th>Prevalence (downloads in the previous week)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High Impact</td>
<td>&gt;20,000</td>
</tr>
<tr>
<td>High Impact</td>
<td>1,000 - 20,000</td>
</tr>
<tr>
<td>Medium Impact</td>
<td>100 - 999</td>
</tr>
<tr>
<td>Low Impact</td>
<td>25 - 99</td>
</tr>
<tr>
<td>Very Low Impact</td>
<td>&lt; 25</td>
</tr>
</tbody>
</table>

5.3 Modifying scores
The following set of score modifiers were used to create an impact-weighted accuracy score. Each time a product allowed a new legitimate program to install and run it was awarded one point. It lost points (or fractions of a point) if and when it generated a false positive. We used the following score modifiers:

<table>
<thead>
<tr>
<th>False positive action</th>
<th>Impact category</th>
<th>Score modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocked</td>
<td>Very High Impact</td>
<td>-5</td>
</tr>
<tr>
<td></td>
<td>High Impact</td>
<td>-2</td>
</tr>
<tr>
<td></td>
<td>Medium Impact</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>Low Impact</td>
<td>-0.5</td>
</tr>
<tr>
<td></td>
<td>Very Low Impact</td>
<td>-0.1</td>
</tr>
<tr>
<td>Warning</td>
<td>Very High Impact</td>
<td>-2.5</td>
</tr>
<tr>
<td></td>
<td>High Impact</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>Medium Impact</td>
<td>-0.5</td>
</tr>
<tr>
<td></td>
<td>Low Impact</td>
<td>-0.25</td>
</tr>
<tr>
<td></td>
<td>Very Low Impact</td>
<td>-0.05</td>
</tr>
</tbody>
</table>
5.4 Distribution of impact categories

Products that scored highest were the most accurate when handling the legitimate applications used in the test. The best score possible is 50, while the worst would be -250 (assuming that all applications were classified as Very High Impact and were blocked). In fact the distribution of applications in the impact categories was not restricted only to Very High Impact. The table below shows the true distribution:

### FALSE POSITIVE CATEGORY FREQUENCY

<table>
<thead>
<tr>
<th>Prevalence Rating</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High Impact</td>
<td>10</td>
</tr>
<tr>
<td>High Impact</td>
<td>20</td>
</tr>
<tr>
<td>Medium Impact</td>
<td>10</td>
</tr>
<tr>
<td>Low Impact</td>
<td>5</td>
</tr>
<tr>
<td>Very Low Impact</td>
<td>5</td>
</tr>
</tbody>
</table>

5.5 False positive ratings

Combining the impact categories with weighted scores produces the following false positive accuracy ratings.

![False Positive Ratings](image)

When a product misclassified a popular program it faced a stronger penalty than if the file was more obscure.
6. The Tests

6.1 The threats
Providing a realistic user experience was important in order to illustrate what really happens when a user encounters a threat on the internet. For example, in these tests web-based malware was accessed by visiting an original, infected website using a web browser, and not downloaded from a CD or internal test website.

All target systems were fully exposed to the threats. This means that any exploit code was allowed to run, as were other malicious files. They were run and permitted to perform exactly as they were designed to, subject to checks made by the installed security software. A minimum time period of five minutes was provided to allow the malware an opportunity to act.

6.2 Test rounds
Tests were conducted in rounds. Each round recorded the exposure of every product to a specific threat. For example, in ‘round one’ each of the products were exposed to the same malicious website.

At the end of each round the test systems were completely reset to remove any possible trace of malware before the next test began.

<table>
<thead>
<tr>
<th>Incident</th>
<th>Product Code</th>
<th>Alert (intro)</th>
<th>Effect (intro)</th>
<th>Threat Report (intro)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SEP</td>
<td>Toaster</td>
<td>Blocked</td>
<td>food.greurion.com/content/s</td>
</tr>
<tr>
<td>2</td>
<td>SEP</td>
<td>Toaster</td>
<td>Blocked</td>
<td>Blackhole Toolkit Website</td>
</tr>
<tr>
<td>3</td>
<td>SEP</td>
<td>Toaster</td>
<td>Blocked</td>
<td>Blackhole Toolkit Website</td>
</tr>
<tr>
<td>4</td>
<td>SEP</td>
<td>Pop-up</td>
<td>Quarantined</td>
<td>vale%20presents%20ostetricic</td>
</tr>
<tr>
<td>5</td>
<td>SEP</td>
<td>Toaster</td>
<td>Blocked</td>
<td>TSPY_BANKER.DMS</td>
</tr>
<tr>
<td>6</td>
<td>SEP</td>
<td>Toaster</td>
<td>Blocked</td>
<td>Blackhole Toolkit Website</td>
</tr>
<tr>
<td>7</td>
<td>SEP</td>
<td>Pop-up</td>
<td>Quarantined</td>
<td>Trojan.ADH.2</td>
</tr>
<tr>
<td>8</td>
<td>SEP</td>
<td>Toaster</td>
<td>Quarantined</td>
<td>PAK_Gwnwric.D01</td>
</tr>
</tbody>
</table>

Each ‘round’ exposed every product to one specific threat. The set of records for round four (highlighted above) shows how the products responded to a particular threat. The Alert entry shows the type of alert, if any, that the product generated. The Effect entry describes what the product claimed to do with the threat, while the Threat Report lists more details. We used forensic techniques to verify whether or not the product achieved what it claims.

6.3 Monitoring
Close logging of the target systems was necessary to gauge the relative successes of the malware and the anti-malware software. This included recording activity such as network traffic, the creation of files and processes and changes made to important files.

6.4 Levels of protection
The products displayed different levels of protection. Sometimes a product would prevent a threat from executing, or at least making any significant changes to the target system. In other cases a threat might be able to perform some tasks on the target (such as exploiting a security vulnerability or executing a malicious program), after which the security product would intervene and remove some or all of the malware.

Finally, a threat may be able to bypass the security product and carry out its malicious tasks unhindered. It may even be able to disable the security software. Occasionally Windows’ own protection system might handle a threat while the anti-virus program ignored...
it. Another outcome is that the malware may crash for various reasons.

The different levels of protection provided by each product were recorded following analysis of the log files.

If malware failed to perform properly in a given incident, perhaps because of the very presence of the security product, rather than any specific defending action that the product took, the product was given the benefit of the doubt and a Defended result was recorded.

If the test system was damaged, becoming hard to use following an attempted attack, this was counted as a compromise even if the active parts of the malware had eventually been removed by the product.

6.5 Types of protection
All of the products tested provided two main types of protection: real-time and on-demand. Real-time protection monitors the system constantly in an attempt to prevent a threat from gaining access.

On-demand protection is essentially a ‘virus scan’ that is run by the user at an arbitrary time.

The test results note each product’s behavior when a threat is introduced and afterwards. The real-time protection mechanism was monitored throughout the test, while an on-demand scan was run towards the end of each test to measure how safe the product determined the system to be.

Manual scans were run only when a tester determined that malware had made an interaction with the target system. In other words, if the security product claimed to block the attack at the initial stage, and the monitoring logs supported this claim, the case was considered closed and a Defended result was recorded.
7. TEST DETAILS

7.1 The targets
To create a fair testing environment, each product was installed on a clean Windows XP Professional target system. The operating system was updated with Windows XP Service Pack 3 (SP3), although no later patches or updates were applied.

We test with Windows XP SP3 and Internet Explorer 7 due to the high prevalence of internet threats that rely on this combination. The prevalence of these threats suggests that there are many systems with this level of patching currently connected to the internet.

A selection of legitimate but old software was pre-installed on the target systems. These posed security risks, as they contained known vulnerabilities. They included out of date versions of Adobe Flash Player and Adobe Reader.

A different security product was then installed on each system according to recommendations made by each vendor. Symantec Endpoint Protection was installed using default settings, as requested.

Trend Micro Deep Security was deployed without a software agent and without applying any recommended security patches (hence the “no recommendations” label used in the charts). The management system and protection features were set up as recommended by Trend Micro.

Each product’s update mechanism was used to download the latest version with the most recent definitions and other elements.

Due to the dynamic nature of the tests, which were carried out in real-time with live malicious websites, the products’ update systems were allowed to run automatically and were also run manually before each test round was carried out.

The products were also allowed to 'call home' should they be programmed to query databases in real-time. Some products might automatically upgrade themselves during the test. At any given time of testing, the very latest version of each program was used.

Each target system was a virtual desktop running on an HP ProLiant DL360 G5 server running VMware ESXi 4.1. Each virtual desktop was allocated 2GB RAM, one processor and up to 27GB disk space.

The management tools required by the different products were deployed, including the management consoles Symantec Endpoint Protection was run as a stand-alone product without management tools.

7.2 Threat selection
The malicious web links (URLs) used in the tests were not provided by any anti-malware vendor. They were picked from lists generated by Dennis Technology Labs’ own malicious site detection system, which uses popular search engine keywords submitted to Google. It analyses sites that are returned in the search results from a number of search engines and adds them to a database of malicious websites. In all cases, a control system (Verification Target System - VTS) was used to confirm that the URLs linked to actively malicious sites.

Malicious URLs and files are not shared with any vendors during the testing process.

7.3 Test stages
There were three main stages in each individual test:

1. Introduction
2. Observation
3. Remediation

During the Introduction stage, the target system was exposed to a threat. Before the threat was introduced, a snapshot was taken of the system. This created a list of Registry entries and files on the hard disk. We used Regshot (see Appendix B: Tools) to take and compare system snapshots. The threat was then introduced.

Immediately after the system’s exposure to the threat, the Observation stage is reached. During this time, which typically lasted at least 10 minutes, the tester monitored the system both visually and using a range of third-party tools. The tester reacted to pop-ups and other prompts according to the directives described below (see 7.6 Observation and intervention).

In the event that hostile activity to other internet users was observed, such as when spam was being sent by the target, this stage was cut short. The Observation stage concluded with another system snapshot. This ‘exposed’ snapshot was compared to the original ‘clean’ snapshot and a report generated. The system was then rebooted.
The **Remediation** stage is designed to test the products’ ability to clean an infected system. If it defended against the threat in the **Observation** stage, then we skipped this stage. An on-demand scan was run on the target, after which a ‘scanned’ snapshot was taken. This was compared to the original ‘clean’ snapshot and a report was generated. All log files, including the snapshot reports and the product’s own log files, were recovered from the target. In some cases the target became so damaged that log recovery was considered impractical. The target was then reset to a clean state, ready for the next test.

### 7.4 Threat introduction
Malicious websites were visited in real-time using Internet Explorer. This risky behavior was conducted using live internet connections. URLs were typed manually into Internet Explorer’s address bar.

Web-hosted malware often changes over time. Visiting the same site over a short period of time can expose systems to what appear to be a range of threats (although it may be the same threat, slightly altered to avoid detection). Also, many infected sites will only attack a particular IP address once, which makes it hard to test more than one product against the same threat.

In order to improve the chances that each target system received the same experience from a malicious web server, we used a web replay system. When the verification target systems visited a malicious site, the page’s content, including malicious code, was downloaded, stored and loaded into the replay system. When each target system subsequently visited the site, it received exactly the same content.

The network configurations were set to allow all products unfettered access to the internet throughout the test, regardless of the web replay systems.

### 7.5 Secondary downloads
Established malware may attempt to download further files (secondary downloads), which are stored in a cache by a proxy on the network and re-served to other targets in some circumstances. These circumstances include cases where:

1. The download request is made using HTTP (e.g. http://badsite.example.com/...) and
2. The same filename is requested each time (e.g. badfile1.exe)

There are scenarios in which target systems receive different secondary downloads. These include cases where:

1. The download request is made using HTTPS or a non-web protocol such as FTP or
2. A different filename is requested each time (e.g. badfile2.exe; random357.exe)

### 7.6 Observation and intervention
Throughout each test, the target system was observed both manually and in real-time. This enabled the tester to take comprehensive notes about the system’s perceived behavior, as well as to compare visual alerts with the products’ log entries. At certain stages the tester was required to act as a regular user. To achieve consistency, the tester followed a policy for handling certain situations, including dealing with pop-ups displayed by products or the operating system, system crashes, invitations by malware to perform tasks and so on.

This user behavior policy included the following directives:

1. Act naively. Allow the threat a good chance to introduce itself to the target by clicking OK to malicious prompts, for example.
2. Don’t be too stubborn in retrying blocked downloads. If a product warns against visiting a site, don’t take further measures to visit that site.
3. Where malware is downloaded as a Zip file, or similar, extract it to the Desktop then attempt to run it. If the archive is protected by a password, and that password is known to you (e.g. it was included in the body of the original malicious email), use it.
4. Always click the default option. This applies to security product pop-ups, operating system prompts (including Windows firewall) and malware invitations to act.
5. If there is no default option, wait. Give the prompt 20 seconds to choose a course of action automatically.
6. If no action is taken automatically, choose the first option. Where options are listed vertically, choose the top one. Where options are listed horizontally, choose the left-hand one.

### 7.7 Remediation
When a target is exposed to malware, the threat may have a number of opportunities to infect the system. The security product also has a number of chances to protect the target. The snapshots explained in **7.3 Test stages** provided information that was used to analyze a system’s final state at the end of a test.
Before, during and after each test, a ‘snapshot’ of the target system was taken to provide information about what had changed during the exposure to malware. For example, comparing a snapshot taken before a malicious website was visited to one taken after might highlight new entries in the Registry and new files on the hard disk. Snapshots were also used to determine how effective a product was at removing a threat that had managed to establish itself on the target system. This analysis gives an indication as to the levels of protection that a product has provided.

These levels of protection have been recorded using three main terms: defended, neutralized, and compromised. A threat that was unable to gain a foothold on the target was defended against; one that was prevented from continuing its activities was neutralized; while a successful threat was considered to have compromised the target.

A defended incident occurs where no malicious activity is observed with the naked eye or third-party monitoring tools following the initial threat introduction. The snapshot report files are used to verify this happy state.

If a threat is observed to run actively on the system, but not beyond the point where an on-demand scan is run, it is considered to have been neutralized. Comparing the snapshot reports should show that malicious files were created and Registry entries were made after the introduction. However, as long as the ‘scanned’ snapshot report shows that either the files have been removed or the Registry entries have been deleted, the threat has been neutralized.

The target is compromised if malware is observed to run after the on-demand scan. In some cases a product might request a further scan to complete the removal. We considered secondary scans to be acceptable, but further scan requests would be ignored. Even if no malware was observed, a compromise result was recorded if snapshot reports showed the existence of new, presumably malicious files on the hard disk, in conjunction with Registry entries designed to run at least one of these files when the system booted. An edited ‘hosts’ file or altered system file also counted as a compromise.

7.8 Automatic monitoring
Logs were generated using third-party applications, as well as by the security products themselves. Manual observation of the target system throughout its exposure to malware (and legitimate applications) provided more information about the security products’ behavior. Monitoring was performed directly on the target system and on the network.

Client-side logging
A combination of Process Explorer, Process Monitor, TcpView and Wireshark were configured to monitor the target systems. Regshot was used between each testing stage to record a system snapshot. A number of Dennis Technology Labs-created scripts were also used to provide additional system information. Each product was able to generate some level of logging itself.

Process Explorer and TcpView were run throughout the tests, providing a visual cue to the tester about possible malicious activity on the system. In addition, Wireshark’s real-time output, and the display from the web proxy (see Network logging, below), indicated specific network activity such as secondary downloads.

Process Monitor also provided valuable information to help reconstruct malicious incidents. Both Process Monitor and Wireshark were configured to save their logs automatically to a file. This reduced data loss when malware caused a target to crash or reboot.

In-built Windows commands such as 'systeminfo' and 'sc query' were used in custom scripts to provide additional snapshots of the running system’s state.

Network logging
All target systems were connected to a live internet connection, which incorporated a transparent web proxy and a network monitoring system. All traffic to and from the internet had to pass through this system. Further to that, all web traffic had to pass through the proxy as well. This allowed the testers to capture files containing the complete network traffic. It also provided a quick and easy view of web-based traffic, which was displayed to the testers in real-time.

The network monitor was a dual-homed Linux system running as a transparent router, passing all web traffic through a Squid proxy.

An HTTP replay system ensured that all target systems received the same malware as each other. It was configured to allow access to the internet so that products could download updates and communicate with any available ‘in the cloud’ servers.
8. CONCLUSIONS

Where are the threats?
The threats used in this test were genuine, real-life threats that were infecting victims globally at the same time as we tested the products. In almost every case the threat was launched from a legitimate website that had been compromised by an attacker. The types of infected or malicious sites were varied, which demonstrates that effective anti-virus software is essential for those who want to use the web using a Windows PC.

Some threats installed automatically when a user visited the infected webpage. This infection was usually invisible to a casual observer and rarely did the malware make itself known, unless it was installing a fake anti-virus or other utility program.

One type of threat changes the system’s Hosts file as part of an attempt to steal bank details.

Where does protection start?
The product that succeeded best at preventing the threats from succeeding was Symantec Endpoint Protection. In every case it blocked the threat before it was able to execute. Trend Micro Deep Security blocked some malicious sites but it also neutralized threats after they arrived on the system.

It is better to block sites or exploits before they can interfere with the target. Evidence of this can be seen in the number of compromises suffered by the system protected by Trend Micro Deep Security.

Why are the results so different between products?
This was an anti-malware test designed for virtualized desktop systems. There are generally two main approaches that security vendors take to protecting virtual desktops: one is to install a full security product on each desktop; the other is to integrate the protection within the virtualization platform. This is often in the form of a ‘virtual appliance’.

There are advantages to both approaches. Installing a full security suite on each virtual desktop provides a wide range of protection measures. For example, the product can monitor what happens in the system’s memory and block certain potentially-malicious behavior.

A significant advantage to using a virtual appliance is that one installation of a security product can take on the workload of protecting lots of systems, leaving the individual virtual desktops free of software that could potentially cause performance problems.

It is worth noting that virtual desktop environments may be set up in a number of ways and for a wide range of purposes. In some cases users may be able to browse the internet freely and install software. In other scenarios users will be locked down to a greater extent.

Other security products may also be used in such environments. Web blocking gateway appliances, for example, would be likely additions to the network.

In this test the ‘users’ are clearly able to browse a full range of websites without restriction.

In Trend Micro Deep Security’s case the product is unable to interact directly with the memory of each target desktop installation without the use of an additional software agent installed on each desktop. This is a situation common with all current VMware-based virtual appliances. As such, in this test it has a significant disadvantage to Symantec Endpoint Protection, which is able to access the memory.

Sorting good sites and applications from bad
Anti-malware products need to be able to distinguish between malicious and non-malicious programs. Both products in this test performed well in this area.

The false positive candidate programs are mainly suitable only for consumers and are less likely to find their way onto a business network. For this reason, although both products blocked some high impact files, we don’t consider this to be a serious problem.

Anti-virus is important (but not a panacea)
This test shows that even with a relatively small sample set of 50 threats there is a significant difference in performance between the products. Most importantly, it illustrates this difference using real threats that were attacking real computers at the time of testing.

The presence of anti-virus software can be seen to decrease the chances of a malware infection even when the only sites being visited are proven to be actively malicious.
### APPENDIX A: TERMS AND DEFINITIONS

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compromised</td>
<td>Malware continues to run on an infected system, even after an on-demand scan.</td>
</tr>
<tr>
<td>Defended</td>
<td>Malware was prevented from running on, or making changes to, the target.</td>
</tr>
<tr>
<td>False Positive</td>
<td>A legitimate application was incorrectly classified as being malicious.</td>
</tr>
<tr>
<td>Introduction</td>
<td>Test stage where a target system is exposed to a threat.</td>
</tr>
<tr>
<td>Neutralized</td>
<td>Malware or exploit was able to run on the target, but was then removed by the security product.</td>
</tr>
<tr>
<td>Observation</td>
<td>Test stage during which malware may affect the target.</td>
</tr>
<tr>
<td>On-demand (protection)</td>
<td>Manual ‘virus’ scan, run by the user at an arbitrary time.</td>
</tr>
<tr>
<td>Prompt</td>
<td>Questions asked by software, including malware, security products and the operating system. With security products, prompts usually appear in the form of pop-up windows. Some prompts don’t ask questions but provide alerts. When these appear and disappear without a user’s interaction, they are called ‘toasters’.</td>
</tr>
<tr>
<td>Real-time (protection)</td>
<td>The ‘always-on’ protection offered by many security products.</td>
</tr>
<tr>
<td>Remediation</td>
<td>Test stage that measures a product’s abilities to remove any installed threat.</td>
</tr>
<tr>
<td>Round</td>
<td>Test series of multiple products, exposing each target to the same threat.</td>
</tr>
<tr>
<td>Snapshot</td>
<td>Record of a target’s file system and Registry contents.</td>
</tr>
<tr>
<td>Target</td>
<td>Test system exposed to threats in order to monitor the behavior of security products.</td>
</tr>
<tr>
<td>Threat</td>
<td>A program or other measure designed to subvert a system.</td>
</tr>
<tr>
<td>Update</td>
<td>Code provided by a vendor to keep its software up to date. This includes virus definitions, engine updates and operating system patches.</td>
</tr>
</tbody>
</table>
APPENDIX B: TOOLS

Ebtables
http://ebtables.sourceforge.net
The ebtables program is a filtering tool for a bridging firewall. It can be used to force network traffic transparently through the Squid proxy.

Fiddler2
www.fiddlertool.com
A web traffic (HTTP/S) debugger used to capture sessions when visiting an infected site using a verification target system (VTS).

HTTPREPLAY
www.microsoft.com
A SOCKTRC plug-in enabling the analysis and replaying of HTTP traffic.

Process Explorer
Process Explorer shows information about which handles and DLLs processes have opened or loaded. It also provides a clear and real-time indication when new processes start and old ones stop.

Process Monitor
Process Monitor is a monitoring tool that shows real-time file system, Registry and process/thread activity.

Regshot
http://sourceforge.net/projects/regshot
Regshot is an open-source Registry comparison utility that takes a snapshot of the Registry and compares it with a second one.

Squid
www.squid-cache.org
Squid is a caching web proxy that supports HTTP, HTTPS, FTP and other protocols.

Tcpdump
www.tcpdump.org
Tcpdump is a packet capture utility that can create a copy of network traffic, including binaries.

TcpView
TcpView displays network connections to and from the system in real-time.

Windows Command-Line Tools
Those used included 'systeminfo' and 'sc query'. The systeminfo command "enables an administrator to query for basic system configuration information". The sc command is "used for communicating with the NT Service Controller and services.

Wireshark
www.wireshark.org
Wireshark is a network protocol analyzer capable of storing network traffic, including binaries, for later analysis.
APPENDIX C: TERMS OF THE TEST

This test was sponsored by Symantec.

The test rounds were conducted between 16/03/2012 and 05/04/2012 using the most up to date versions of the software available on any given day.

All products were able to communicate with their back-end systems over the internet.

The products selected for this test were chosen by Symantec.

Samples were located and verified by Dennis Technology Labs.

Products were exposed to threats within 24 hours of the same threats being verified. In practice there was only a delay of up to three to four hours.

Details of the samples, including their URLs and code, were provided to Symantec only after the test was complete.

The sample set comprised 50 actively-malicious URLs and 50 legitimate applications.

Some frequently asked questions, and our answers:

Does the sponsor know what samples are used, before or during the test?
No. We don’t even know what threats will be used until the test starts. Each day we find new ones, so it is impossible for us to give this information before the test starts. Neither do we disclose this information until the test has concluded. If we did the sponsor may be able to gain an advantage that would not reflect real life.

Do you share samples with the vendors?
The sponsor is able to download all samples from us after the test is complete. Other vendors involved in the test may request a subset of the threats that compromised their products in order for them to verify our results.

The same applies to client-side logs, including the network capture files. There is a small administration fee for the provision of this service.

What is a sample?
In our tests a sample is not simply a set of malicious executable files that runs on the system. A sample is an entire replay archive that enables researchers to reproduce the attack and to determine which layer of protection was able to bypass.

Replaying the attack should, in most cases, produce the relevant executable files. If not, these are available in the client-side network capture (pcap) file.

Does the sponsor have a completely free choice of products?
No. While the sponsor may specify which products it wants us to compare, we will always advise on this decision and may refuse to include certain products if we feel that a comparison with the others is not fair.

How can a product be ‘compromised’ when your results clearly show that it detected and deleted the threat?
Our Threat Report chart shows both what the product claims to do and the end result as we determine it.

A product may well claim to have protected the system, but the forensic evidence may tell a different tale. It is quite common for a product to detect some element of an attack but to fail to prevent another.