# Network Threat Detection

## DECEMBER 2015

Dennis Technology Labs
www.DennisTechnologyLabs.com
Follow @DennisTechLabs on Twitter.com

This report aims to compare the effectiveness of a selection of threat detection appliances.

The products were exposed to internet threats that were live during the test period, as well as crafted exploit-based attacks.

These exposures were carried out in a realistic way, closely reflecting a customer's experience.

These results reflect what would have happened if a user was using one of the products and visited an infected website or was attacked in a more targeted way such as when receiving infected documents and customized links to infected websites.

This is purely a detection test, not a protection test. The threat detection appliances were deployed in 'tap mode' so that they could monitor network traffic but would not be able to block threats.

No additional anti-malware products were installed on the endpoints.

# EXECUTIVE SUMMARY

### *Products tested*

| Product | Detection accuracy | Legitimate accuracy |
|---|---|---|
| Symantec Advanced Threat Protection | 100% | 100% |
| Palo Alto Networks PA200 | 90% | 97% |
| Cisco Snort | 72% | 100% |
| Fortinet FortiGate60D | 69% | 100% |

### *Product names*

The products tested in this report were the latest versions available from each vendor on the date that the test started.

Specific 'build numbers' are available for those who wish to ascertain the exact versions that were used for testing.

These are listed in Appendix C: Product versions on page 13.

Additionally, many products offer different protection modules, including URL filtering, anti-malware detection and cloud-based sandboxes for malware execution. These were all enabled.

- **Network threat detection systems miss even well-known threats.**
  The majority of the threats used in this test were live web-based threats that were attacking users on the internet at the same time as the test was run. With the notable exception of Symantec's product, most detection appliances failed to spot all of the threats, with two recognizing less than 75 per cent.

- **The appliances were accurate when handling legitimate software**
  None of the products blocked the installation of legitimate software and only one rated a moderately popular application as being suspicious.

- **Which was the best product?**
  The most accurate appliance was Symantec Advanced Threat Protection. This was followed by a good performance by Palo Alto Networks' PA200.

Simon Edwards, Dennis Technology Labs, 18th Dec 2015

# Contents

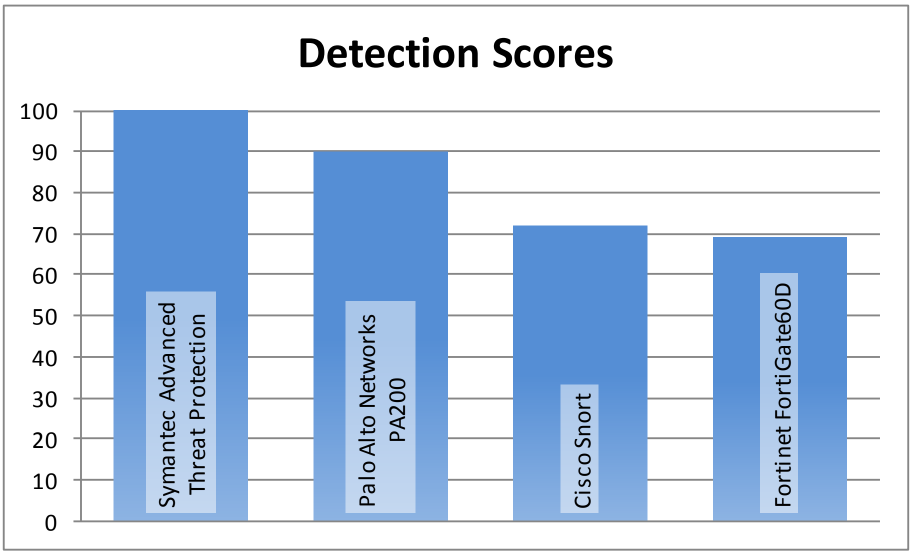
Document version 1. 0. Written 18th Dec 2015.

# 1. Detection Scores

The total accuracy ratings provide a way to judge how effectively the security programs work by looking at a single graph.

Anti-malware software should not just detect threats. It should allow legitimate software to run unhindered as well.

The results below take into account how accurately the programs treated threats and handled legitimate software.



**Every time a product detected a threat it was awarded one point. It gained no points for failing to register an attack in its logs.**

It is important to view the detection scores alongside the legitimate software ratings on page 5.

The ideal product would detect all threats and not generate false alerts when encountering legitimate applications.

When a product fails to detect a threat the network administrators will be less able to react effectively to a successful attack. When it warns against legitimate software then it generates a 'false positive' result and network administrators risk wasting time on unnecessary alerts or, worse, will fail to notice important accurate alerts that are lost in a mass of false alerts.

See 2. *Legitimate Software Ratings* on page 5 for detailed results and an explanation on how the false positive ratings are calculated.

DETECTION SCORES

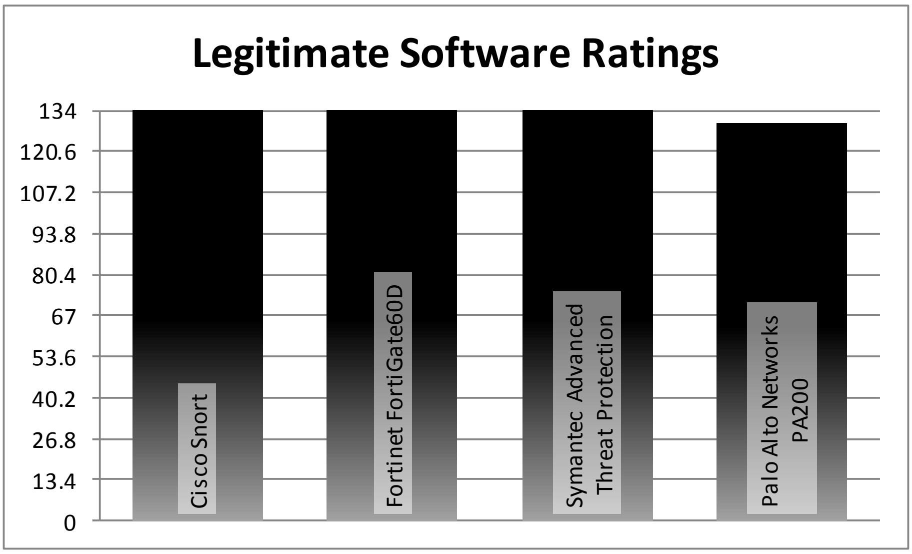
| Product | Detected Scores |
|---|---|
| Symantec Advanced Threat Protection | 100 |
| Palo Alto Networks PA200 | 90 |
| Cisco Snort | 72 |
| Fortinet FortiGate60D | 69 |

# 2. Legitimate Software Ratings

The legitimate software accuracy ratings provide a way to judge how effectively the security programs handle non-malicious software by looking at a single graph.

Anti-malware software should allow legitimate software to run unhindered. These results take into account the level of any interaction that the product demands of the user, as well as the prevalence of the legitimate program.

To understand how we calculate these ratings see 2.3 Accuracy ratings on page 7.



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

LEGITIMATE SOFTWARE RATINGS

| Product | Accuracy Rating |
|---|---|
| Cisco Snort | 134 |
| Fortinet FortiGate60D | 134 |
| Symantec Advanced Threat Protection | 134 |
| Palo Alto Networks PA200 | 129.5 |

## 2.1 Interaction ratings

A threat detection product needs to be able to detect threats, while not generating false alerts when encountering legitimate software. When legitimate software is misclassified as malware a false positive is generated.

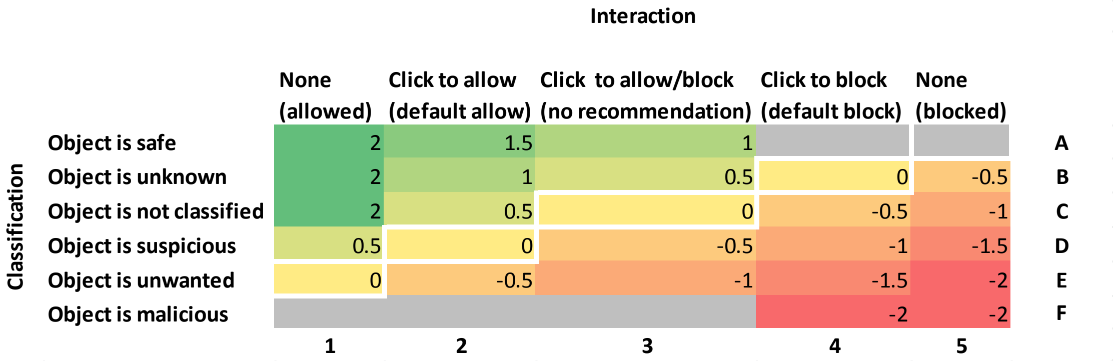
Some security products will ask the user questions when they encounter software and are not certain if it is either fully legitimate or definitely malware.

When measuring how effective each product is we take into account all of the likely outcomes: whether the product allows, blocks or asks different types of questions. In each case a score is allocated.

A product gains top marks if it allows legitimate software to install without requiring the user to answer questions or otherwise interact. It loses points the more interaction is required and the less accurately it behaves.

If a product actually generates a genuine false positive (e.g. "software is malicious") it is penalized heavily.

The results grid below shows the most likely possibilities, along with some outcomes that could only happen if a product was not working properly (e.g. A5 – Object is safe but is blocked automatically).



| Classification | Interaction<br>None (allowed) | Click to allow (default allow) | Click to allow/block (no recommendation) | Click to block (default block) | None (blocked) | |
|---|---|---|---|---|---|---|
| Object is safe | 2 | 1.5 | 1 | | | A |
| Object is unknown | 2 | 1 | 0.5 | 0 | -0.5 | B |
| Object is not classified | 2 | 0.5 | 0 | -0.5 | -1 | C |
| Object is suspicious | 0.5 | 0 | -0.5 | -1 | -1.5 | D |
| Object is unwanted | 0 | -0.5 | -1 | -1.5 | -2 | E |
| Object is malicious | | | | -2 | -2 | F |
| | 1 | 2 | 3 | 4 | 5 | |

**Top marks to products that are accurate; those that ask too many questions or are overly suspicious are penalized.**

LEGITIMATE SOFTWARE INCIDENTS

| Product | Classification | Total |
|---|---|---|
| Palo Alto Networks PA200 | Object is suspicious | 1 |

## 2.2 Prevalence ratings

The prevalence of each piece of software is significant. If a security product interferes with common applications then the situation is more serious than if it does so with rare ones. That said, it is usually expected that anti-malware programs should not interfere with any legitimate software.

The programs selected for the legitimate software testing were organized into five groups:

Very High Impact; High Impact; Medium Impact; Low Impact; and Very Low Impact.

The table below shows the relative importance of each group expressed as a numerical value. A Very High Impact application is ranked as being five times more significant than a Very Low Impact program.

LEGITIMATE SOFTWARE
PREVALENCE RATING MODIFIERS

| Impact category | Rating modifier |
|---|---|
| Very High Impact | 5 |
| High Impact | 4 |
| Medium Impact | 3 |
| Low Impact | 2 |
| Very Low Impact | 1 |

These categories were attributed to software programs based on their individual weekly download numbers as reported by third-party download sites including Download.com at the time of testing.

Files were downloaded from their original sources, excluding third-party download sites, such as Download.com, wherever possible. This was to reduce the chances that the software had been altered in any way, perhaps having potentially unwanted add-ons included with the installer.

The presence of potentially unwanted add-ons transforms the legitimate software into a product that could be blocked or altered justifiably by anti-malware software. As such they are not suitable for this legitimate software test.

The ranges for these categories, in terms of weekly downloads, are recorded in the table Legitimate Software Prevalence Categories.

LEGITIMATE SOFTWARE
PREVALENCE CATEGORIES

| Impact category | Prevalence |
|---|---|
| Very High Impact | >20,000 |
| High Impact | 1,000 - 20,000 |
| Medium Impact | 100 - 999 |
| Low Impact | 25 - 99 |
| Very Low Impact | < 25 |

## 2.3 Accuracy ratings

The legitimate software accuracy ratings are calculated by multiplying together the interaction and prevalence ratings.

*accuracy rating = number of programs x*
*(interaction rating x prevalence rating)*

For example, if a product allows 10 legitimate, Medium Impact programs to install without any interference then its rating would be calculated like this:

$$\begin{aligned} \textit{accuracy rating} &= 10 \times (2 \times 3) \\ &= 60 \end{aligned}$$

This formula creates the impact-weighted accuracy ratings used in the graph 2. Legitimate Software Ratings on page 5.

## 2.4 Distribution of impact categories

Products that scored highest were the most accurate when handling the legitimate applications used in the test.

The best theoretical score possible is 1,000, while the worst would be -1,000 (assuming that all applications were classified as Very High Impact).

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:

LEGITIMATE SOFTWARE
CATEGORY FREQUENCY

| Prevalence Rating | Frequency |
|---|---|
| Very High Impact | 2 |
| High Impact | 10 |
| Medium Impact | 3 |
| Low Impact | 3 |
| Very Low Impact | 2 |

# 3. The Tests

### 3.1 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.

### 3.2 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 was 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.

### 3.3 Monitoring

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

# 4. Test Details

### 4.1 Product configuration

The threat detection systems were configured as network taps, which means that they were given the opportunity to monitor network traffic and to detect threats that passed through that traffic.

The main advantage of 'tap mode' is that even if the system fails the network is still able to operate. The usual alternative, 'inline mode' allows products the chance to block threats as well as detect them. The downside is that if the product fails the normal network traffic could be disrupted.

A selection of legitimate but vulnerable software was pre-installed on the target systems deployed on the network behind the detection systems. The software installed on these target systems posed security risks, as they contained known security issues. They included versions of Adobe Flash Player, Adobe Reader and Java.

Due to the dynamic nature of the tests, which were carried out in real-time with live malicious websites and customized attacks, 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 firmware was used.

Each target systems was a physical PC, not a virtual machine, and was connected to the internet via its own virtual network (VLAN) to avoid cross-infection of malware.

### 4.2 Threat selection

The malicious web links (URLs) and other samples used in the tests were not provided by any anti-malware vendor.

Live URLs 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.

Custom attacks were generated using Metasploit, with minimal changes to the default settings. When these threats were web-based, they were deployed as part of apparently legitimate websites, effectively producing man-in-the middle attacks at the external network segment. In this way targets and threat detection systems have a chance to detect internet-based attacks from real websites using custom threats.

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

## 4.3 Test stages

There were two main stages in each individual test:

1. Introduction
2. Observation

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. 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.

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.

All log files, including the snapshot reports and the product's own log files, were recovered from the target.

In some cases the target may become so damaged that log recovery is considered impractical. The target was then reset to a clean state, ready for the next test.

Logs from the threat detection systems were collected both in real-time and at the end of the test. Testers also recorded any relevant activity visible to the user of the target systems, such as messages sent to the browser regarding URL blocking.

## 4.4 Threat introduction

Malicious websites were visited in real-time using the web browser. This risky behavior was conducted using live internet connections. URLs were typed manually into the browser.

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 security products unfettered access to the internet throughout the test, regardless of the web replay systems.

## 4.5 Observation and intervention

Throughout each test, the target and threat detection systems were observed both manually and in real-time. This enabled the testers 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.

## 4.6 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 used 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.

### 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.

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 threat detection systems could download updates and communicate with any available ‘in the cloud’ servers.

## 5. Conclusions

### ■ Where are the threats?

The threats used in this test were genuine, real-life threats that were infecting victims globally at the time that we tested the products.

The types of infected or malicious sites were varied, which demonstrates that effective threat detection is essential for businesses that wish to understand the nature of the internet threats facing their networks.

Most threats installed automatically when a user visited the infected webpage. Such infections were often invisible to a casual observer and so network administrators should not expect users to be able to manually alert them about attacks.

### ■ Alerts, false alerts and silence

All products succeeded in avoiding any significant level of false positive alerts.

In one minor incident the Palo Alto device claimed that one file, which was moderately popular on the internet at the time, was "suspicious". Other than that the products were accurate when handling legitimate URLs and software.

This level of accuracy is not reflected in the threat detection scores.

While Symantec's appliance detected all of the threats and Palo Alto's alerted on 90 per cent of the same, both Cisco Snort and Fortinet detected less than 75 per cent.

### ■ Next stop, protection

Clearly if a network-based threat detection systems are not universally capable of detecting threats then other, additional measures are needed to discover how attackers are attempting to compromise business networks.

Running detection systems inline, as opposed to in tap mode, will help provide some protection but, in the case of Cisco Snort and Fortinet FortiGate 60D, even this approach would not be sufficient to prevent successful public web-based attacks, let alone more targeted attacks that are customized to breach a specific organization.

It is highly likely, based on previous research by Dennis Technology Labs, that deploying endpoint anti-malware solutions, updating vulnerable software and blocking infected emails will improve an organization's level of protection.

For more details about the protection levels provided by such measures please visit www.DennistechnologyLabs.com.

# Appendix A: Terms Used

| | |
|---|---|
| Compromised | Malware continues to run on an infected system, even after an on-demand scan. |
| False Positive | A legitimate application was incorrectly classified as being malicious. |
| Introduction | Test stage where a target system is exposed to a threat. |
| Observation | Test stage during which malware may affect the target. |
| Prompt | 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'. |
| Remediation | Test stage that measures a product's abilities to remove any installed threat. |
| Round | Test series of multiple products, exposing each target to the same threat. |
| Snapshot | Record of a target's file system and Registry contents. |
| Target | Test system exposed to threats in order to monitor the behavior of security products. |
| Threat | A program or other measure designed to subvert a system. |
| Update | Code provided by a vendor to keep its software up to date. This includes virus definitions, engine updates and operating system patches. |

## Appendix B: FAQs

- This test was sponsored by Symantec.
- The test rounds were conducted between 12th October 2015 and 3rd December 2015 using the most up to date versions of the appliances 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 under close consultation with Dennis Technology Labs.
- Samples were located and verified by Dennis Technology Labs.
- Products were exposed to live 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 75 actively-malicious URLs, 25 custom threats and 20 legitimate applications.

### Do participating vendors 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.

### Do you share samples with the vendors?

The sponsor vendor is able to download samples from us after the test is complete.

Other vendors may request a small subset of the threats that compromised their products in order for them to verify our results and further understand our methodology. 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 replicate the incident, even if the original infected website is no longer available. This means that it is possible to reproduce the attack and to determine which layer of protection is was able to bypass. Replaying the attack should, in most cases, produce the relevant executable files. If not, these are usually available in the client-side network capture (pcap) file.

## Appendix C: Product versions

A product's update mechanism may upgrade the software to a new version automatically so the version used at the start of the test may be different to that used at the end.

| Vendor | Product | Build |
|---|---|---|
| Cisco | Snort | 2.9.7.6 |
| Fortinet | FortiGate 60D | Fortigate-60D v5.2.4 , build 0688,150722 (GA) |
| Palo Alto Networks | PA200 | PAN OS :6.0.9; App-version: 537-2965 |
| Symantec | Advanced Threat Protection | 2.0.0-58 |

WHILE EVERY EFFORT IS MADE TO ENSURE THE ACCURACY OF THE INFORMATION PUBLISHED IN THIS DOCUMENT, NO GUARANTEE IS EXPRESSED OR IMPLIED AND DENNIS PUBLISHING LTD DOES NOT ACCEPT LIABILITY FOR ANY LOSS OR DAMAGE THAT MAY ARISE FROM ANY ERRORS OR OMISSIONS.