Guidelines to False Positive Testing

Notice and Disclaimer of Liability Concerning the Use of AMTSO Documents

This document is published with the understanding that AMTSO members are supplying this information for general educational purposes only. No professional engineering or any other professional services or advice is being offered hereby. Therefore, you must use your own skill and judgment when reviewing this document and not solely rely on the information provided herein.

AMTSO believes that the information in this document is accurate as of the date of publication although it has not verified its accuracy or determined if there are any errors. Further, such information is subject to change without notice and AMTSO is under no obligation to provide any updates or corrections.

You understand and agree that this document is provided to you exclusively on an as-is basis without any representations or warranties of any kind whether express, implied or statutory. Without limiting the foregoing, AMTSO expressly disclaims all warranties of merchantability, non-infringement, continuous operation, completeness, quality, accuracy and fitness for a particular purpose.

In no event shall AMTSO be liable for any damages or losses of any kind (including, without limitation, any lost profits, lost data or business interruption) arising directly or indirectly out of any use of this document including, without limitation, any direct, indirect, special, incidental, consequential, exemplary and punitive damages regardless of whether any person or entity was advised of the possibility of such damages.

This document is protected by AMTSO's intellectual property rights and may be additionally protected by the intellectual property rights of others.

Guidelines to False Positive Testing

Preamble

It is a very challenging problem to measure a security product's false positive rate and to further characterize the impact of this false positive rate on both consumers and enterprises, in relation to the product's overall efficacy. The purpose of this document is to point out the most significant issues that we identified during our investigation over the past year to help testers better mitigate these issues during future evaluations. We welcome any suggested solutions to the problems described.

There are different types of False Positives. For the purposes of this document, a false positive is a detection (or notification/alert) on a file or resource which has no malicious payload. There is another relevant area of False Positives regarding dynamic objects such as URL's. These are not addressed in this document, but will be addressed in a future AMTSO document.

Introduction

As most security companies know, False Positives (FPs) can have a larger impact on customers than a product's protection – and they are also remembered far longer. As more and more security products leverage proactive technologies such as behavior blocking, generic signatures and heuristics to address the expanded threat landscape, the likelihood of FPs has increased dramatically. In addition to harming the reputation of a product, false positives can disrupt operations within a business and cause financial distress to the affected software vendor. While significant FPs occur rarely, the consequences of such a significant false positive can far outweigh the consequences of a false negative.

It is striking that a number of tests that do not consider FPs. Even those tests that do evaluate false positives take a simplistic approach. Most testers simply scan a large collection of non-malicious files (often including grey-ware) and then report the number of non-malicious files that each product detected. For example: on the same set of clean files Product A falsely detects 100 files, while Product B falsely detects only 50, ergo Product B has a lower False Positive rate. QED. Why is this simplistic? Is this not the very definition of False Positive and False Positive rate? The problem (and, as we will see it is quite a complex problem) is that this presumes that all non-malicious files are equally important. But are they?

What Is a False Positive?

A false positive is a detection (or notification/alert) on a file or resource which has no malicious payload. Defining a malicious payload is not always clear cut. There are some gray areas such as Potentially Unwanted Applications, also known as Riskware. For example, a legitimate remote-access client (e.g., a VNC client) might be entirely legitimate if the user knowingly installs it. On the other hand, if a piece of malware surreptitiously installs that same VNC client to use it to obtain access to the victim's computer, such a program would be unwanted. A detection of such a VNC client in the former case would definitely

constitute a false positive, while detection in the latter case could be argued to be legitimate. Thus, the context of an application determines whether or not it is a false positive or not.

Additionally, some vendors opt to detect key generators or cracks that bypass software piracy checks. While these are not, strictly speaking, malicious, many corporate customers request that they be detected and removed.

How to Determine the Magnitude of a False Positive?

William Blackstone once said of the justice system, "Better that ten guilty persons escape than that one innocent suffer." The modern equivalent might be, "it is better that 10 malicious files run than that one non-malicious file is detected." But would the users agree? Given the growth in the threat landscape, users consistently demand better protection – and the only way security vendors know to deliver such improved protection is to deploy more proactive technologies (e.g., heuristics and behavior blocking), which are often subject to higher false positive rates than traditional signatures. While it is unlikely that a security vendor would ever knowingly detect a clean file, one of the costs of this increased protection is a higher chance of False Positives. There is a tradeoff to be weighed, and this is where the concept of Magnitude comes in.

There are a number of different criteria that need to be considered:

1.1 Criticality

We argue that it is important to determine the criticality of each false positive. Not all FPs have the same impact on the user experience.

We recommend that the industry segment false positives into the following categories when conducting a false positive test:

Ideally, the software industry should agree upon common metrics for each of these categories (in no particular order):

- **System critical**: includes false positives that render the computer unusable.
- **Network critical**: includes false positives that **preve**nt the computer from connecting to the network.
- **Browsing critical**: includes false positives that prevent the use of a web browser, limiting the user's access to reach the internet.
- **Business critical**: includes false positives on applications or data files which are critical to the operation of a business.
- **Core OS, non-critical**: includes false positives on core OS files, such as notepad, which are not required for the computer's basic operation.

- **Application critical**: includes false positives on 3rd-party applications that renders these applications unusable.
- **Application non-critical**[1]: this category of false positive leaves critical elements of an application functional, but with reduced ancillary functionality.
- **Data file/Non-executable critical**: this includes false positives on documents such as Word, Excel, PDF, and SWF.
- **Data file/Non-executable non-critical**: includes false positives on temporary files, caches, non-critical settings which don't impact the operation of the core OS or of system applications.

System Critical

System Critical files are those required for the system to boot up, the user to be able to log in, and still be functional. SVCHost or WinLogon are examples of System Critical files.

Network Critical

Network Critical files are those required for normal network connectivity, such as being able to browse the internet or process email. WinINet.dll is an example of a network critical file.

Browsing Critical

Browsing Critical files are those which are required in order to be able to browse the internet. While related to Network Critical, these are specific towards browsing. Firefox.exe is an example of Browsing Critical files.

Business Critical

Business Critical are those applications or data files which are important to business operations. A custom application used for production, or a PDF with important business information would be examples. These will be particularly difficult for a tester to test, as they would not generally be publicly available.

Core OS, Non-Critical

Core OS, non-critical are those files which are part of the operating system, but are not required to boot and login. Notepad or Calc would be examples of Core OS, non-critical files.

Application Critical

Application Critical are those files required for the operation of a given application. Word.exe is an example of an Application Critical file.

[1] Testers may group application non-critical and application critical FPs together for resource purposes as it can be very time consuming to differentiate.

Application Non-Critical

Application Non-Critical files are those files which belong to a specific application, but are not required for its basic operation. Various types of plugins are examples of Application Non-Critical files.

Data File/Non-Executable Critical

Data File/Non-executable Critical are those user files containing critical information, such as Word documents or mail archives.

Data File/Non-Executable Non-Critical

Data File/Non-executable Non-Critical files are those files which belong to the application, but are not critical to its functions. Caches or templates are examples of Data File/Non-executable Non-Critical files.

Browsing Non-Critical

Browsing Non-Critical files are those which are used by the browsers, but are not integral to its function. Temporary internet files or history URL's are examples.

1.2 Prevalence of an Object

Next to criticality, the prevalence of an object is an important measure to determine what the magnitude of a false positive is. How many users would be impacted by the FP? Those affecting thousands or millions of users are different than those that affect five.

The following should be taken into consideration:

- When possible, false positives should be ranked according to the prevalence of the impacted file; many security vendors now measure prevalence, so testers may wish to query vendors for this data, post-evaluation.
    - Many security products submit telemetric data to the vendor. This information should be shared with testers to better allow them to assess the prevalence of FPs. The tester should merge prevalence data from different vendors, since different vendors will have different data based on the size and makeup of their customer base (in most cases, these prevalence statistics are unlikely to overlap). This metric is still problematic, since some files are extremely prevalent yet a false positive on them would have literally no impact. For example, Windows 7 might include a legacy hard disk driver for a long-defunct model of hardware. Even if no users in the Windows 7 user-base used this file, its prevalence would be counted in the tens of millions).
- Prevalence statistics from popular download portals may be used to corroborate prevalence, but should be vetted first. Popular download portals, like ***download.com***, often track the prevalence of hosted downloads. It is important to check with the portal before using these statistics, however; in some cases, the prevalence counts are cumulative (for all versions of an application, rather than the

currently posted version the application). For example, ***download.com*** might state that GraphEdit has 1M (cumulative) users, whereas, in fact, only 10 users have downloaded the latest version of the application. A false positive on the latest version of the application would therefore only impact 10 users, whereas it would appear that such a false positive is impacting a million users.

1.3 Recoverability

The ramifications of a False Positive are not always the same. For example, having to download a file again from a website might be annoying, but it is quite different from having to use an off-line recovery tool to repair machines that no longer boot.

The following should be taken into consideration when rating the recoverability of a False Positive:

- **Permanent destruction:** Is the data irreparable, such as the loss of a document or a photograph?
- **Off-line recovery**: Does the system have to be taken offline in order to recover?
- **Recovery from product quarantine/backup: can the file/data be recovered from the product's quarantine**
    - Including centralized admin recovery: Does this recovery require an administrator to physically access the machine, or can it be recovered remotely?
- **Web site/download**: Can the user download the data again?

1.4 Environment

Testers should take into consideration the intended purpose of the products they are testing. For instance, perimeter defense solutions (such as mail gateways) may have much looser heuristics than desktop solutions. In these instances the False Positive is more a Denial of Service than a true loss of data or an impact on operations. As such, the impact is generally also much less severe.

The following considerations should be made:

- Policy detections vs. core protection detections: If a core protection technology (in its default settings) encounters a false positive, this is different than a false positive due to an administrator-configured blocking policy (which may be intended to block more than just malware).
- How significant is the impact of a false positive: Incorrectly detecting and blocking a legitimate svchost.exe file in email is not nearly as bad as blocking that same critical file on the desktop.

Policy Detections

Detections or blocks which occur as a result of policy should be separated from those which occur by signature. For example, many email clients prevent the user from accessing attachments which are executable. It could be argued that this is 100% False Positive rate. The difference is the user has selected this policy himself and it was his choice.

Unlikely Scenarios

Reviewers should take into consideration the conditions under which a False Positive occurs, and whether that condition is likely to happen. For example, a security solution which detects an operating system component on an email gateway (presuming it does not detect it on the machine). It is unlikely that a gateway would ever naturally see such files, so such detections should be discounted.

1.5 Response Time

A tester needs to consider also factoring in the amount of time it took a vendor to fix a particular false positive. Vendors tend to very quickly respond to major false positives. Most also have a mechanism for customers to report potential false positives. In assessing the effectiveness of a security solution it would be useful to measure how quickly the vendor responds to reported false positives.

1.6 Product Context

Many products have different modes of operation. These so called "paranoid" modes can often be activated by user selection. When the user selects this mode they are making a conscious choice to increase protection at the greater risk of false positives. Since fewer users would choose to use such a mode, false positives detected in this mode should be rated as less severe – even if on prevalent files. Criticality should be treated the same.

1.7 Other Considerations

Here are a few other considerations which do not fit neatly into the above mentioned categories. First, FPs often come with higher detection rates. Correlating True Positive (TP) and FP ratios can provide a more accurate reflection of the efficacy of a security solution.

Tester should take into consideration the version of the program. If an anti-malware product experiences a false positive on v1.7 of a program yet v1.9 is the latest version (and presuming the product in question does not yield a false positive on v1.9), then this should be reported. Of course, just because there is a later version of a program available does not mean that the earlier version is not in use (and in some cases can be more prevalent than the later version).

Measuring False Positives

Having a false positive on a system-critical file is much worse than on a regular file or resource.

Ideally, there should be a non-linear scale of sorts to rate FPs based on critically, prevalence, and recoverability.

To give an analogy: 'Falsing' on highly-critical system files should be viewed in a similar way as missing files from the WildCore.

Additional consideration should be given to the following when doing this testing:

- Check if the detection itself may actually be valid. This specifically applies to RiskWare/PUAs such as mIRC. The same care must be given to confirming the legitimacy of a "clean" sample as is given to a "malicious" one. There have been a number of instances where "clean" files have been infected prior to signing and releasing. The old axiom still holds: trust but verify.
- When dealing with AdWare/RiskWare detections, make sure that detected files are not misclassified. For instance, if the file is detected as AdWare then it is not a false detection. However, if the file is detected as a virus or Trojan then that would be a false detection.
- The vendor should be contacted to make sure that detection was not added intentionally. Vendors do not have uniform policies – particularly regarding "greyware" applications.
- Some vendors may employ contextual policies. For example, the product may not block tftp running as tftp.exe from the windows directory, but might block the same program running as sldjfsjl.exe from the temporary file directory) . Additionally, filename and folder name can both be separate contributing factors in contextual detections).
- A history of a file may play a role – e.g. files installed from CD-ROMs may be treated with a less suspicion than files from the Internet or USB drives.

Telemetry

Many security vendors have the ability to collect telemetry from their customers. This can include files, URL's, hashes, and events. This data can be extremely useful in assessing the prevalence of files within a product's user base. While this data can be strategically important to the vendor, sharing some of it with testers can help determine the files' prevalence.

Ideally the telemetry shared with a tester includes the following:

- Freshness of file (when was it first seen, when was it last seen?)
- Prevalence of the file (how many customer machines is the file on?)
- Breakdown of prevalence per region (where are these machines located?)
    - Ideally the tester would group FPs into countries of origin. For instance, a product may have an FP on programs created in China. This is important.
- Origin of distribution. Does it come with the operating system (or some other very popular application), or is it a specialized utility?

How to Perform False Positive Testing?

Ideally FP testing is performed in a similar fashion to dynamic testing. A stream of fresh clean files should be used to more accurately test FP efficacy. This is because vendors tend to whitelist prevalent clean files quickly, so delays in testing can yield misleading results.

1.1 Static Testing

While AMTSO does not advocate Static Testing, we recognize that these tests will continue to be performed. Given that, there are some basic rules which should be applied:

- Use fresh files that are likely to be in use by real users
- Context is important. Products are built to protect customers, and some of that involves identifying situations which deviate from the norm. In "normal" situations files have usual names/locations. Additionally, "normal" systems do not have millions of malicious samples.
    - Test False Positives specifically. Clean systems should be used for testing False Positives.
    - Use files in their "natural" location and name. Similar to above, clean systems should be used.

1.2 Dynamic Testing/Whole Product Testing

Some testers may opt to test for false positives in the same test where they are doing detection testing (i.e. they may intersperse 1000 legitimate files among 10,000 bad files to check for false positives and/or "gaming" of the testing methodology). This is a reasonable approach; however, explicit note should be made of this. Keep in mind that performing FP testing in combination may lead to different results compared to performing individual FP testing. This can be due to a product perhaps switching automatically to a more paranoid mode when malware is detected entering the system (in these cases the product must behave the same in both the real world and the testing environment).

Additionally, there might be a "guilt by association" tendency of some products. If a malicious application also drops some non-malicious files (such as tftp) these might also be detected and removed. This specific context needs to be noted such that the reader can make their own determination as to the usefulness or problem with this approach.

Some security products will take into account the name and location of certain applications in an attempt to discover malicious intent – they conditionally detect based on the context of the detection. When a tester has a directory full of clean files, perhaps named as their hash value, the security product might flag this as the application being in the wrong place or under the wrong name.

For notifications vs. detections the same rules should be maintained between TP and FP testing. If a prompting dialog is presented it must be answered the same regardless of FP or TP testing. This can be complicated by some products which might provide contextual information in order to elicit a more correct response from the user – but how to decide? One way is to capture a number of dialogs and use them to conduct a poll of a number of "typical" users to determine how they would answer those prompts[2].

Artificial Test Scenarios

[2] See AMTSO *Best Practices for Validation of Samples* at www.amtso.org.

Similar to creating new malicious programs for testing[3], creating new programs for false positive testing has been considered. However, such artificial scenarios should not be employed. The test should reflect real life scenarios. For further explanation see the *Issues Involved in the "Creation" of Samples for Testing* document.

Other Considerations for False Positive Testing

Lastly, there are a number of other considerations the tester should account for:

Testing with Other Security Products

In general testers should avoid scanning competing anti-malware products to see if False Positives occur.

One of the main reasons for this being that a scanner may detect the databases of the competitor. This is an edge case with a complicated scenario from both products' perspectives. Moreover, in the case that one product does not allow coexistence with other security products, this becomes an "artificial" test scenario. These should best be reported to all concerned and not included in a test.

Corrupted, Disinfected, or Modified Files

Testers should refrain from having corrupted, (incorrectly) disinfected or otherwise modified files in their FP test set. One exception to this would under dynamic or whole product testing. Here the product may encounter False Positives on incomplete files (for instance when the browser is downloading a file). In such cases the tester should treat the detection as an FP.

Potentially Unwanted Programs (PUPs)/Riskware

Different vendors may have different policies regarding PUPs or Riskware (useful programs that can – and are – used by malware for nefarious purposes). If such programs are going to be tested, this should be specifically identified and the samples properly verified. This way the reader can make their own evaluation as to how important these detections (or non-detections) are.

Non-Viral Detections

When a detection occurs, the classification of that detection is important. For example, a ServU sample could be detected either as not-a-virus:Riskware.ServU.501 or Trojan.agent.blabla. The first is not a False Positive, it is a correct classification. The second is a False Positive.

Case Study: AV-Comparative's Vendor Experiment

In March of 2010, AV Comparatives conducted an experiment with several security companies to determine if it was practical to use the prevalence and criticality information provided by the vendors to assess the impact of a False Positive. The results were quite interesting.

[3] See AMTSO *Issues Involved in the "Creation" of Samples For Testing* at www.amtso.org.

The following email was sent to all the AMTSO members by Andreas Clementi of AV Comparatives:

All,

At the AMTSO members meeting in Santa Clara I proposed a challenge to the Security Vendors to test the efficacy of some of the proposals for qualifying False Positives. It has been asked that testers classify FPs by prevalence and importance. However, this may be easier said than done. To find out how feasible this will be I propose the following challenge:

Part 1: You will be provided the MD5/SHA1/SHA256 of 11 files. You are to determine the prevalence of these files to assess that portion of the importance ranking.

Part 2: You are to assess the importance of these files to either the Operating System or the Application to which they belong. You should report your results back no later than 20th March 2010 (if you do not answer by then, it will be assumed that you do not take part into this experiment). Your report should include the following:

- Your classification of the importance of this FP (based on Parts 1 & 2 above)
- The number of man hours required to obtain the data for all 11 samples.

Rules:

- Do not consult with other vendors regarding these samples (judge independently)
- Measure the resources taken to perform the classification
- Send your results to xxxxxxxxx@xxxx.xxx (DO NOT POST YOUR RESULTS TO THE AMTSO MEMBERS LIST!)

Good luck! And let the experiment begin!

MD5

7bd87ca2644d39fbec5cf98baaa42b5db5d963ff2e09514256b9a4e6b9e
ea6e8c3510870130e140843513208c7a0e199407b31366865462ac20d42
989c7f03cf234530ef053c83ce40aa14f440a1ac91d5ad0d6fbd0a992ad
164d9b10c4bb4d31f559cbfe3476d1f2d6ff5cec2d0bfe25cbc5f0f69d5
2dff07cd6d93ef1820f4da70de606de15ae140c3e7d444509550e7ea288
cd0567e78fa98cc27495e427319c173f4f57daa49e57fd483be193db3

SHA1

394dc6c68e46e05247453c93fc9f3f24b144bd56d9b9c4d2e8a7bbccb8e
50f9ab0b5659e047cf49f009a49c4a6255e9d7a3f7ffaf2ab482ea9732d
0d16a5215853099febb53a0784367d4808a8100f7644d8babb1b3e70ea2
3e24beec967b67d15a968ece07c94ba6efc493d3f383f661a76e3e5924b
c846895d20829e664ef474ac21c769b9b74e130a7ca35ceb7a41cbc1e1c

```
f7451ac7b4d2820c2254d14ebc2f5580d62acc63e3986ede88397ff8f89
d3f4d2ff8f79f7647aaaa06e7d30a8400f5b710171d685
bbcf1d633c2ad645b41d841ee483b89508946e1a

SHA256
ae440c5b00fbd5ea63d3837021cd703beee3289faba7ecb3343c0edc684
8186491f95e504232ca78fad5344ea581164be5162f07be66588c1fe59b
2e4df913ec959b22856900beda29135b23c70db6761e5d44273ee2fd8b6
6d4f3e1d2449535daf7883556604fed26de460597775d20f8b133c579b5
d318ed08f5c7bcac023ba7d444ce2db227145a2672bc9087475b16ab1e1
fca22a1366e7d434f572d6afc097d061ffda39c69612cf0102698d588cd
92bbe70060c85be39e0223722a8151195921694a68a154ebfc7…

Regards,
Andreas
```

Seven vendors took part in this FP experiment. AV-Comparatives was asked to keep the vendor names confidential. Additionally, AV-Comparatives had access to four of the vendor's clouds to assess prevalence, and those results are included for each test as Cloud A, B, C, and D. This result also contains some of Andreas Clementi's personal opinions/comments.

The 11 files were all PE files. As most clouds are only able to provide useful data for PE files, it may be even more difficult to get useful data for FPs on non-PE files. Cloud data may vary according to user bases. Some clouds do not collect data on known digitally signed files. Not all vendors have clouds etc. on which they could base their decision on, and one vendor used also Google hits as one indication of prevalence and further findings.

Time to Analyze

Here are the results of the time spent (in man hours) by each vendor to research the 11 hashes:

Vendor 1	20 minutes
Vendor 2	30 minutes (only basic data; high level info would take several man-hours)
Vendor 3	35 minutes
Vendor 4	2-3 hours
Vendor 5	2 hours
Vendor 6	30 minutes
Vendor 7	5-6 hours

SAMPLE 1

Name: AreaBluetooth (Proximity Marketing Tool)
URL: http://www.areabluetooth.com
Filename: ABSend.exe

MD5: 7bd87ca2644d39fbec5cf98baaa42b5d
SHA1: 394dc6c68e46e05247453c93fc9f3f24b144bd56
SHA256: ae440c5b00fbd5ea63d3837021cd703beee3289faba7ecb3343c0edc68481864

User's by Cloud[4]	Count	Correct
Cloud A	around 50	✔
Cloud B	around 20	✔
Cloud C	4	✔
Cloud D[5]	2 (in last 10 days)	✔

Download/Sales Stats: Around 8000

Program Description: "AreaBluetooth is a point to point transmission system that relays on the Bluetooth universal protocol. When a bluetooth enabled device (mobile phone, PDA, computer, etc) enters the coverage area, the systems analyzes if there are available contents for the device and then prompts the user for authorization before sending your campaign media files."

Notes: if the software is not registered (shareware), it works only in 30 minutes intervals and sends a banner together with the sent campaigns. This does not happen when the software is registered. Some vendors may therefore consider this program as "Adware"; we do not. It was initially a false alarm of vendor xy, due which it got later detected as malware by several vendors (while vendor xy fixed the FP in the meantime).

Prevalence/Importance[6] According to Vendors:

Vendor	Determination	Correct Prevalence	Correct Importance
1	low prevalence -	✔	✔
2	very low prevalence -	✔	✔
3	very low prevalence low importance	✔	✔
4	very low prevalence low importance	✔	✔
5	low prevalence low importance (adware)	✔	✔
6	low prevalence low importance	✔	✔
7	very low prevalence -	✔	✔

Conclusion FP #1: data is congruent and correct.

[4] All clouds are based on the product's user base. Some clouds primarily measure objects that are actively running.
[5] Unique occurrences of the files among their users over around 10 days. The numbers are difficult to interpret in their absolute numbers, that's why the vendor normalized the data to reasonable orders to make them comparable (the real numbers are higher, but pretty much on the same orders; the ratios are maintained). For example, latest IrfanView version would have 72538 occurrences, Firefox 196668, Skype 257501, etc.
[6] Not all vendors were able to rate the importance of the files.

SAMPLE 2

Name: Brockhaus Multimedial Premium
URL: http://www.brockhaus.de
Filename: cdcops.dll
MD5: b5d963ff2e09514256b9a4e6b9eea6e8
SHA1: d9b9c4d2e8a7bbccb8e50f9ab0b5659e047cf49f
SHA256: 91f95e504232ca78fad5344ea581164be5162f07be66588c1fe59b2e4df913ec

Users by Cloud	Count	Correct
Cloud A	several thousands	✔
Cloud B	around 4000	✔
Cloud C	0	
Cloud D	947 (in last 10 days)	✔

Download/Sales Stats: over 100000

Notes: If cdcops.dll gets quarantined, program is unusable. If user tried to run the program without the file, even if quarantined file gets then restored, program remains unusable until the user registers the program again with the serial number provided (which I do not find here anymore, which means I lost €100 due this FP experiment :P).

Prevalence/Importance According to Vendors:

Vendor	Determination	Correct Prevalence	Correct Importance
1	(unknown)[7] very low prevalence -		
2	medium prevalence -	✔	
3	(unknown) -		
4	high prevalence medium-to-low importance	✔	
5	(unknown) very low prevalence very low importance		
6	(unknown) -		
7	low prevalence -		

Conclusion FP #2: Data is **NOT** congruent

SAMPLE 3

Name: Eulalyzer
URL: http://www.javacoolsoftware.com
Filename: eulalyzer.exe
MD5: c3510870130e140843513208c7a0e199
SHA1: 009a49c4a6255e9d7a3f7ffaf2ab482ea9732d0d
SHA256: 959b22856900beda29135b23c70db6761e5d44273ee2fd8b66d4f3e1d2449535

[7] For some vendors "unknown/no data" from their cloud corresponds to "zero or very low prevalence"

Users by Cloud	Count	Correct
Cloud A	several hundreds	✔
Cloud B	around 150	✔
Cloud C	0	
Cloud D	88 (in last 10 days)	✔

Download/Sales Stats: Unknown, but probably 50000 (apparently 240000 downloads on Majorgeeks - distributed/promoted in many magazines)

Program Description: Eulalyzer analyzes license agreements for interesting words and phrases.

Notes: Why download stats are not always a good indicator of prevalence – See the download stats of Eulalyzer according to the following download portals:

Majorgeeks: 239851 (Majorgeeks is one of the main download hosts for Eulalyzer)
CNET: 27212
PCWorld: 11108
Scanwith: 1689
Softpedia: 1231
Freewarefiles: 800
Betanews: 630
Datanews: 251 downloads

Language-specific software may be more popular on some download-portals (depending on language/promoted countries/partnerships). Sometimes download sites aggregate download numbers for all versions of a particular piece of software. Also, not everyone who downloads the installer actually installs it. Sometimes people download and install the software, but later on, uninstall it. So, at any time, the actual number of people using the software will be less than what is reported on the download site.

Prevalence/Importance According to Vendors:

Vendor	Determination	Correct Prevalence	Correct Importance
1	very high prevalence high importance	✔	✔
2	very low prevalence -		
3	medium-to-low prevalence medium importance	✔	
4	low prevalence low importance, application critical		✔
5	(unknown) very low prevalence very low importance		
6	low prevalence low importance		
7	very low prevalence -		

Conclusion FP #3: Data is **NOT** congruent

SAMPLE 4

Name: PC Kaufmann (Sage KHK Formulargestalter)
URL: http://www.business-software.at/pckaufmann.html
Filename: formed.exe
MD5: 407b31366865462ac20d42989c7f03cf
SHA1: 16a5215853099febb53a0784367d4808a8100f76
SHA256: daf7883556604fed26de460597775d20f8b133c579b5d318ed08f5c7bcac023b

Users by Cloud	Count	Correct
Cloud A	around 100	✔
Cloud B	around 20	✔
Cloud C	0	
Cloud D	0 (in last 10 days)	

Download/Sales Stats: Over 10000

Note: PC Kaufmann is one of the most well-known ERP systems for SMB in the German-speaking area.

Prevalence/Importance According to Vendors:

Vendor	Determination	Correct Prevalence	Correct Importance
1	very low prevalence -	✔	
2	very low prevalence -	✔	
3	(unknown) -	✔	
4	very low prevalence medium-to-low importance	✔	
5	(unknown) very low prevalence very low importance	✔	
6	very low prevalence very low importance	✔	
7	very low prevalence -	✔	

Conclusion FP #4: Prevalence OK, Importance **NOT OK**

SAMPLE 5

Name: IKEA Home Planner Furnish Pro
URL: http://www.ikea.com
Filename: Furnish.exe
MD5: 234530ef053c83ce40aa14f440a1ac91
SHA1: 44d8babb1b3e70ea23e24beec967b67d15a968ec
SHA256: a7d444ce2db227145a2672bc9087475b16ab1e1fca22a1366e7d434f572d6afc

Users by Cloud	Count	Correct
Cloud A	several hundreds of thousands	✔
Cloud B	around 45000	✔
Cloud C	7	

Cloud D	13035 (in last 10 days)	✔

Download/Sales Stats: Unknown, but supposed around one million

Prevalence/Importance According to Vendors:

Vendor	Determination	Correct Prevalence	Correct Importance
1	very high prevalence high importance	✔	✔
2	high prevalence -	✔	
3	(unknown) -		
4	high prevalence medium importance, application critical	✔	✔
5	low prevalence low importance		
6	low prevalence low importance		
7	high prevalence -	✔	

Conclusion FP #5: Data is **NOT** congruent

SAMPLE 6

Name: 3-WebToGo
URL: http://www.drei.at
Filename: InstallWTGService.exe
MD5: d5ad0d6fbd0a992ad164d9b10c4bb4d3
SHA1: e07c94ba6efc493d3f383f661a76e3e5924bc846
SHA256: 097d061ffda39c69612cf0102698d588cd92bbe70060c85be39e0223722a8151

Users by Cloud	Count	Correct
Cloud A	thousands	✔
Cloud B	0	
Cloud C	3	
Cloud D	50 (in last 10 days)	✔

Download/Sales Stats: Over 700000, but effective only about 150000 in use

Program Description: This program is required for mobile internet access thru mobile sticks. Some cloud products may not notice it on the USB stick as it is usually launched/accessed before an Internet connection is established.

Prevalence/Importance According to Vendors:

Vendor	Determination	Correct Prevalence	Correct Importance
1	very high prevalence very high importance	✔	✔
2	(unknown) -		
3	(unknown) -		
4	high prevalence medium-to-low importance	✔	

5	low prevalence low importance		
6	very low prevalence very low importance		
7	very low prevalence -		

Conclusion FP #6: Data is **NOT** congruent

SAMPLE 7

Name: Konica Minolta magicolor 2490/2590 MF Printer Driver
URL: http://www.konicaminolta.com
Filename: MSDMLT0B.DLL
MD5: 1f559cbfe3476d1f2d6ff5cec2d0bfe2
SHA1: 895d20829e664ef474ac21c769b9b74e130a7ca3
SHA256: 195921694a68a154ebfc763bd83df5e58bcb3726d5a92d4e6026570e6bc9d460

Users by Cloud	Count	Correct
Cloud A	around 50	
Cloud B	0	
Cloud C	0	
Cloud D	0 (in last 10 days)	

Download/Sales Stats: Over 100000

Prevalence/Importance According to Vendors:

Vendor	Determination	Correct Prevalence	Correct Importance
1	very low prevalence -	✔	
2	(unknown) -	✔	
3	(unknown) -	✔	
4	very low prevalence medium-to-low importance	✔	
5	(unknown) very low prevalence very low importance	✔	
6	(unknown) -	✔	
7	very low prevalence -	✔	

Conclusion FP #7: Data is almost congruent

SAMPLE 8

Name: Wood-Online Room Plan
URL: http://www.b2b-wood.eu
Filename: NETShop.exe
MD5: 5cbc5f0f69d52dff07cd6d93ef1820f4
SHA1: 5ceb7a41cbc1e1cf7451ac7b4d2820c2254d14eb
SHA256: 74cb3a3d328a032dac06f90bb1d2da9f541ae612e547ede7e05c3f42a671159e

Users by Cloud	Count	Correct
Cloud A	around 100	✔
Cloud B	0	✔
Cloud C	0	✔
Cloud D	4 (in last 10 days)	✔

Download/Sales Stats: Around 1000

Program Description: Wood-Shop Software (software used by business users to order for their customers).

Prevalence/Importance According to Vendors:

Vendor	Determination	Correct Prevalence	Correct Importance
1	low prevalence -	✔	
2	(unknown) -	✔	
3	(unknown) -	✔	
4	very low prevalence medium-to-low importance	✔	
5	(unknown) very low prevalence very low importance	✔	
6	(unknown) -	✔	
7	very low prevalence -	✔	

Conclusion FP #8: Prevalence OK, Importance **NOT OK**

SAMPLE 9

Name: Microsoft Windows Server 2008 RTM (Power Management Configuration Panel)
URL: http://www.microsoft.com
Filename: powercfg.cpl
MD5: da70de606de15ae140c3e7d444509550
SHA1: c2f5580d62acc63e3986ede88397ff8f89d3f4d2
SHA256: 37835d920afdf7f398b8f8a8a4675d1a77a73947a9963cfa65189eaede06fbb4

Users by Cloud	Count	Correct
Cloud A	several hundreds	
Cloud B	around 85000	✔
Cloud C	?	
Cloud D	56293 (in last 10 days)	✔

Download/Sales Stats: Unknown, probably several hundreds of thousands

Program Description: Microsoft Windows Server 2008 RTM (Power Management Configuration Panel)

Prevalence/Importance According to Vendors:

Vendor	Determination	Correct Prevalence	Correct Importance
1	very low prevalence -		
2	high prevalence -	✔	
3	very low prevalence very low importance		
4	very low prevalence high importance, OS non-critical		✔
5	low prevalence high importance		✔
6	low prevalence low importance		
7	high prevalence -	✔	

Conclusion FP #9: Data is **NOT** congruent

SAMPLE 10

Name: ESET SysInspector
URL: http://www.eset.com
Filename: SysInspector.exe
MD5: e7ea288cd0567e78fa98cc27495e4273
SHA1: ff8f79f7647aaaa06e7d30a8400f5b710171d685
SHA256: 4b4eb0c2dba139738e8806db17bfc0fab62a7ca3dcf8bd94c132cca450a5992c

Users by Cloud	Count	Correct
Cloud A	several hundreds	✔
Cloud B	around 1000	✔
Cloud C	?	
Cloud D	3469 (in last 10 days)	✔

Download/Sales Stats: Around 200000[8]

Program Description: System diagnostic tool for Windows systems.

Prevalence/Importance According to Vendors:

Vendor	Determination	Correct Prevalence	Correct Importance
1	very high prevalence -	✔	
2	low prevalence -		
3	medium-to-low prevalence high importance		
4	high prevalence medium importance	✔	
5	low prevalence medium importance		
6	low prevalence low importance		

[8] It is to be expected that components which are related to security products might show lower in the clouds of competing programs.

7	medium prevalence -		

Conclusion FP #10: Data is **NOT** congruent

SAMPLE 11

Name: Notebook Hardware Control
URL: http://www.pbus-167.com
Filename: uninst.exe
MD5: 19c173f4f57daa49e57fd483be193db3
SHA1: bbcf1d633c2ad645b41d841ee483b89508946e1a
SHA256: 39c041abfb944625a546683ec94927c05a41bc94b38897bdc2d6e9e192d946e0

Users by Cloud	Count	Correct
Cloud A	several thousands	✔
Cloud B	around 650	✔
Cloud C	2	
Cloud D	1227 (in last 10 days)	✔

Download/Sales Stats: This version is currently still used on about 13000 notebooks (about 3000 of them still using the paid product); distributed/promoted in several magazines.

Program Description: Uninstaller for NHC; Notebook Hardware Control allows to easily control the hardware components of Notebooks.

Notes: Considered as behaving suspicious by a vendor and suggested to do not use for FP testing by a vendor

Prevalence/Importance According to Vendors:

Vendor	Determination	Correct Prevalence	Correct Importance
1	very high prevalence high importance	✔	
2	low prevalence -		
3	low prevalence low importance		
4	high prevalence medium-to-low importance	✔	
5	low prevalence low importance		
6	(unknown) -		
7	medium prevalence -	✔	

Conclusion FP #11: Data is **NOT** congruent

Conclusions from the Experiment

How Do the Clouds Work?

There were four vendors who provided a tool to compute cloud prevalence during this test. Each of these clouds works a little differently. Some of the vendors asked that their description be anonymous. So, in absolutely random order, here is a description of the four clouds.

Cloud W

HTTP-based signature cloud-scanning for PE files that are non file-infectors, polymorphic nor script viruses.

We check both on-demand as well as on-access. However there are certain criteria we use before we check against the cloud, checking locally against whitelist, local signatures and local heuristics. Depending on the results from these local technologies, we will check against the cloud or not.

But we have both local whitelist (based on digital certificates for ex) as well as cloud-whitelist, so we do check many white files against the cloud as well, both on-access as well as on-demand.

Cloud X

Vendor of Cloud X made some sort of silent reporting for 10 days, so that every time the files were launched on the systems of their customers they got reported to them. Real numbers were higher, but they normalized them. E.g. installers would run only once and reported only once, ending with a much lower reputation.

Cloud Y

"Cloud reporting occurs on execution, opening, copying and is done during both on-demand and on-access scanning. Local white list works first and if it hits - this prevents reporting to the cloud.

Local white list is frequently and automatically updated - it excludes common clean files.

Therefore, due to local white list filtering cloud under-reporting for common items is expected."

Cloud Z

For users participating we submit the hash of all PE and MSI files which are executed or created on disk. (Note that files which are already present on the system and never execute will not be submitted.) Our prevalence values are an approximate range of the number of submitting, licensed, non-suspicious, unique users of a particular file.

One thing is clear from these descriptions: none of the clouds reports the presence of every PE file on disk. Most are limited to “active” ones, and even of those local whitelists will prevent accurate reporting. However, given the different approaches taken by each cloud, when taken in combination they should yield an overall fairly accurate picture of the prevalence of executables that are actually running in the world.

With that in mind, let’s look at how each Cloud did.

Cloud	Samples	Correct	Percentage
Cloud A	11	9	88.8%
Cloud B	11	9	88.8%
Cloud C	11	1	9.1%
Cloud D	11	9	88.8%

It appears that the various clouds are working quite well, particularly when taken with the view that they will only cover their customer base. So, how did the vendors do?

Vendor	Samples	Correct Prevalence	Percentage	Correct Importance	Percentage
1	11	9	88.8%	4	36.4%
2	11	7	63.6%	1	9.1%
3	11	5	45.5%	1	9.1%
4	11	9	88.8%	4	36.4%
5	11	4	36.4%	2	18.2%
6	11	4	36.4%	1	9.1%
7	11	7	63.6%	1	9.1%

Pulling it all together, it seems that the best avenue for testers to take is to use the cloud tools provided by the vendors, and to combine that with their own assessment of the importance.

This document was adopted by AMTSO on October 22, 2010