Sample Selection for 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.

Sample Selection for Testing

Introduction

The classification and appropriate, well-founded selection of samples for testing is necessary in order to make a test reliable, unbiased, relevant and meaningful. Following these practices properly lessens the risk of rendering testing and test results of doubtful validity and the conclusions based upon them are less likely to be misleading.

In any test, sample selection is important. In general, the quality of the samples used is more important than the quantity, but a reasonable minimum quantity of samples is necessary.

Sample selection can be broken down into the following processes:

- Collecting
- Validation
- Classification

Collecting of samples is the process of gathering/selecting files, URLs, or other objects to be used as test cases.

Validation of samples is the process of making sure that the file or object to be used functions properly in the defined testing environment.

Classification (or verification) is the process of properly categorizing the files or objects into their correct category set, which can be as simple as a good, bad or "gray" set, or as complex as worms, trojans, rootkits, adware, "potentially unwanted", or other more detailed categories. It may also include sub-categories within the good, bad or gray set, as described later.

By following these processes, and the best practices associated with each, any tester will have a good foundation for conducting a test. *Collect* the pieces; *validate* that they work; and *verify* them for accurate categorization.

Collecting

The *source* of samples to be used in a test really does often dictate the success or failure of a test. This is often one of the very first questions that a tester needs to ask. Focusing on a single, specific source may be acceptable as long as it was the specific purpose of the test and as long as it has been properly defined as the test objective. However, it can also result in the narrowing of the test coverage, which might limit the audience targeted by the review. For example, a consumer-oriented source of samples might not be of interest to or relevant to a corporate audience, and vice versa. The sourcing of samples should be aligned with and appropriate to the test purpose, but coverage of a wider range of sources often appeals to a correspondingly wider range of audiences and is recommended in principle.

Samples can be categorized from two different points of view:

1. How were the samples collected? Examples of this type of source categories include: honeypots, passive crawlers, active crawlers, ISP, etc.
2. Where have the samples been collected from? Examples of this type of source categories are: URI, intranet, email, file sharing, social networks, peer to peer, etc.

It's important to take both categories of sample source into consideration when the test is designed, conducted and tailored for the audience. Due to the high volumes, regional distribution factors, and the intended classification of existing samples, description of the collection methodology becomes a key factor in determining the likelihood and degree of bias in a given test.

The ideal source of samples offers real-world, prevalent, fresh, diverse samples collected independently of security software providers. It is important that testers actively collect samples and create their own sources/collections, so that the samples are as independent and neutral as possible. Obtaining samples from independent sources is discussed also in AMTSO's *Issues Involved in the 'Creation' of Samples for Testing* document at www.amtso.org.

Validation can be a problem, based on the resources typically available to testers, especially when using independent sources. However, at this point we reiterate that validation and verification of test samples by scanning with multiple products does not in itself offer reliable, accurate, vendor-neutral validation or verification.

If using samples drawn from the feeds of various AV companies, the selection must be done in a balanced way so that bias is not introduced even before the test is actually conducted. This is the area in which metadata sharing may come useful. Various attributes can be taken into account, such as malware (geo) prevalence, age, family name and so on. The list of significant attributes to share is under discussion within the IEEE ICSG working group.

Lastly, the freshness of collected samples is also important, since it affects how relevant a test set is to the real-life threat landscape. For example, a trojan discovered 5 years ago may still be as potent as trojans found today, but the likelihood of seeing such a 5-year-old threat might be low compared to threats just found today. In the case of short-lived threats, a one day old URL might already be obsolete.

Table 1 in the Appendix provides guidance for testers on the sources they might use and the pros and cons of each source, based on AMTSO good practice guidelines for sample collection.

This assessment is not meant to identify the best method of collection but merely to indicate the amount of post-collection effort a tester needs into put in building up appropriate and representative test sets. For example, when collecting from non-security industry sources, the independence and diversity gained is balanced by the increased post-collection effort needed to validate and verify the collected objects. When collecting from commercial sources, although the collected objects may be fresh and validated, diversity and independence may take a hit.

There is no single, ideal way to collect samples for tests. A tester needs to balance the factors mentioned here in order to build a good set of samples that can increase the quality of the test.

Validation

The sample validation process essentially consists of a series of tests to make sure that the sample is functional. There are several ways to validate samples: hand-checking, usage of automated tools (auto-replicating systems, sandboxing) or by using various specialized tools to check file geometry, integrity or functionality (not applicable to all sample types). Best practices show that validation is most valuable when it's based on sample functionality and performed in the same environment as the test will use. In this way, the validation can be also done during or after the test. However, the validation procedure still needs to be documented.

Merely scanning the samples using various products and accepting or rejecting according to the detection results cannot be considered an acceptable method of validation for several reasons:

- Vendors do not only use exact detections, so it is not guaranteed that a sample detected as malicious is really an intact or working sample
- Occasionally detections are created for samples that are known not to be valid, working objects
- Detections created for working samples may also detect non-working samples inadvertently (i.e. not on purpose), depending on the detection algorithm.
- Using AV products for sample verification can add a huge bias in the test, especially if the same products are going to be tested on those "verified" samples.
- AV products are known to occasionally follow each other's misclassifications (a.k.a. cascaded false positives)
- Using cloud scanners (for details refer to AMTSO's document *Best Practices for Testing In-the-Cloud Security Products*) should be avoided before the test is performed, since it may affect the test results by leaking information about the test set in advance.
- Using external multi-scanner services has all the problems listed above, and more: for example, it adds the risk of leaking the test set and losing control over the product settings.

AMTSO has already published acceptable validation methods and testers are advised to read the AMTSO document *Best Practices for Validation of Samples* for suggestions on how samples can be validated.

Classification

The classification process involves the categorization of the collected and validated sample set. This usually involves grouping samples as good (non-malicious), bad (malicious), or gray (whether the object is malicious depends on the intent of the author/distributor and the understanding of the target user – for example, whether the presentation of the object is unequivocally misleading), or as any other categories that are defined and which are intended to be included in the test. The intended categories need to be clearly stated in the documented test objectives.

Classification as good, bad or gray, can be further broken down into sub-categories depending on the tests that need to be performed. For example, malware can be broken down into trojans, worms, viruses, and so on, while clean files can be broken down based on prevalence or criticality. Although presented here as a separate step, classification may be performed at the time of validation. In this case,

the behavior of a file or object is observed and noted while checking whether it is working or not. Classification procedures also need to be documented and consistent.

The tester has to define the characteristics, the parameters and boundaries of what is considered to be good, bad, gray, or any other category. These definitions or definition references need to be documented. This is especially the case if they do not align to the generally accepted definitions (if they exist at all) for the mentioned categories. Lastly, the classification and/or categorization must be relevant for the purpose of the test.

Below are some practices used in verifying the sample's behavior and the questions a tester has to assess if this method is an option:

a. Reverse Engineering Verification of each sample.
    i. Would it be prohibited for testers to apply reverse engineering? At this point it is necessary to establish whether, for example, reverse engineering is prohibited by law.
    ii. Is it practical from a Time/Cost/Resource perspective?
b. Using Analysis tools
    i. Commercial Tools
        1. Are some of the tools prohibitively expensive?
        2. Does the tool provide the necessary functionality?
        3. Some malware detects commercial tools. Does this lessen their usefulness and eventually lead towards misclassification?
        4. Are the functionalities and/or limitations of the commercial tool known?
    ii. Open Source Tools
        1. Some malware detects open source tools. Does this lessen their usefulness and eventually lead towards misclassification?
        2. Does the open source tool provide the necessary functionality?
        3. Are the functionalities and/or limitations of the open source tool known?
        4. Has the open source tool been modified for the test? Some open source tools require the publication of the modifications.
    iii. Internally Developed Tools
        1. How much disclosure should be provided when using internally developed tools?
        2. Should it be explained why such a tool was developed?
c. Using Multiple Scanners (should not be used alone)

i. How many scanners have to concur for verification to be relevant?

ii. How does the choice of scanners used for verification affect the test? What measures have been taken to avoid bias in favour of any of the tested vendors? Will this information be disclosed?

iii. Does the detection name affect the classification of samples? What if the classification/names change over time? What about generic detections and multiple classifications?

iv. How reliable are the scanner results?

d. Using a Clean Collection

i. How was the clean collection collected/validated/classified? ii. How broadly has the clean collection been selected? Are commercial, shareware, and/or freeware applications included?

Other factors that testers should consider in the verification process are:

Freshness

An important aspect of any anti-threat or anti-theft technology is proactive protection. This is best evaluated using fresh and currently relevant threats. Thus the age of samples and/or the age of their sources (in case of URL, domains as test objects) need to be taken in consideration. Sample selection and categorization is a significant issue in all test methodologies, and to fully test the responsiveness of real-time systems, samples should normally be as 'fresh' as possible. Best practice would be to validate in advance; however, an acceptable compromise might be to show that maximum freshness can be achieved by testing solutions against all available samples and performing sample validation and/or classification later. In this case only success or failure against proven-valid samples should be taken into consideration when reporting results.

Prevalence

While making sure the samples in a test set are diverse and comprise a sufficiently large variety of files (either malicious or clean), it may – depending on the test scope – also be very important to take into account their prevalence. This is just as valid for malware as it is for clean files – the testers should avoid specific, low-spread problematic software (grayware) that's known to be likely to trigger false positives or disputed detection because of their nature. For both innocent and malicious samples, it should be taken into consideration that prevalence may depend on source and method of collection.

For example, if samples are sourced only from one geographical region, it is to be expected that these will be prevalent within the area in which they were collected, but that doesn't necessarily reflect prevalence worldwide.

Prevalence metadata (for example the model developed by IEEE ICSG working group members) could become a valuable source for determining sample prevalence. Vendors are encouraged to share metadata and testers are encouraged to use multiple sources in order to reduce the risk of bias.

Diversity

Diversity in this sense refers to both the variety of malware families tested and the underlying behavior of the malware. A sample set is diverse when it reflects the real world distribution of the samples relevant for the testing purpose. In particular, the resource-intensive tests (like dynamic or cleaning tests) are frequently carried out on a smaller sample set than large-scale static tests. It is thus more important that the sample set is diverse.

It is not best practice to include a large number of samples to reach some desired quantity if they are not diverse. Such an approach to "padding" the number of samples doesn't necessarily add any value; in fact, the more samples there are, the worse they're usually verified.

Diversity might be also limited in the special cases of testing detection capabilities regarding polymorphic viruses, server-side polymorphic malware, and so on.

Although diversity may lower the minimum quantity of statistically relevant number of samples in a test set, the higher the number the higher the test accuracy should be *as long as* the set is reasonably diverse and the quality of the samples is maintained.

Reasonable Number of Samples

As briefly touched upon in the previous paragraph, the number of samples used is very important, in order to make testing statistically meaningful. In reality, the number of samples tested is strongly dependent on the validation method and/or difficulties with conducting of the test, since the resources required vary with each test, even among tests from a specific tester. Tens of samples can hardly be considered statistically relevant for any test unless they represent a high proportion of a very small total population. The sample size and choice of samples should be statistically adequate to support the conclusions of the test. Where practical, the tester should quote the margin of error or at very least explain limitations of the test results imposed by methodology. One of the deciding factors is the statistical validity of the sample set.

Geo-location Issues

One of the important issues that testers need to take into account is the market coverage of the products tested. For example, detecting a legitimate Chinese toolbar may be a non-critical false positive in Germany, and not cause any problems there, but nevertheless be a false alarm. However, this same program would have a critical impact on products active in China. This applies mainly to legitimate software and grayware, where different acceptance rates for these applications can be observed in various regions around the globe.

With malicious samples geo-location is less important, since such programs are malicious regardless of world region and platform.

The tester might be strongly influenced by the region he resides in, and needs to be careful not to draw conclusions that are too generalized for the data, and beyond the geographical scope of the test samples/scenarios.

Reputation

Reputation information delivered by the vendors is assumed to be the same in all cases, but in reality this is often not the case. Depending on the business philosophy of each vendor, the reputation of specific grayware may be classified very differently by different products. This aspect should be taken into consideration either when classifying the sample, or when evaluating results, as well as when configuring software under test: certain examples of grayware may not be detected by a particular product *by default*.

Meeting the Objective of the Test

While gathering samples for the test it is necessary to focus on the purpose of the test and select the samples accordingly. Different testing scenarios require different types of samples. Considerations that need to be taken into account for specific types of test include:

- Static: no data files; in case of SFX bundled files it should be considered that the unpacking support for every product might be different and would thus influence the results. These files may not be a problem in dynamic tests/whole product tests where the security solution can intercept the malicious content while being unpacked.
- Polymorphic malware detection test: in this case the diversity of samples can be limited, and the numbers of samples belonging to the same family or variant rather high.
- Potentially unsafe/unwanted applications/adware/spyware tests: this is very subjective, being dependent on the opinions of the vendor and customer, and is different country to country as well; the testing of these categories of software is very sensitive and controversial and can be influenced by the tester's own opinions. Verifying that samples really fall into this category is extremely resource-intensive and the results could be brought into question. Detailed discussion can be found in the Considerations for Anti-Spyware Product Testing document by the Anti-Spyware Coalition (ASC).
- Dynamic/behavior/whole product: It needs to be confirmed that the samples used in the test exhibit malicious activity in such a way that a product being tested has an opportunity to block this malicious activity. The samples used must exhibit malicious behavior in ways that are reflected in the test methodology. Otherwise there is a risk that products will be penalized for failing to detect or block malware that it *would* have caught in the real world.
- This also means that the selection of the samples cannot be made by removing samples which are detected by a specific detection *method1* of the product to test another detection *method2* of the same product. This approach would be in a conflict with the product's architecture and design, and would not reflect real-world protection.
- Exploits prevention: The tester must make sure that the samples actually get to execute the exploitation code – so the system has to be vulnerable, the exploits match the platform and the environment is correctly configured to reflect real world attack and defense.
- URL blocking/web attacks prevention: validity of URL samples is very transient: a tester must ensure that the samples are valid at the time the URL is used as a test case. Because of the dynamic nature of threats in this form (geo, ttl, server-side poly, platform-specific, browser-specific, time-specific, "served only once") special care and considerations should be applied.

- In the cloud tests: it has to be taken into account that the detection of the sample can be altered by the test itself, for example the result might depend on the actual filename/path/attributes.
- Clean set tests/FP tests: To provide a balanced test of user experience, tests need to include looking for false positives by testing against clean applications. These applications should cover the set of common operations that users undertake on their machines, e.g. installing applications, updating applications, running applications, applying operating system patches and installing and using browser plugins. Installed applications should be run to ensure that they function correctly. Reputation of the benign samples can be taken into account and the clean sets should represent the real-world situation as much as possible (for further information can be found in AMTSO's *False Positive Testing Guidelines*).
- Cleaning tests: Given the high resource requirements of these tests, testers are not able to test against many samples, so sample prevalence is a critical factor, along with diversity and criticality.
- Unpacking tests/SFX tests: in this type of test it is usually difficult to collect a significant and diverse test set from the field. Testers should refer to AMTSO's *Issues Involved in the "Creation" of Samples for Testing* document for advice on acceptable practices when artificially generated samples are added to the test set.
- Performance tests: Should be generally performed on clean files rather than on malicious, but this depends on the methodology (for further details refer to AMTSO's *Performance Testing Guidelines* document).
- Targeted attacks: the target environment and the attack scenario have to be reconstructed properly, which is usually extremely difficult.

The classification process *does* entail a great deal of effort and thought: however, this is a prerequisite for sound testing.

Appendix

Sources	Security Vendor	Security Industry Research /Projects	Commercial Sources	Non-security industry sources	Tester Collected
Examples	AV Company	Security Working Groups	Sample feeds provided for a fee	ISPs, Universities	Honeypots, crawlers
Validation is performed	Should not be relied on	Should not be relied on	Should not be relied on	Should not be relied on	Should not be relied on
Freshness	May or may not be	Likely	Likely	Likely	Likely
Prevalence	May or may not be available	Likely	Unlikely	Unlikely	Unlikely
Diversity	Likely	Unlikely	Unlikely	Unlikely	Unlikely
Independence (not biased in favour of one or more vendors)	Highly unlikely	Unlikely	Unlikely	Likely	Likely

Table 1: Characteristics of Different Sample Sources

This document was adopted by AMTSO on February 24, 2012