

# Reference clustering

Paolo Milani Comparetti

May 28, 2010

## Abstract

This document describes the process we used to obtain a reference clustering from anti-virus labels to evaluate the malware clustering techniques presented in [3]. This text was included in an early version of [3], but it had to be discarded because of space limitations. Since this reference dataset has been made available and is being used by several researchers, it seems useful to provide detailed information on how it was obtained.

## 1 Reference Clustering

When an anti-virus program recognizes a malicious program, it assigns a name to this binary. Typically, virus labels are hierarchically structured. That is, each label has at least one part denoting the malware family and one part describing the particular variant. However, there is no standard naming convention. Each anti-virus vendor creates and assigns its own virus labels. This is why the same file is typically labeled differently by different vendors. Moreover, the granularity of virus labels varies between virus-scanners. McAfee and Grisoft, for example, have very general labels, where a single name covers a broad range of malware instances. McAfee also assigns labels such as ‘Generic BackDoor’ that have neither a variant nor a family part. Kaspersky, on the other hand, consistently uses labels such as ‘Email-Worm.Win32.Zhelatin.jz’, which allows for the straight forward extraction of a family and variant name.

It is possible to cluster a given set of malware samples based on the labels produced by an anti-virus program. However, as pointed out in [2], virus scanners are not particularly well-suited for clustering a given set. First, they

cannot cope with new malware, i.e., malware for which no signature exists. Second, their labels are frequently incorrect. Assigning a virus label to a new sample is a manual task that has to be performed by a human analyst at an anti-malware company. Because of time-constraints, the sheer mass of new samples appearing each day, the similarity of several malware families, and a lack of appropriate tools, human analysts frequently misjudge a sample's virus family and thus, assign a wrong label.

Despite the aforementioned problems, employing virus-scanners is currently the only way to automatically cluster malware samples. We have collected 14,212 unique samples (unique in terms of their MD5 checksum) in the period from October 27, 2007 to January 31, 2008. These samples were submitted to the ANUBIS online analysis service [1] and have been originally obtained by monitoring a wide variety of infection vector such as web infections, peer to peer, botnet monitoring, URL extraction from other malware analysis services, etc. We then scanned these samples with six different anti-virus programs, namely Ikarus, McAfee, Grisoft AVG, Avira, Kaspersky, and Bitdefender. The goal was to obtain an initial reference clustering, as described in the next paragraph. By using six different programs, we attempt to minimize the individual weakness of each program in clustering samples.

Six virus-scanners produce six different clusterings of the malware set. We would like to find the intersection of these six different clusterings, i.e., the subset of samples where the results of all virus scanners agree. To this end, we first generalize the virus labels by ignoring the variant specific portions of a label and by prepending the label with a token indicating the virus-scanner such as "kaspersky". This gives us a set of virus labels. Each (generalized) virus label can be defined in terms of the set of malware samples carrying that label. In a second step, we have to determine which virus names are similar. For example, we would like to find out that the labels "mcafee.rahack" and "kaspersky.net-worm.win32.allapple" are similar because these two names have been assigned to the same set of samples. To find the subsets of similar names, we perform clustering on the virus labels. For this, each virus label is represented by the set of malware samples to which it was assigned by AV engines. We use Jaccard Index as a distance function. We then simply perform single-linkage hierarchical clustering on the labels with a threshold of 0.5. The result of this clustering is a mapping between engine-specific virus labels and cluster identifiers that can be used as 'universal' virus labels.

Using these universal virus labels, we select samples for which a majority of AV engines provide the same label. More precisely, we include into the

reference dataset samples  $s$  such that (a) at least three virus scanners identify  $s$  as malware and, (b) at least  $2/3$  of the virus scanners that do identify  $s$  as malware provide the same label for it. The universal virus labels provide us with an initial clustering for this reference dataset.

Finally, we analyzed this set with with our dynamic analysis tool and based on a manual inspection of their analysis reports we corrected classification problems of the initial reference clustering.

## References

- [1] ANUBIS. <http://anubis.seclab.tuwien.ac.at>, 2008.
- [2] Michael Bailey, Jon Oberheide, Jon Andersen, Z. Morley Mao, Farnam Jahanian, and Jose Nazario. Automated classification and analysis of internet malware. In *Proceedings of the 10th International Symposium on Recent Advances in Intrusion Detection (RAID'07)*, September 2007.
- [3] Ulrich Bayer, Paolo Milani Comparetti, Christopher Kruegel, and Engin Kirda. Scalable, Behavior-Based Malware Clustering. In *16th Symp. on Network and Distributed System Security (NDSS)*, 2009.