ANTI-VIRUS TOOLS ANALYSIS USING DEEP WEB MALWARES

Igor Mishkovski[1], Sanja Šćepanović[2],
Miroslav Mirchev[1] and Sasho Gramatikov[1]

[1]University Ss. Cyril and Methodius, FCSE, Skopje, 1000, Macedonia
[2]Department of Computer Science, Aalto University, Finland

ABSTRACT

Knowledge about the strength of the anti-virus engines (i.e. tools) to detect malware files on the Deep web is important for people and companies to devise proper security polices and to choose the proper tool in order to be more secure. In this study, using malware file set crawled from the Deep web we detect similarities and possible groupings between plethora of anti-virus tools (AVTs) that exist on the market. Moreover, using graph theory, data science and visualization we find which of the existing AVTs has greater advantage in detecting malware over the other AVTs, in a sense that the AVT detects many unique. Finally, we propose a solution, for the given malware set, what is the best strategy for a company to defend against malwares if it uses a multi-scanning approach.

KEYWORDS

Malware, Community detection, Anti-virus engines, data science, multi-scanning approach.

1. INTRODUCTION

AntiVirus products are essential in every business deployment connected to the Internet. Nowadays, with the increase in the number and diversity of malware on the Web [1], there are more AntiVirus Tools (AVT) becoming available to protect users and/or companies from malware. However, the quarterly growth at around 12% for known unique malware samples, according the Intel Security Group's *McAfee Labs Threat Report: August 2015*, and the fact that some AntiVirus companies use the same or significantly similar AntiVirus engines leave us in some way vulnerable to the existing security threats.

Another factor that exposes even more users and companies to security threats is the Deep Web. The size of the indexed (surface) Web is currently estimated to 4.59 billion pages. At the same time, it is estimated that the non-indexed, Deep Web, is 400 or even 550 times larger [2] and rapidly expanding at a rate that cannot be quantified. The Deep Web besides offering to cyber-crime great business opportunity, hacking services, stolen credit cards and weapons, it also represents nest for malware. The hidden nature of Tor and other services means it is easy to host and hide malware controlling servers on the Deep Web. The malware from the Deep Web is not widely accessible and thus, these kind of files, coming from the Deep Web, are still not fully scanned for detection.

Thus, it is of crucial importance to everyone exposed on the Internet to know more details about the available AntiVirus Tools (AVT) on the market, their business and technical relations in terms

Natarajan Meghanathan et al. (Eds) : SPPR, SCAI, CSIA, WiMoA, ICCSEA, InWeS, NECO, GridCom - 2018
pp. 137–151, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81713

of similarity and possible groupings. In addition to this, for a better protection companies could use a multi-scanning approach, for instance use multiple antivirus engines on email gateways in order to enable a faster reaction to the most recent security threats by drastically shortening the time required to obtain the latest virus definitions and wider detection scope. Since many of the engines have their own heuristics and detection methods, in this way companies can gain maximum protection for their email environment.

In this work, using graph analysis and visualization methods, on one hand we empirically infer detection engine similarity and the existing groupings and/or overlapping between them, while on the other hand we infer which AVTs differentiate from the other AVTs and a have greater advantage in detecting malware compared to others. Moreover, using the AVT responses to our malware file set we optimize the combination of AVTs in order to obtain maximum detection rate (i.e. coverage). We strongly believe that this approach can be used by companies who want to implement a multi-scanning approach on their email gateways. The analysis is done on a malware file set provided by F-Secure and the AVTs responses on this file set obtained using the Virus Total API.

Researchers have undertaken evaluating and/or comparing the existing AVTs, some of them using malware samples and the VirusTotal service [3, 4]. We stress that in this work we are not trying to evaluate or compare the existing AVTs. Instead, we present results that undoubtedly show how our analysis can identify if some AVTs either use the same detection engine or quite similar engines between themselves and/or grouping between them. With simple graph analysis we can easily identify which AVT have a greater advantage, i.e. are unique compared to the others. Both results, with similarity and advantage, contribute to the multi-scanning approach in choosing the appropriate AVTs for a given price. We present this problem as a Mixed Integer Linear Programming (MILP) optimization problem and give an empirical solution. The solution shows that if a multi-scanning approach is to be implemented by a company, then the grouping according to the similarity and the advantage matters, besides the detection rate.

The work is organized as follows. In Section 2 we present some related work and then in Section 3 we present what are the novelties and the main contribution of our study. The dataset used for the study is described in Section 4. The results concerning similarity and communities between different AVTs are shown in Section 5, whereas Section 6 is dedicated for the uniqueness and coverage (multi-scanning approach). Section 7 concludes this work.

2. RELATED WORK

The analysis of decision from several Anti-Virus Tools has been addressed for several purposes over the last decade and mainly since the apparition of the VirusTotal service [4]. Submitting a set of known malicious files and performing quantitative comparison to deduce the best/worst AVT was the first purpose. Malware samples collected from Honeypot were submitted to VirusTotal to infer good and bad detection performance in [5]. The authors also explore if the combination of several AVT can improve protection and showed that AVT diversity and a combination of AVTs indeed improve detection without being able to reach 100% though. Similar empirical analysis using honeypot data [6] brought the same conclusions that diversity improves protection.

Our analysis brought similar conclusions while in contrast to previous work, the scale of the data analysed was orders of magnitude larger and considered files coming from the Deep Web, showing likely more diversity than honeypot data.

Previous work showed that detection performance comparison from different AVTs using VirusTotal is irrelevant due to the fast evolution of malware and AVT decision over time [7]. When it comes to the approaches taken to evaluate and compare the AVTs based on published malware samples, it is shown that creating a representative sample is a difficult task, especially nowadays since new malware samples are created on a daily basis [7]. In addition to that, malware creators are also finding ways to obfuscate existing malware with different type of techniques (such as bytecode conversion) to avoid signature-based detection. Hence, AVTs need to adapt to this type of malware detection (research suggests, for instance, using Opcode-sequences to detect malware [8]). In [9], findings on the stress test of AVTs with respect to such slight malware modifications is discussed.

Different AVT present inconsistency in labelling a given file as malicious or not and this label evolves over time. There is even more inconsistency between vendors in the correct identification of a malware family for a file while using different naming [10]. Mohaisen et al. [3] investigated inconsistency in malware family labelling of malicious files from different AVT and questioned the relevancy of using AV labels to build malware ground truth unless several tools are combined.

Typical method to build malware ground truth is to submit several unlabelled files to a set of anti-virus tools and consider the files malicious if at least "*k out of n*" AVTs detect it as a malware [1, 9]. Other approaches [11] have proposed to use anti-virus label decision over a set of files in a generative Bayesian model to improve ground truth composition.

In recent studies [6, 7, 13, 14] it is shown that the results have also temporal scale, i.e. AV regression exists, in a way that a given AVT can declare one file as a malware in a given instance of time, but later fail to recognize the file as a malware.

3. ANTI-VIRUS TOOLS STUDY

This paper presents a comparative analysis of several Anti-Virus Tools (AVT) based on a set of files coming from the Deep Web. In this study, we use a large dataset produced by crawling Web hosts through DNS brute force, hence containing potential malware files both from the Surface and the Deep Web. The resulting dataset consists of 1.64 Million files which were subjected to the VirusTotal API in order to get the decision from the plethora of AVTs on the maliciousness of these files. This work does not present a comparative performance analysis. It has been shown that the labelling of a given file can evolve overtime and performances per AVT for a given set of files are only valid at a given time [12]. Moreover, VirusTotal implements the command line interface of AVTs which is different from the desktop version that can implement more detection capabilities such as signature matching that could be bypassed in VirusTotal [7]. This could lead to an apparent performance degradation for a given AVT that is not actually true. Hence, the comparison of the detection rate against a given set of files cannot be performed using the VirusTotal interface and is out of the scope of this paper, which focuses on inferring AVT detection engine similarities and complementarity.

Given a set of files we seek to reveal several characteristics of AVT detection engine including:

- **Similarity**: The common detection capabilities two different AVTs present. Analysing the set of files detected by different AVTs we seek to infer the similarity in their detection engines operation. This analysis can infer as well *communities* of AVT having similar detection capabilities with community leaders presenting common characteristics from many community members. This can highlight the use of several third party engines in a single product.

- **Coverage**: Given a set of pieces of malware, infer which combination of AVTs can be used to optimize the protection against the largest number of malicious program in a multi-scanning approach. This involves analysing the complementarity of different AVTs detection regarding a given set of files in order to combine AVTs presenting different capabilities.

- **Advantage**: present the capability for one AVT to detect malicious files that other AVTs are not able to detect presenting its advantage over other tools.

This analysis displays the collaboration that different Anti-Virus companies have in developing their products through showing similarities in their detection capabilities. It denotes as well the use of a given AV engine in different tools. The competitive advantage that some products may have, by developing their own techniques, differentiates them for the competitors and underlines their usefulness in a multi-scanning approach. According to a recent survey [13] three main defence mechanisms against Web malware are presented: signature-based detection, code analysis of both client and server-side Web applications, and reputation-based URL blacklists. These defence mechanisms are differently used by different AVTs and thus, big corporations sometimes use a multi-scanning approach in order to protect their assets.

4. DATASET DESCRIPTION

The file set we use for the *similarity*, *coverage* and *advantage* study of existing Anti-Virus Tools (AVT) is crawled from the Deep and Surface Web from the company F-Secure, and consists of L = 1.64M files for each of which we have the file itself, its URI, its SHA1 hash value as a unique identifier. In this set, which we will refer as F-*Secure set*, there are $L = 990$ files, that were examined by F-Secure in details and were labelled as malicious files. We call this subset a *ground-truth set*. The complete *F-Secure set* is collected from 19 June 2015 till 12 October 2015.
In order to tackle the challenges described in details in Section 3, we have used the VirusTotal API [14], which is currently the largest freely available AVT service aimed to provide the users with results from different engines. The service enables the users to upload a file (or its unique hash) for a scan with a number of engines/tools supported by the service. As a final result, the user receives classification of the file as a malware or not by each of the AVTs, together with their own malware type label if the file is marked to be malicious. Thus, we have scanned both file sets (*F-secure* and *ground-truth*) using the Virus Total API where as an input we used the file's SHA1 value. We then, processed the JSON output from the Virus Total API obtaining the following additional information for each SHA1 value (i.e. file): $[AVT_1, Descr_1]$, $[AVT_2, Descr_2]$, ..., $[AVT_k, Descr_k]$, where AVT_i is the name of the AVT that labelled the file as malware, $Descr_i$ is the description of the type of the malware as reported by AVT_i (one example is *Win32: Trojan.Badur*, though there is not standardization between big anti-virus companies), and k is the number of AVTs that reported the file as a malware (some of which are: McAfee, Sophos, GData, VIPRE, Fortinet, Avast, Comodo, Symantec, ESET-NOD32, F-Secure, etc.). From the 1.64M files in the *F-Secure set* only 24.176 files were declared as a potential malware by at least two of the AVTs ($k \geq 2$). We call this set the *similarity set*. The *similarity set* is later used in Section 5 for the similarity and community analysis of the AVTs. The labelling using the VirusTotal API is done only once in May 2016.

However, in the *similarity set* there might be lot of files, which were erroneously declared as malware. The labelling of a file by an AVT as malign or benign evolves overtime. Benign files can further be declared as malicious because they belong to an unknown emerging malware family for which AVTs do not have any signature yet [7]. In contrast, benign files can be wrongly classified as malware (false positives) due to an overly broad detection signature or algorithm used in an anti-virus product. After a short period of time, vendors can be notified of the

mislabelling to correct the error or add an exception. This is likely to happen for newly developed program for instance. On the other hand, the *ground-truth set* contains only small fraction of the existing malware and thus, might impose the problem of under-sampling, leading to higher number of false negative errors. Thus, in order to tackle better the false positives and false negative errors we have chosen the threshold for the detection rate to be $k \geq 5$ as in [1], i.e., 5 or more AVTs must label a file as a potential malware. By thresholding the *F-secure set* we obtain new *malware set*, having $L = 10.745$ potential malware files, which will be used in the later analysis for AVT coverage and advantage over other AVTs, see Section 6. The cumulative distribution function CDF for the AVT detection rate on the *malware set* is shown in Fig. 1. We see that each file is detected by an average of 15.3 AVTs, with a median of 14 AVTs and standard deviation of 9.43 AVTs.

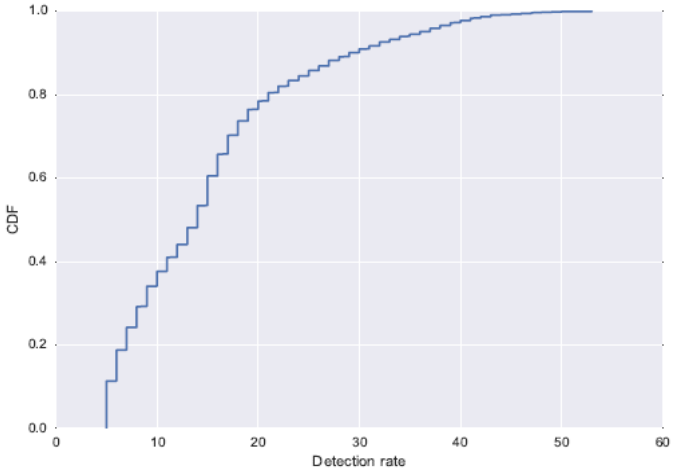

Figure 1. AVT detection CDF for the *malware set*.

5. AVTS SIMILARITY AND COMMUNITIES

Based on the similarity set described in Section 4 we measure the similarity between different AVTs and find existing grouping or communities that share similar decision regarding a given piece of malware. Thus, we first construct the similarity network $G^1 = (V, E, W^1)$ in order to characterize the similarity between different AVTs based on the shared files they label as malware. In order to get relevant results, we discarded from the analysis AVTs that detected less than 0.5% of the files from the *similarity set* i.e. less than 120 files. The node set V consists of the 61 AVTs that meet this condition, whereas the undirected edges set E contains the links between the AVTs that have labelled at least one common malicious file, with an edge weight $w_{ij}^1 \in W^1$ being defined as the Jaccard index between the sets of malware files detected by the two AVTs i and j. Next, we define the similarity between V_i and V_j as the co-occurrence strength. Let us assume that F_i and F_j denote two sets of files, labelled as malware by V_i and V_j, then we can define the Jaccard similarity measure (index) as a co-occurrence strength as follows.

$$sim(V_i, V_i) = \frac{|F_i \cap F_j|}{|F_i \cup F_j|} = w_{ij}^1 = w_{ji}^1, \quad (1)$$

where $|F|$ indicates the size of the set F. The value of w_{ij}^1 is between 0 and 1 (where "0" indicates no co-occurrence relationship between two AVTs and "1" indicates a full co-occurrence).

5.1. AVTS SIMILARITY RESULTS

The visualization plot of the adjacency matrix of the similarity network for the malware set is shown in Fig. 2. The similarity between each AVT is depicted as a square with darker colour. The results show high malware detection similarity between certain AVTs.

Some noticeable similarities are observed for McAfee and McAfee-GW-Edition with a similarity $w_{ij}^{1} = 0.78$. The same observation holds for K7AntiVirus and K7GW with $w_{ij}^{1} = 0.87$. This high similarity is to be expected between different tools coming from same vendors i.e. McAfee and K7 Computing. Yet, we see that different versions of tools i.e. standard and gateway editions, have different capabilities and that AV vendors do not use the same technologies in different products to maximize their detection capabilities, but rather propose tailored solutions for different applications. Gateway edition of these products are company solutions while other are basic customer version. Company solutions my implement more refined and customizable engine that explain this small dissimilarity.

While having comparatively high similarity score $w_{ij}^{1} = 0.71$, VIPRE and BluePex AvWare are developed by different vendors. After making searches we found out that BluePex AvWare actually uses VIPRE engine for malware detection explaining the high similarity in detected files. The conclusion is that detection engines integrated in third party solutions seem to be different than the one integrated in homemade product explaining a still significant dissimilarity (0.28) between these two tools. Observing the VIPRE line in Fig. 2 we see that it has quite high similarity with many AV tools e.g. Sophos, McAfee and Comodo, suggesting that this engine may be used in many other tools.

One AVT group showing high similarity is BitDefender, F-Secure, Emsisoft, MicroWorld-eScan and Ad-Aware with a similarity $w_{ij}^{1} > 0.6$ between these AVTs. Ad-Aware, F-Secure, Emsisoft and MicroWorld-eScan actually use BitDefender's detection engine along with other in-house detection solution, which explains the high similarity and small differences between all these tools. Globally BitDefender engine is largely used in several AVTs. G-data while embedding as well BitDefender engines shows less similarity than previously cited tools ($w_{ij}^{1} = 0.53$) suggesting that their in-house detection solution is more prominent than in other tools.

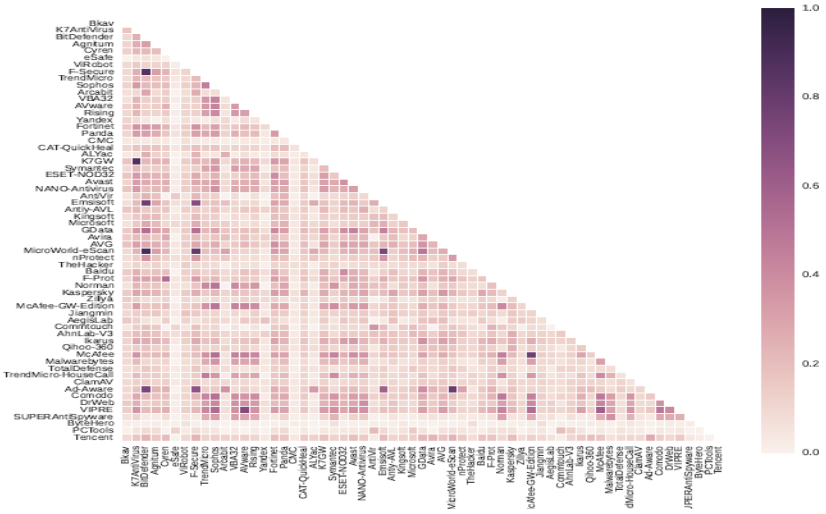

Figure 2. AVT's similarity for the *malware set*.

On the other hand, it is visible that some AVTs must have quite unique detection engine showing low similarity with any other tool with very light colour line in Fig. 2. Some examples are ByteHero with 1.611 files detected and $w_{ij}^1 < 0{:}07$ with any other tool, CMC with 1.439 detected files and $w_{ij}^1 < 0.08$, or Yandex with 639 files and $w_{ij}^1 < 0.19$. ByteHero is a self-developed unknown virus detection software that does not include virus database explaining their uniqueness in detection. Similarly, CMC anti-virus uses its own detection engine. Yandex anti-virus relies partly on Sophos for signature based detection ($w_{ij}^1 = 0.07$). However, our results seem to show that their proprietary anti-virus technology based on behavioural approach is prominent in their product.

5.2. AVTS COMMUNITIES

In this Subsection we detect structural communities, groups and/or modules in the AVT set using modularity-based community-detection algorithm [15]. The structural communities translate into groups of AVTs, which react in a similar manner to a certain malware. However, for a complete functional definition of the detected structural communities [19, 20] we have to know more details about the AVTs, including having an expert knowledge, and the AVTs response to different type of malware for different type of platforms. We underline, that this type of analysis is not part of this work, due to the restricted dataset and the fact that there is no existing effort between the AV companies to have a standardized malware labelling [3], thus, this approach may be used for future analysis.

The modularity-based community-detection algorithm is a simple heuristic method, which extracts community structures in networks, based on modularity optimization. The modularity Q is actually a scalar value (between -1 and 1), which measures the links density inside communities as compared to links between communities and is calculated as:

$$Q = \frac{1}{2m}\sum_{i,j}\left[W^1 - \frac{k_i k_j}{2m}\delta(c_i, c_j)\right], \quad (2)$$

where $k_i = \sum_j W^1$ is is the sum of the weighted degree of node i, c_i is the community to which the node i is assigned, the δ-function $\delta(u,v)$ is 1 if $u = v$, and 0 otherwise, and $m = \frac{1}{2}\sum_{ij} W^1$. In this work in order to find the optimal partitioning, i.e. optimize Q, we use the algorithm presented in [15].

In Fig. 3 we show that the algorithm partition the similarity network in 3 communities, with a highest modularity score of $Q = 0.12$, where each community is represented by a different colour, the size of the node (the AVT) is inversely proportional to its weighted degree ($k_i = \sum_j W^1$), and the width of the edge is proportional to the value w_{ij}^1 (see Eq. 1).

The biggest community, the one with the largest number of AVTs is the violet community, where some of the AVT with a biggest detection rate are Ikarus, ESET-NOD32, K7AntiVirus, K7GW, The Hacker and Baidu. Strong similarity in this community exists between K7GW and K7AntiVirus. The AVTs with the lowest similarity scores (i.e. lowest values for k_i) are ByteHero, CMC, Yandex and TheHacker.

In the orange community some of the AVTs with a highest detection rate are GData, F-Prot, Fortinet, Panda, Agnitium and F-Secure. Strong similarity ties exist between F-Secure, BitDefender, Emsisoft, MicroWorld-eScan and Ad-Aware as previously seen in Fig. 2 as well. BitDefender is used in several AVTs, thus its detected files are a subset of many AVT detected files presenting a smaller node with strong links to many other nodes. The AVTs with the lowest similarity scores are PCTools, eSafe, Commtouch and ALYac. Another remark is that in this

community there are old AVTs that are not maintained anymore. Symantec PCTools line was retired in 2013, VirusBuster was similarly closed in 2012. CommTouch anti-virus company became Cyren in 2014 and eSafe is not a product of Gemalto (previously SafeNet) anymore.

The rest of the AVTs are in the community with the highest detection rate, i.e. the green one. Here the leaders are Symantec, TrendMicro-HouseCall, McAfee, McAfee-GW-Edition, Rising and DrWeb. Strong similarity ties exist between McAfee and McAfee-GW-Edition (w_{ij}^{1} = 0.78), AVware and VIPRE (w_{ij}^{1} = 0.71), McAfee and VIPRE (w_{ij}^{1} = 0.60), Sophos and VIPRE (w_{ij}^{1} = 0.54). The AVTs with the lowest similarity scores are SUPERAntiSpyware, Malwarebytes, TrendMicro-HouseCall and Symantec.

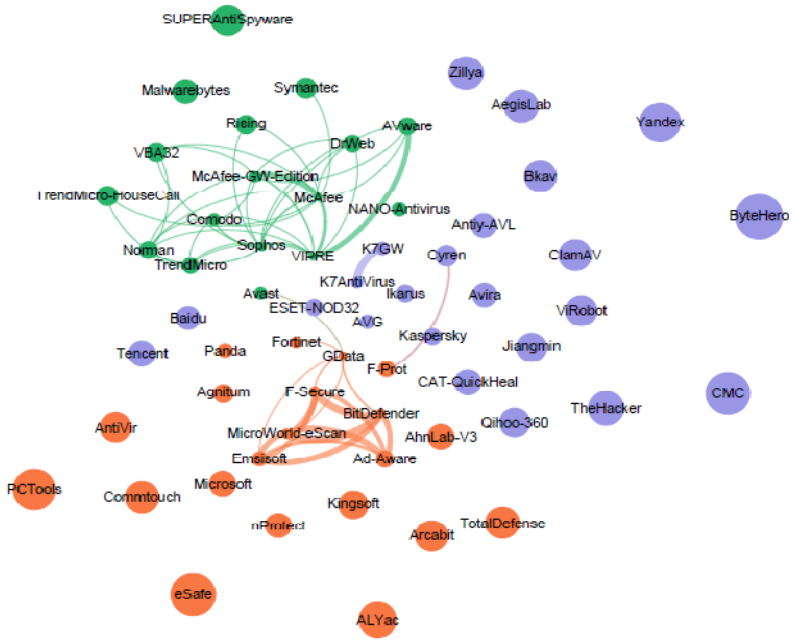

Figure 3. AVTs communities on the *similarity network* based on modularity-based community-detection algorithm. Edges that have Jaccard index below 0.4 are not shown.

6. AVTS COVERAGE AND ADVANTAGE

In order to analyse the AVTs coverage and the advantage of a given AVT compared to the others we define another measure as follows

$$c(V_i, V_j) = w_{i,j}^{2} = \begin{cases} |F_i|, & if\ i = j \\ |F_j \backslash F_i|, & otherwise, \end{cases} \tag{3}$$

when $i \neq j$, higher value of w_{ij}^{2} means that AVT j did found many diverse files as malware, compared to AVT i, i.e. this value defines the AVT advantage, whereas the self-loop weight w_{ii}^{2} shows how many files were found in the file set F_i, i.e. it defines the detection rate.

Now, let us construct a second network, which we call *coverage network* $G^2 = (V, E, W^2)$ in order to characterize the coverage and advantage of different AVTs based on the shared malware. Again, the node set V consists of AVTs that were reported by Virus Total and labelled at least 0.5% from the files in the malware set as malicious (N = 61), whereas the directed edges set E contains the links between the AVTs that have labelled at least one common malicious file with an edge weight $w_{ij}^{2} \in W^{2}$, and self-loops with a weight $w_{ii}^{2} \in W^{2}$.

In Fig. 4 we visualize the *coverage network* for the *malware set*, where the size of the nodes is the in-degree and the colour represents the out-degree. The bigger the node the more unique is its detected malware file set. In a similar way the red colour means lower out-degree, whereas blue means high value for the out-degree. Thus, the AVTs core is actually consisted of big red nodes represented in Fig. 4. Moreover, the colour of the edge represents the direction, i.e. the source AVT, and the width is the value of w_{ij}^2 (for e clearer visualization only the weights above 5.000 are shown). For instance, one can notice that McAfee has a lot of thick blue edges, i.e. incoming edges, which means that it has a great advantage over many AVTs.

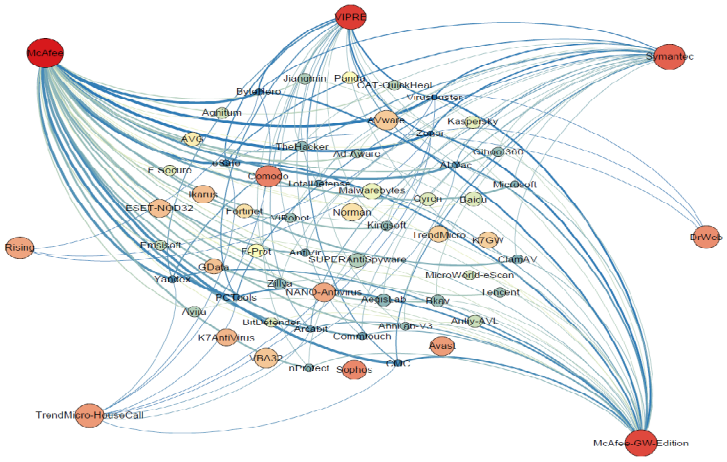

Figure 4. Coverage network for the malware set.

6.1. AVTS COVERAGE RESULTS

Without going in too much details, we observe the detection rate (w_{ii}^2) for the malware set and we find out that the AVTs showing "best" detection rate across all 10.745 files is McAfee, followed by McAfee-GW-Edition, VIPRE, Symantec, TrendMicro-HouseCall, DrWeb, Rising, Comodo, Sophos, etc. (see Fig. 5).

However, these results should not be taken too strict because if we increase the threshold from $k \geq 5$ to $k \geq 30$ then the "best" AVTs are GData and VIPRE, followed by McAfee, Sophos, Avast, Comodo, etc. (the plot is not shown). When the majority of the AVTs "votes" ($k \geq 30$) that a given file is a malware, there is no obvious winner among AVTs, though best results show GData, VIPRE and McAfee. The discrepancies in the results for different thresholds, bring us to one possible conclusion that some of the AVTs might report too many "false positives", i.e. they have a high malware detection rate when the rest of the AVTs disagree, or maybe they have a unique AV engine compared to the other AVTs.

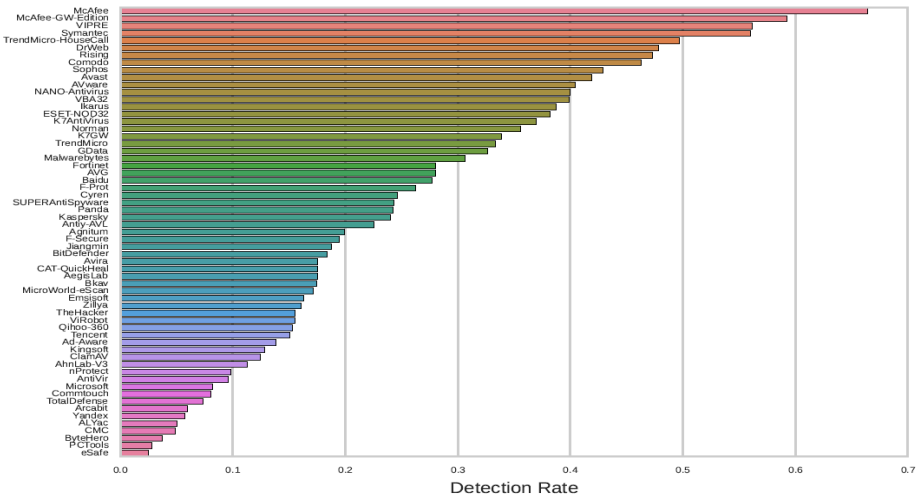

Figure 5 AVE Detection rate for the malware set.

The disagreement between AVTs comes from not having a common definition of what constitutes a malware [16]. For instance, adware can be considered as unwanted software or not by different AV products. As described in Section 4 as well, the labelling of a given file can evolve overtime and performances per AV for a given set of files are only valid at a given time. Finally, Virus Total implements the command line interface of AVTs, which is different from the desktop version that can implement more detection capabilities such as signature matching that could be bypassed in Virus Total [7]. This could lead to apparent performance degradation for a given AV program. Hence, a comparison of AVT detection rate against a given set of files cannot be performed using the VirusTotal interface and is out of the scope of this paper.

Instead, in the following using the *malware set* we focus more on optimizing the protection against malwares in a multi-scanning approach, i.e. ***find an optimal AVT set M, which will have the best malware detection coverage for a given price P***. This problem can be represented as a Mixed Integer Linear Programming (MILP) optimization problem, as following.

$$\begin{aligned} &\max \cup_{i=1}^{|M|} |F_i| \\ &s.t. \sum_{i=1}^{|M|} cost_i \leq P \end{aligned} \tag{4}$$

where $|M|$ is the number of AVTs in the optimal set M, $cost_i$ is the cost needed to buy AVT i and P are the available resources.

Using Eq.4 we show which AVTs to choose in order to have the highest coverage of the detected malware under given price constraint P. Due to the unknown price of the AV software we set **cost = $\mathbf{1}^T$** in Eq. 4. The best malware coverage, both for the *malware set* and the *ground-truth set*, as a function of the number of AVTs is shown in Fig. 6. The coverage follows logarithmic increase as a function of the number of AVTs. For instance, if a company would like to cover 95% of the labelled malware from the *ground-truth set* it would need four AVTs, and six for the *malware set*. In Tables 1 and 2 we give the names of the AVTs and the exact coverage obtained with them. In Table 1 we show the best coverage for the *malware set* for a given resource constraint P, where $P \in [1, 10]$. The best coverage when choosing 3 (three) AVTs is obtained by McAfee, ESET-NOD32 and Trend Micro-House Call with a total coverage of 87.6%.

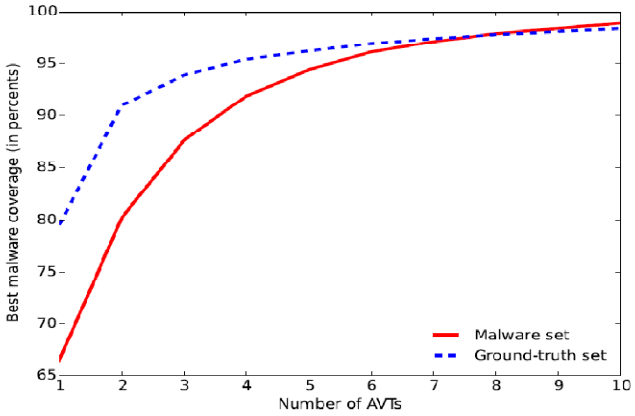

Figure 6 Maximizing malware coverage.

If we map the AVTs that provide best coverage to the community they belong, it is obvious that the best choice is to mix the AVT to be either from the orange or violet community shown in Fig. 3 and the last column in Table 1.

Table 1. Best coverage for a *malware set*

P	**AVT**	**Coverage (%)**	**Community (Color)**
1	McAfee	66.5	green
2	+ ESET-NOD32	80.2	violet
3	+ TrendMicro-HouseCall	87.6	green
4	+ Ikarus	91.9	violet
5	+ NANO-Antivirus	94.4	green
6	+ TheHacker	96.1	violet
7	+ Symantec	97.1	green
8	+ BKAV	97.9	violet
9	+ Antiy-AVL	98.4	violet
10	+ VIPRE	98.9	green

Finally, in Table 2 we show the best coverage for the *ground-truth set* for a given resource constraint *P*. The best coverage when choosing 3 (three) AVTs is obtained by McAfee, F-Secure and Ikarus with a total coverage of 93.9%. However, we must mention that these results are biased towards F-Secure because they have evaluated the *ground-truth set* of malwares.

Table 2. Best coverage for the *ground-truth set.*

P	**AVT**	**Coverage (%)**
1	McAfee	79.5
2	F-Secure + Ikarus	91.0
3	McAfee + F-Secure + Ikarus	93.9
4	+ Cyren	95.4
5	+ Symantec	96.2
6	+ Zillya	96.9
7	+ SUPERAntiSpyware	97.4
8	+ AegisLab	97.8
9	+ CAT-QuickHeal	98.1
10	+ Rising	98.4

6.2. AVTS ADVANTAGE RESULTS

Using the *coverage network* we can define the *advantage* of a given AVT by implementing a random walk on G^2. The directed network with self-loops G^2 consists of information about the *coverage* of AVT i, represented by the weight w_{ii}^2 showing how large is the covered file set F_i, and on the other hand we have the uniqueness of AVT i, compared to the other AVTs, represented by the weight w_{ji}^2. Thus, by running random walk on G^2 we should obtain information about the advantage of the AVT i which depends i) on its coverage (w_{ii}^2), i.e., how many files were detected by the AVT as a malware and ii) recursively on the weights of the incoming links from other AVTs $j \neq i$, i.e. how much unique is the AVT i.

Before presenting the results for the *malware set* and some results for the *ground-truth set*, let us show a simple scenario for the *coverage network*. Let us assume that there are 4 AVTs (x,y,w,z), i.e. $N = 4$, and there are $L = 13$ malware files, named {1, 2,… , 13}, and the following possible scenario. AVT x has detected $F_x = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ as malware, AVT y has detected $F_y = \{1, 3, 11\}$ as malware, AVT z has detected $F_z = \{1, 3, 7, 9, 12, 13\}$ as malware, and AVT w has detected $F_w = \{11, 12, 13\}$ as malware then using Eq. 3 we obtain

$$W^2 = \begin{bmatrix} 10 & 1 & 2 & 3 \\ 8 & 3 & 4 & 2 \\ 6 & 1 & 6 & 1 \\ 10 & 2 & 4 & 3 \end{bmatrix}. \tag{5}$$

In the case shown in Eq. 5 the stationary distribution vector equals to $\pi = [0.56, 0.08, 0.21, 0.15]$, showing that the AVT with the biggest advantage is x, whereas the least advantageous is y, even though it has the same weight self-loop as w, which is due to the fact that w has detected 3 new files that were not in F_x.

Using the weighted adjacency matrix $\mathbf{W}^2$ of the *coverage network*, and the random walk process we obtain different results for the AVT's advantage for the *malware set* and the *ground-truth set*. For instance, for the *malware set* the most advantageous AVT is McAfee, followed by McAfee-GW-Edition, Symantec, TrendMicro-HouseCall, Rising and DrWeb (see Fig. 7). For the *ground-truth set* the most advantageous AVT is again McAfee, this time followed by GData, Ikarus, F-Secure, VIPRE and Avast (see Fig. 8).

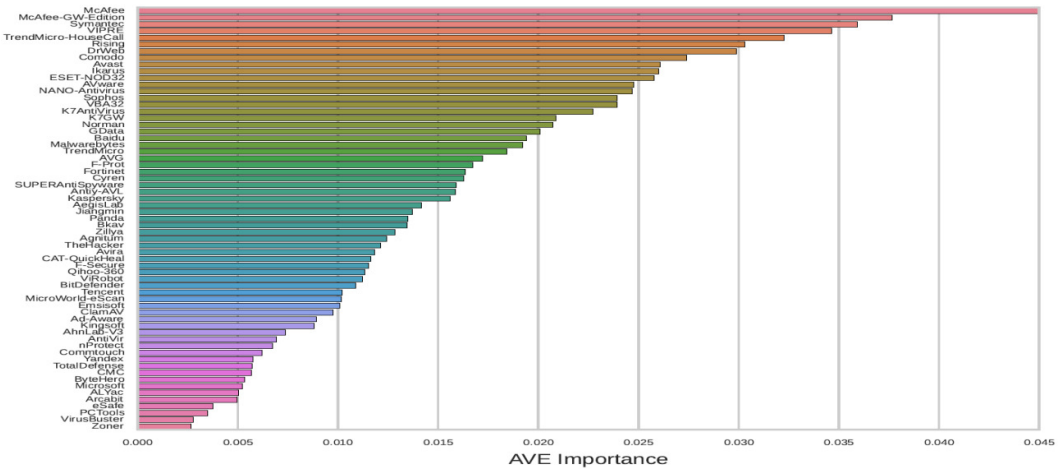

Figure 7. AVT advantage for the *malware set*.

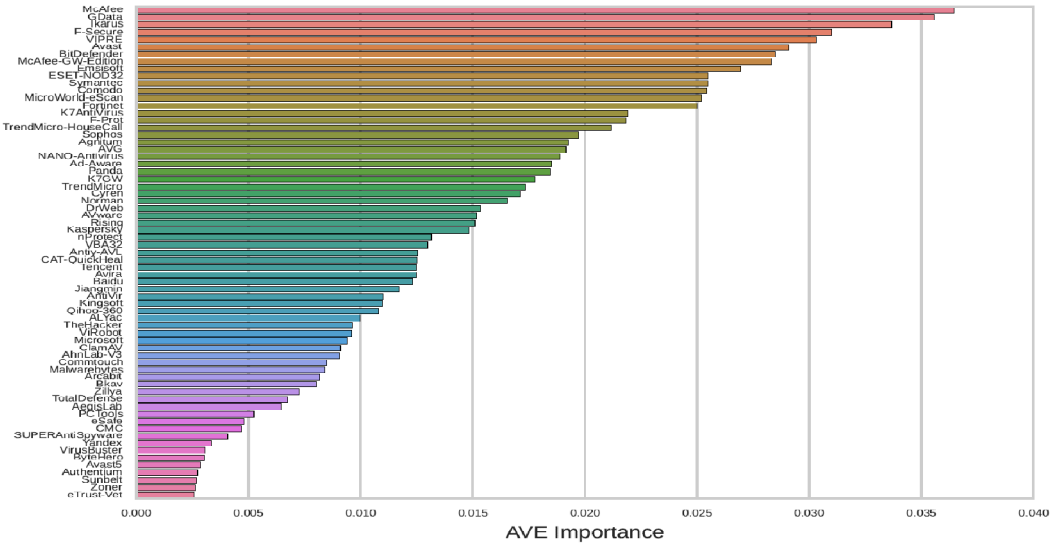

Figure 8. AVT advantage for the *ground-truth set.*

7. CONCLUSIONS

In this work we presented an anti-virus tools analysis using Deep Web malware dataset. The analysis was done on large malware dataset that was crawled by the F-Secure company, using state-of-the-art data analysis techniques, visualizations and graph theory tools, such as community detection algorithm. The analysis was done in order to i) detect common detection capabilities between different anti-virus tools (AVTs), ii) optimize the protection against the largest number of malicious program in a multi-scanning approach and iii) find which AVTs present capability to detect malicious files that other AVTs were not able to detect. The results showed that a lot of the AVTs share similar detection capabilities, due to the fact that they use same detection engine. However, there are some discrepancies between them, such as between gateway and standard AVTs edition, or two AVTs that use same detection engine (due to some in-house solutions). On the other hand, the AVTs that use behavioural approach in detecting malware showed quite unique detection capabilities. The similarity/dissimilarity between AVTs was also shown using community detection algorithm, where three larger AVTs communities were found.

When using a multi-scanning approach, the best solution for the company is to use the most advantageous AVTs, in combination with AVTs from different communities. The MILP approach proposed in the paper, can be used in the future by any company that uses a multi-scanning approach in detecting malware on their mail gateways.

As future work, it remains to analyse the capabilities of different AVT to detect files coming from different sources i.e. downloaded from different domain names. This study could show that some AVTs are more amenable than others to detect several files coming from a given source. The results can denote detection ability for a given malware family (distributed with a domain name specialized for it), which may be due to the crawling of suspicious domain by AV companies to analyse suspicious files in a proactive manner and improve the detection capabilities against new malware distributed by known malicious domains.

ACKNOWLEDGEMENTS

Authors gratefully acknowledge the CyberTrust research project and F-Secure for their support. I.M. work was partially financed by the Faculty of Computer Science and Engineering at the University 'Ss. Cyril and Methodius'.

REFERENCES

[1] M. Lindorfer, M. Neugschwandtner, L. Weichselbaum, Y. Fratantonio, V. v. d. Veen, and C. Platzer, "Andrubis - 1,000,000 apps later: A view on current android malware behaviors," in Proceedings of the Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BAD-GERS), 2014, pp. 3-17.

[2] M. K. Bergman, "White paper: the deep web: surfacing hidden value," Journal of electronic publishing, vol. 7, no. 1, 2001.

[3] A. Mohaisen and O. Alrawi, "Av-meter: An evaluation of antivirus scans and labels," in Proceedings of the 11th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, ser. DIMVA '14. Springer International Publishing, 2014, pp. 112-131.

[4] "VirusTotal: Free service to analyze suspicious files and URLs," https://www.virustotal.com/en/, online; accessed 14 July 2016.

[5] I. Gashi, V. Stankovic, C. Leita, and O. Thonnard, "An experimental study of diversity with off -the-shelf antiVirus engines," in Proceedings of the 8th IEEE International Symposium on Network Computing and Applications, 2009.

[6] I. Gashi, B. Sobesto, V. Stankovic, and M. Cukier, "Does malware detection improve with diverse antivirus products? an empirical study," in Proceedings of the 32nd International Conference on Computer Safety, Reliability, and Security, ser. SAFECOMP '13. Springer Berlin Heidelberg, 2013, pp. 94-105.

[7] J. Canto, M. Dacier, E. Kirda, and C. Leita, "Large scale malware collection: lessons learned," in Proceedings of the 27th International Symposium on Reliable Distributed Systems, ser. SRDS '08, 2008.

[8] Q. Jerome, K. Allix, R. State, and T. Engel, "Using opcode-sequences to detect malicious android applications," in 2014 IEEE International Conference on Communications (ICC), 2014, pp. 914-919.

[9] M. Zheng, P. P. Lee, and J. C. Lui, "Adam: an automatic and extensible platform to stress test android anti-virus systems," in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, 2012, pp. 82-101.

[10] F. Maggi, A. Bellini, G. Salvaneschi, and S. Zanero, "Finding non-trivial malware naming inconsistencies," in Proceedings of the 7th International Conference on Information Systems Security, ser. ICISS '11. Springer Berlin Heidelberg, 2011, pp. 144-159.

[11] A. Kantchelian, M. C. Tschantz, S. Afroz, B. Miller, V. Shankar, R. Bachwani, A. D. Joseph, and J. D. Tygar, "Better malware ground truth: Techniques for weighting anti-virus vendor labels," in Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security, ser. AISec '15. ACM, 2015, pp. 45-56.

[12] A. Kantchelian, M. C. Tschantz, S. Afroz, B. Miller, V. Shankar, R. Bachwani, A. D. Joseph, and J. Tygar, "Better malware ground truth: Techniques for weighting anti-virus vendor labels," in Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security. ACM, 2015, pp. 45-56.

[13] J. Chang, K. K. Venkatasubramanian, A. G. West, and I. Lee, “Analyzing and defending against web-based malware," ACM Computing Surveys (CSUR), vol. 45, no. 4, p. 49, 2013.

[14] H. S. S.L., “Virustotal public api."

[15] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks," Journal of statistical mechanics: theory and experiment, vol. 2008, no. 10, p. P10008, 2008.

[16] H. T. T. Truong, E. Lagerspetz, P. Nurmi, A. J. Oliner, S. Tarkoma, N. Asokan, and S. Bhattacharya, “The company you keep: Mobile malware infection rates and inexpensive risk indicators," in Proceedings of the 23rd International Conference on World Wide Web, ser. WWW '14. ACM, 2014, pp. 39-50.

AUTHORS

Igor Mishkovski was born in Skopje, Macedonia, in 1981. He graduated and received the master degree in computer science and engineering at the University Ss. Cyril and Methodius, Skopje and the Ph.D. degree from Politecnico di Torino, Torino, Italy, in 2008 and 2012, respectively. After he received the Ph.D. degree in 2012 he was elected as an assistant professor at the Faculty of Computer Science and Engineering in Skopje. His research interests include complex networks and modelling dynamical processes, network science, computer networks, semantic web, operating systems.

Sanja Scepanovic received PhD degree at Aalto Univ. in Helsinki on Big Data with focus to Network Analysis applications to socio-technical systems. She holds diploma in Mathematics from Univ. of Montenegro and MSc in Computer Science from Aalto Univ., Finland and Univ. of Tartu, Estonia. Having spent two years in industry and six in research, Sanja aims to work in the applications of research to space industry; in particular by applying her data science skills to analyzing vast amounts of astronomical and other space data. She is an ISU SSP12 alum and serves as a National Point of Contact (NPoC) for Montenegro for the Space Generation Advisory Council (SGAC)

Miroslav Mirchev received his B.S. in computer engineering and M.S. in computer networks in 2008 and 2009 respectively from the Ss. Cyril and Methodius University in Skopje (UKIM), Macedonia. During 2010 he had a research stay at the City University, Hong Kong. In 2014 he defended his Ph.D. thesis at the Polytechnic University of Turin, Italy, and as part of his studies he spent a period at the BioCircuits Institute, UCSD, USA. His areas of interest include network science, computer networks, nonlinear systems, optimization and machine learning. Currently he is an Assistant Professor at the Faculty of Computer Science and Engineering at UKIM.

Sasho Gramatikov received a Bachelor degree in Computer science, information technologies and automation in 2005 and a Master degree in Computer Science and computer engineering degree in 2009, both from the University of Ss. Cyril and Methodius in Skopje, Macedonia. In 2013 he received a PhD degree at the Universidad Politecnica de Madrid (UPM), Madrid, Spain. He is currently working as an Assistant Professor at the Faculty of Computer Science and Engineering at the University of Ss. Cyril and Methodius in Skopje, Macedonia. His research interests are distribution and streaming of video contents in networks.