Holistic Approach for Malware Web Campaigns Identification

Michal Kruczkowski[1,2], Ewa Niewiadomska-Szynkiewicz[2,3]

[1] Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland
[2] Research and Academic Computer Network (NASK),
ul. Wąwozowa 18, 02-796 Warsaw, Poland
[3] Institute of Control and Computation Engineering, Warsaw University of Technology,
ul. Nowowiejska 15/19, Warsaw, Poland

Abstract. Malicious software (malware) is a software designed to disrupt or damage computer system, gain access to users' account or gather sensitive information. Malicious programs can be classified into worms, viruses, trojans, spywares, etc. In recent years numerous attacks have threatened the ability and operation of the Internet. Therefore, mechanisms for successful detection of malicious software are crucial components of network security systems. In this paper the use of selected data mining techniques to malware analysis is investigated. A large number of learning based methods have been developed over the past decades and applied to complex data analysis.

Anti-malware protection and malicious campaigns identification have to be supported by a comprehensive and extensive analysis of data on the Web, and it is a hot topic nowadays. Data mining and learning methods are commonly used techniques to detect malicious software. However, in most investigations the analyses are performed on the data from homogeneous datasources. Novelty of the proposed approach assumes the utilization of data taken from heterogeneous datasources, i.e., data taken from various databases collecting samples from multiple layers of the network ISO/OSI reference model. Assuming multi-source and multi-layered nature of propagation of malware, the authors claim that such approach should help to obtain better results with reference to a widely used analysis performed on data from a given database and a single network layer. In this paper the methodology for automatic identification of malicious campaigns based on the cross-layer analysis of different datasets containing data related to various types of malicious activities is presented. The novelty of the approach is the cross-layer analysis, i.e., comparison the attributes related to an attacker (source point) and a victim (end point), and referred to different layers of the OSI communication model. Final result is the project of a holistic approach for malicious campaigns identification. It combines several machine learning and data mining methods to classify malicious incidents into malware campaigns.

1 Introduction

There are many different classes of malicious programs – worms, viruses, trojans, spyware, etc. Massive, coordinated multi-phased attacks using many computers (addresses, users, etc.) are called malicious campaigns. Mechanisms for effective detection of malicious campaigns are crucial components of network security systems.

Continuous monitoring and measurement of computer network traffic characteristics is necessary. From an operational viewpoint, the Internet service providers and organisations who fight with malware have to face the problems of identifying malicious activity, determining its origin and how the network infrastructure is being abused in order for the attackers to remain anonymous. Because collected traffic logs are usually huge, we employ data mining techniques to unveil significant information from the collected traffic.

Campaign identification determines groups of incidents that have the same goal and employ the same dissemination strategy [1]. The research problem of this paper is to identify malicious campaigns from data associated with malware incidents. Based on the definition of malware campaign we proposed method for malicious software identification.

In this paper a holistic approach for analysis of malicious campaigns is presented. Proposed analysis utilizes the particular combination of various data mining and machine learning methods. Selection of techniques and their parameters has a key influence on accuracy of malware campaigns identification. The case study takes into account a cascade of Frequent Pattern Tree, Support Vector Machine and Classification Graph used for malware campaign identification. Comprehensive and extensive study confirms efficiency of considered approach.

1.1 Motivation

Our main motivation to write this paper is to propose a holistic approach for malicious campaign identification. It combines the use of various supervised learning based techniques to classify threat incident into malicious campaigns. Besides, it employs data mining techniques to deal with these challenges and to propose the malicious campaigns analysis method that deals with the evolutionary aspect of propagating malicious software. Proof of the effectiveness of the presented approach is the experiment based on datasets from real database collecting samples taken from heterogeneous datasources.

1.2 Goal

The ultimate goal of this research is investigation of usefulness of holistic approach for malware campaigns analysis. The authors claim that appropriate selection and skillful processing of threat data can provide powerful information for network security analytics.

The paper is structured as follows. Section 2 presents frequently used methods for threat analysis. In section 3 holistic approach for malware data analysis

is presented. Section 4 presents the case study that utilizes selected techniques for malicious campaigns identification. Outlook for further development of investigated approach is briefly described in section 6. Results are presented in section 5 Finally, conclusions are drawn in section 7.

2 Related Works

In recent years an important direction of research in network security is devoted to design and development of methods and tools for malicious software analysis and detection [2, ?,4–6]. The widely used approach to malware analysis is based on the extraction of information about suspicious communication with the system, the detection of IDS signatures and the generation of new IDS signatures. Honeypot systems are often employed to detect, deflect or counteract attempts at an unauthorized use of information systems. Other techniques utilize algorithms inspired on a human immune system to detect and prevent web intrusions [7, ?]. Malware samples can be used to create a behavioral model to generate signatures, which are served as an input to a malware detector, acting as the antibodies in the antigen detection process. In case of malicious botnets a new trend is to use alternative communication channels, i.e., DNS-tunnelling or HTTP instead of IRC to connect command and control (C&C) servers and infected hosts [9].

A highly scalable and robust approach is to generate a graph based on a malware data analysis and use a graph clustering technique [10] to derive common malware behaviour [11, ?]. The system call traces are used to construct the individual behavioral graph representing the execution behavior of a given malware instance. The method to generate a common behavioral graph representing the execution behavior of a family of malware instances by clustering a set of individual behavioral graphs is proposed in [11]. To speed up the malware data analysis by reducing of sample counts generic hash functions are applied. The generic hash function for portable executable files that generates a per-binary specific hash value based on structural data found in the file headers and structural information about the executable's section data is described in [13].

The key step in malware analysis is the identification of malicious campaigns. This requires a comprehensive and extensive analysis of data taken from the Internet. The common direction is to use statistic analysis [14] and data mining methods [15]. The supervised learning algorithms are successfully used for data classification taking into account the unique set of features. Wide range of applications of these techniques to malware detection is described in literature [16, ?]. The focus is on anomaly detection and similarity analysis of data samples related to the malware programs and malicious campaigns[18, ?].

Study of inter-relationships among botnets through their spam campaigns presented their long term behaviours is described by Thonnard [19]. The visual analytics tool for graph visualization is introduced in work [20]. It demonstrates the use of the tool on a large corpus of spam emails by visualizing spam campaigns performed by spam botnets. However, in most papers devoted to mali-

cious data analysis the focus is on homogeneous data. To identify the malicious campaign (massive attack) one should analyse heterogeneous datasets containing malicious data concerned with various types of network attacks and related events of the infections taken from a numerous sources and organizations.

3 Methodology

Approaches for malware analysis presented in scientific papers are dominated by the use of different methods for threat data processing. Machine learning and data mining techniques are frequently used. Normally, decisions are made based on information about incidents taken from a single source and data are classified based on individual attributes (IP addresses, URLs, domain names, etc.).

In the case of massive attacks detection, it is important to use as many data as well. The utilization of information taken from different network layers (eg. infrastructure, application). In addition, massive attacks can use various propagation techniques, ie. phishing, spam, malicous URL links, etc. Thus, investigation of correlations between the malicious incidents can significantly improve efficiency of campaigns identification.

The holistic approach proposed in this paper is based on an analysis of threat data collected by numerous sensors deployed in various networks and propagated using various techniques. Analysed data are mutually related to various layers of network communication model. Utilizaton of proposed analysis for malware samples enables us to classify threat incidents into malicious campaigns.

Our approach assumes utilizaton of data mining and machine learning methods for classification of malicious incidents. Novelty in relation to the well-known literature is the concept of several techniques combination including association rules and supervised learning methods. It allows to use the complete knowledge about network situation stored in multiple data repositories.

3.1 Holistic approach outline

This section discusses the author's method of malicious campaigns analysis. In the figure 1 the methodology of malware campaigns analysis is shown.

Malware datasets collected in a database are used as a input data into holistic approach. We assume that the data samples are represented as a vector of attributes (generally in different format):

$$data\ sample = [attribute_1, attribute_2, \ldots, attribute_n]. \tag{1}$$

Let us consider the sets of malicious data, respectively related to listed above malicious events, i.e., data samples related to the following attacks: S_b – botnet, S_s – spam, S_c – command and control, S_u – malware URLs and S_p – phishing. The goal is to detect the correlation among these datasets. It comes down to show the relationships among values of relevant attributes related to the data samples from S_b, S_s, S_c, S_u and S_p datasets.

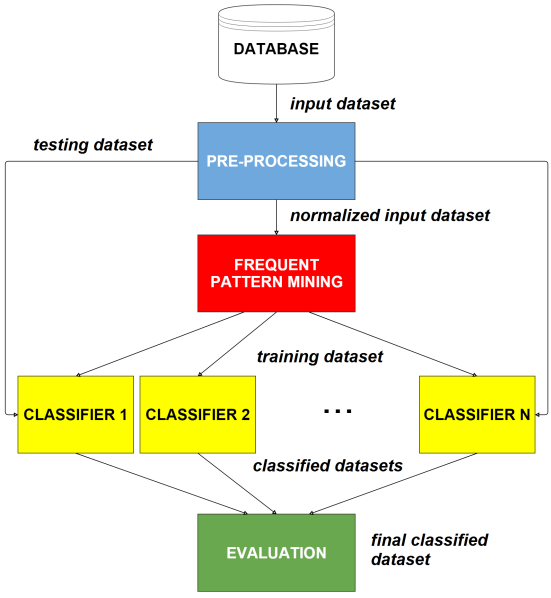

Fig. 1. Holistic method for malware campaigns analysis.

Firstly, we have to select the attributes that can be used to the campaigns identification. In our research we consider cross-layer attributes assigned to each sample collected in S_b, S_s, S_c, S_u and S_p datasets. These attributes are presented and described in tables 1 and 2. Moreover, they are related to various layers of the reference OSI model. Next, the data are processed in four steps:

1. Data are preprocessed in order to construct normalized input dataset - attributes used in classification are represented as a decimal fraction taking the values from $(0, 1)$ interval.
2. Frequent pattern analysis is performed in order to training dataset generation – assigning of a target value to data.
3. Simultaneous data classification by various classifiers. The classifiers are built based on a training dataset (created by frequent pattern mining block) and tested by completely different dataset taken from the database.
4. The results of classification are compared in evaluation block and final classification is performed.

The following subsections discuss operations that are carried out in the blocks presented in the figure 1.

3.2 Data preprocessing

The process of malicious campaigns identification begins with the collection of appropriate datasets containing information about network attacks that are

taken from multiple heterogeneous datasources, ie. different organizations and network security institutions. These data are stored in the knowledge database. Conducting analyzes on the data with different structures, taken from different repositories requires preprocessing. The initial data processing includes standardization and normalization. Final result of data preprocessing is a standardized dataset with value from $(0, 1)$ interval. Normalization refers to the attributes used in the classification process, ie. *IP address*, *source*, *ASN*, *CC*, *category*. This process is described in detail in the paper [21].

3.3 Frequent pattern mining

Frequent pattern mining belongs to widely used techniques for discovering interesting relations between data items in large databases. It is employed today in many application areas including malware detection, Web usage mining, etc. An important and commonly used in detecting network attack, defining attribute of an incident is a URL address.

In this case we extract the frequency correlation between selected attributes. We decide to utilize frequency analysis, because it is obvious that frequency of occurring of appropriate malware incidents determines the identification of malicious campaigns. For example high frequency of occurrence of given IP addresses and domain names in analyzed dataset suggests that these samples can participate in malicious campaigns (the same or different). On the basis of such frequency analyzes we assign class labels (target variable) for data samples. Then, the resultant dataset is used for classification process as a training dataset.

3.4 Classification

In the classification blocks numerous different classification methods are implemented (it depends on experimenter and computation power). All of them are trained based on the same training dataset and tested based on testing dataset taken from the database. The authors claim the use of several method enables us to achieve better results than a single classifier, because the multiple classification methods complement each other and are able to extract knowledge impossible to gain by a single method. In other words utilization of many classification methods is computationally aggravating, but can lead to improved classification results.

3.5 Evaluation

Last block is used to compare classifier results. Each classifier as a result passes tested dataset with assigned classes - classified dataset. Then it can be evaluated based on defined classification criteria. The main task of the block is comparison of these criteria values and extraction of final classification result. The rules of evaluation depend on experimenter.

4 Case Study

In this section we present the experiment setup based on the holistic approach for malware campaign identification. Data processing scheme is presented in the Figure 2.

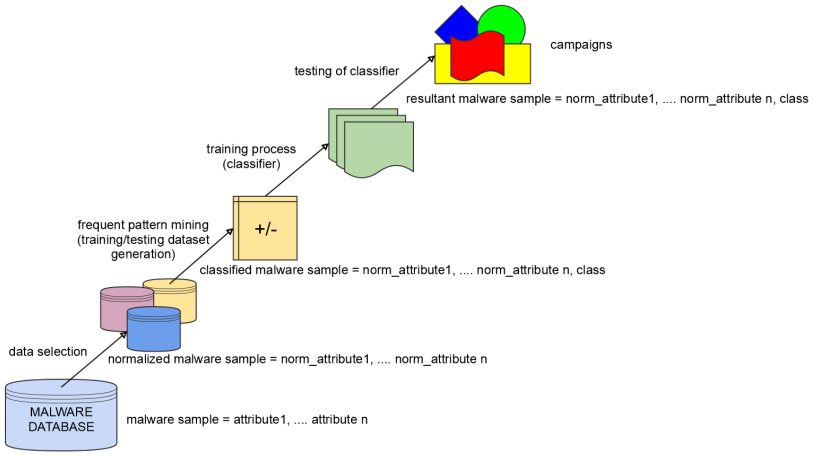

Fig. 2. Data processing.

4.1 Experiment overview

Figure 3 presents the experiment setup. The n6 platform is used as the database. Frequent pattern mining block implements Frequent Pattern-Growth algorithm (FP-Growth). As classifiers three-class Support Vector Machine (3-SVM) and author's classification graph are implemented. The classification results are evaluated as follows:

1. if both classifiers predictions are in line (predicted the same class value $\{+1\}$ or $\{0\}$ or $\{-1\}$) then the class is treated as a final prediction,
2. if one classifier predicted class $\{+1\}$ or $\{-1\}$ and second predicted class $\{0\}$ then class $\{+1\}$ or $\{-1\}$ is treated as a final prediction,
3. if classifiers predicted completely different classes (one predicted $\{+1\}$ and second $\{-1\}$ or vice versa) then the final prediction is assigned to class $\{0\}$.

The following subsections discuss the blocks presented in the figure 3.

4.2 Database

The proposed experiment used data from a real malicious software database provided by the n6 platform [22] developed at NASK (Research and Academic

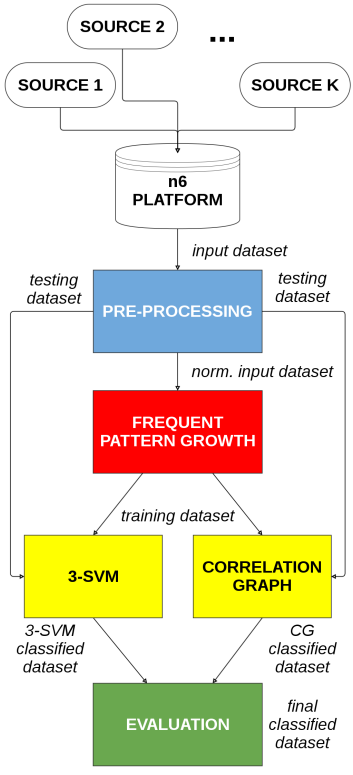

Fig. 3. The setup of the experiment.

Computer Network). The n6 platform (Fig. 4) is used to monitor computer networks, store and analyse data about incidents, threats, etc.

The n6 database collects data taken from various sources, including security organizations, software providers, independent experts, etc., and monitoring systems serviced by CERT Poland. The datasets contain URLs of malicious websites, addresses of infected machines, open DNS resolvers, etc. Most of the data is updated daily. Information about malicious sources is provided by the platform as URL's, domain, IP addresses, names of malware, etc.

In the experiment five datasets characterised by different malware propagation techniques are utilized:

- S_u – malicious URLs,
- S_s – spam,
- S_p – phishing,
- S_b – botnets,
- S_c – C&C servers.

In the tables 1 and 2 attributes associated with analyzed datasets are described. Each of data sample corresponds with malicious incident. It is repre-

Fig. 4. The n6 platform architecture.

sented by a vector of attributes. For the classification we utilize only compulsory set of attributes. It should be noted that the format of these attributes are different, but the use of preprocessing enables us to obtain a normalized of values of attributes used to classification process.

Table 1. Compulsory attributes.

Attribute	Description
Time	Time of occurrence of a incident in Unix Timestamp format.
Category	Category of a incident: 1-malicious URL, 2-spam, 3-phishing, 4-botnet, 5-C&C servers.
Source	Source name that detected an event.
IP address	IP address in byte-decimal format.
Country code	Country code compatible with ISO 3166-1 alpha-2.
ASN	Autonomous System Number.

Table 2. Optional attributes.

Attribute	Description
Domain	Full domain name.
Name	Name of a incident (if is known).
MD5	A hash to a binary file with a threat.
URL address	An Uniform Resource Locator.

4.3 Frequent pattern growth (FP-Growth)

In our research we use the frequent pattern tree structure (FP-tree) – a prefix structure for storing quantitative information about frequent patterns in a

database – and the FP-growth algorithm for frequent pattern discovery using a divide-and-conquer strategy, both developed by J. Han et al. [23]. The FP-growth algorithm operates in two steps: 1) the FP-tree compact structure is constructed, 2) frequent patterns are extracted from the FP-tree. In the first step the occurrence of patterns in the input transaction database is counted. Next, infrequent patterns are discarded, frequent patterns are sorted in descending order of their frequency in the database, and the FP-tree structure is built. Common, usually most frequent patterns are shared. Therefore, FP-tree provides high compression close to tree root and can be processed quickly. Recursive growth is applied to extract the frequent patterns. FP-growth starts from the bottom of the tree structure (longest branches), by finding all patterns matching given condition. New tree is created, etc. Recursive growth ends when no patterns meet the condition, and processing continues on the remaining main branches of the original FP-tree.

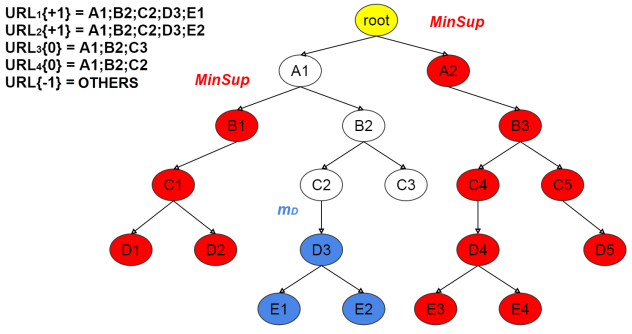

Fig. 5. Constriction of training dataset based on Frequent Pattern Tree exploration.

Values of thresholds $MinSupp$, m_D are determined in order to generate training dataset. Based on these values the input dataset is divided into three subsets. $MinSupp$ is minimum support value specifying the threshold for frequent tokens. In our case token correspond with a fragment of URL address resultant from URL fragmentation (Sec. 5.2). For example if a $MinSupp = 30\%$ then each token with support value less than $MinSupp$ is treated as infrequent and is assigned as $\{-1\}$ – not participating in any campaigns. Others are frequent tokens. While decision value m_D determines minimal number of tokens occurring in URLs needed to classify it to malicious campaign. In other words it m_D described how many tokens should be the same in URLs to treat these URLs as a common malicious campaigns. Based on the m_D threshold classes $\{+1\}$ – participating in malicious campaign and $\{0\}$ – suspect are assigned to data. The method for assigning classes to URLs is shown in the figure 5. It utilizes $MinSupp$ and m_D value. These variables determines the assigning of target values to the training dataset. All of analyzed URLs (data) are associated with

malicious software. In proposed solution FP-Growth algorithm generates the reference data - assign class labels to the data vectors. Obviously, the situation can occur that reference (assigned) datasets are known. Then the frequent pattern analysis can be omitted in this case. Actually we have not got access to any classified samples into malicious campaigns and we treat the resultant file from frequent pattern analysis as a 100% confidence.

4.4 Three Class Support Vector Machine (3-SVM)

The Support Vector Machine (SVM) [24] is a supervised learning classification method widely used in data mining research. The classical concept of SVM is to classify each data sample into one of two categories: positive class denoted by $\{+1\}$ and negative class denoted by $\{-1\}$. Thus, the goal is to determine a decision boundary, which divides data into two sets (one for each class), a plane for $n \leq 3$ or a hyperplane for $n > 3$, where n denotes the size of a dataset. Next, all the measurements on one side of this boundary are classified as belonging to $\{+1\}$ class and all those on the other side as belonging to $\{-1\}$ class. The problem is that many such hyperplanes can be determined, and the best one has to be selected. Hence, SVM tries to learn the decision boundary which gives the best generalization. A good separation is achieved by the hyperplane that has the largest distance to the nearest data sample of any class – a wider margin implies the lower generalization error of the classifier. To select the maximum margin hyperplane an optimization problem is formulated and solved for a given training dataset.

3-SVM is a modification of the classical SVM that allows to classify samples into three classes. The principle of the algorithm is based on solving three problems with two-classes each. Denoting the discrimination rule for three classes by the expression:

$$f_k(\mathbf{x}) = \operatorname{sgn}((\sum_{\text{supp. vectors}} y_i \alpha_i^{0,k}(\mathbf{x}_i, \mathbf{x}_j) + b^{0,k})) \equiv \operatorname{sgn}(\tilde{f}_k(\mathbf{x})), \tag{2}$$

for a given observation $\mathbf{x}$, final decision rule $k(\mathbf{x})$ becomes:

$$k(\mathbf{x}) = \arg\max\{\tilde{f}_1(\mathbf{x}), \tilde{f}_2(\mathbf{x}), \tilde{f}_3(\mathbf{x})\}, \tag{3}$$

where $f_k(\mathbf{x})$ is discrimination rule, $\alpha_i^{0,k}$, $b^{0,k}$ are Lagrange function coefficients for k-th class, $\mathbf{x}_i$ is datasample, $\tilde{f}_k(\mathbf{x})$ is discrimination hyperplane.

For the 3-SVM classification we utilize normalized data from n6 platform. The training dataset consist of attributes: *IP address*, *source*, *ASN*, *CC*, *category*, *class*. Class labels of data are extracted from frequent pattern analysis. Finally, the samples are classified to classes chosen by the majority of considered classifiers (in this case at least two). In this way samples are assigned to the following classes:

- class $\{+1\}$ - the samples participating in the malicious campaigns,
- class $\{0\}$ - the suspected samples,

– class $\{-1\}$ - the samples not participating in the malicious campaigns.

The trained 3-SVM classifier can be used for classification of samples from a new dataset based only on the knowledge about their attributes.

4.5 Classification graph

The author's method – classification graph is generated based on the cross-layer correlations between malicious URL incidents and of various datasets related to various incidents: spam, phishing, botnets, C&C servers. It corresponds with malicious campaigns and shows the relationships among these datasets based on values of selected attributes. Correlation graph provides additional knowledge about campaigns character and frequency. Finally the results are compared with the results from 3-SVM classification. The classification graph is formed by vertices and edges. The vertices represent all detected values of a selected attribute. First attribute can be assigned to vertices: IP address (*ip*) and second to edges: domain names (*domain*). In this way we can create bipartiate graph, where edges represent IP addresses connected by common domain name. The classification graph algorithm is based on a comprehensive multiple malware datasets analysis. It enables us to classify samples into three classes corresponding with the campaigns:

– $\{+1\}$ for heterogeneous and frequent samples – participating in the malicious campaigns,
– $\{0\}$ for heterogeneous and infrequent or homogeneous and frequent samples – suspicious,
– $\{-1\}$ for homogeneous and infrequent samples not participating in any malicious campaigns.

A sample is heterogeneous if its IP address occurs in various types of datasets. Otherwise it is homogeneous. While a sample is considered as a frequent (or not) according to the following assumptions:

$$\begin{cases} IP^k \text{ is frequent} & \Leftrightarrow \quad R_f(IP^k) \geq \frac{0.5}{N} \sum_{i=1}^{N} R_f(IP^i), \\ IP^k \text{ is infrequent} & \Leftrightarrow \quad R_f(IP^k) < \frac{0.5}{N} \sum_{i=1}^{N} R_f(IP^i), \end{cases} \tag{4}$$

where $R_f(IP^i)$ is the function that calculates the frequency of the i-th IP address and N is the number of samples. Finally, the classification graph algorithm works as the classifier that divide dataset into three disjoint classes. The classification graph algorithm is described in detail in the paper [21].

5 Results

5.1 Assumptions for the experiment

The experiment is carried out based on the following assumptions:

1. In the experiment holistic approach for malicious campaign analysis described in section 3 is used.
2. In the experiment data are taken from the n6 platform described in subsection 4.2.
3. Malicious URLs dataset is the investigation object. It consist of 10 attributes: *time*, *source*, *category*, *IP address*, *ASN*, *domain name*, *name*, *md5*, *URL address*.
4. In the classification graph datasets are associated with: spam, phishing, botnets, C&C servers are utilized in order to detect the correlation between these datasets and malicious URLs by domain names and IP addresses.
5. The experiment is used of identification of malicious campaigns based on historical database that collects 100 000 samples (Tab.3).

Table 3. Characteristic of input malicious URLs dataset.

Number of URLs	100 000
Number of unique URLs	18 261
Number of unique IP addresses	11 002

It can be concluded that only about 18% of URLs and about 11% of IP addresses represent unique values. This points to many correlations between the analyzed threats and suspicion of massive attacks – malicious campaigns.

6. Construction of training datasets is performed based on exacted frequent tokens by the FP-Growth algorithm. The minimum support value $MinSupp$ is fixed at 30% the number of occurrences of the token, which is characterised by the highest incidence (one of the root neighbour in $FP-Tree$). All tokens with the support value less than 30% are classified into a group of tokens unrelated to any campaigns ($\{-1\}$). Other tokens were separated into two groups based on the decision variable m_D. The value of m_D indicates to the minimum number of frequent tokens required to classify a given URL as a participant of malicious campaigns. In the experiment we assume $m_D = 3$, so all URLs with the number of frequent tokens greater or equal than three are classified as $\{+1\}$, and less or equal to three are assigned as $\{0\}$ (Fig. 5).
7. As classifiers results validation cross-validation technique with five folds is used and evaluation criteria are defined below:
   Sensitivity – *Sens*

$$Sens = \frac{TP}{TP + FN}, \tag{5}$$

   Specificity – *Spec*

$$Spec = \frac{TN}{TN + FP} = 1 - \frac{FP}{TN + FP}. \tag{6}$$

   Accuracy – *CA*

$$CA = \frac{TP + TN}{TN + FP + FN + TP}. \tag{7}$$

Area Under ROC Curve – *AUC*

$$AUC = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} 1_{y_i^+ > y_j^-}}{mn}, \tag{8}$$

Finally, based on the experiment we shows:

- results of operation and correctness of the design of the individual methods in holistic approach for malicious campaign analysis,
- the interconnectedness of the experiment setup's blocks,
- the impact of the training dataset size and test the results of data classification.

5.2 Generation of training dataset

At first we analyze FP-Growth algorithm for training dataset generation ie. the assignment of classes to the input data. The preprocessed URLs are extracted from malicious URLs dataset. For the analysis each URL is converted and divided into substrings (tokens) in accordance with the rules set out below:

- schemes removal:
  $''http''$, $''https''$.
- domain names mining (schemes and extensions removal):
  $''www''$, $''org''$, $''com''$, etc.
- URLs fragmentation based on following separators:
  $''/''$, $''.''$, $''?''$, $''\#''$.

The result of the transformation is collection of URLs containing transformed addresses represented by sets of tokens.

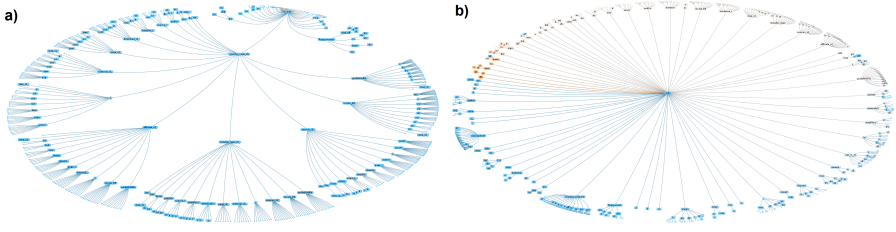

Fig. 6. The FP-Tree of analyzed dataset a) zoom in campaign branch, b) general view.

Then, using the FP-Growth algorithm, the FP-Tree structure is built, whose vertices are extracted tokens (Fig.5). The assumed values of $MinSupp$ and m_D determine the distribution of classes into the dataset. This dataset is used for classifiers in the learning process as a training dataset.

The distribution of classes in the training dataset is presented in the table 4.

Table 4. Characteristic of training dataset.

Data participating in campaigns ($\{+1\}$)	33%
Suspicious data ($\{0\}$)	41%
Data unrelated with any campaigns ($\{-1\}$)	26%
Total number of data	100 000

5.3 Construction of 3-SVM classifier

Now, we consider the classifier model that implements 3-Support Vector Machine classification method described in subsection 4.4. Data are classified based on 5 of the 10 attributes described in subsection 4.2 (*IP address*, *source*, *ASN*, *CC*, *category*).

The classifier was trained based on training dataset that target values were extracted from URLs frequent pattern analysis. Moreover the classifier was tested by completely different dataset from n6 platform, ie. training dataset contains one-day data from n6 platform and testing dataset contains data corresponding with next day. Each of dataset contains target value based on frequent pattern analysis. In the table 5 the confusion matrices for learning process are presented. It follows that 82.3% of samples from class $\{+1\}$, 83.1% samples from class $\{0\}$ and 72.2% samples from class $\{-1\}$ are correctly classified. It ensures the accuracy value equals 80% and area under ROC curve (AUC) equals 97%. Other values of classification criteria are collected in the table 6.

Table 5. Confusion matrix for cross-validation with 5 folds.

Classes	$\{+1\}$	$\{0\}$	$\{-1\}$
$\{+1\}$	82.3%	9.8%	8.0%
$\{0\}$	11.8%	83.1%	5.1%
$\{-1\}$	19.4%	8.4%	72.2%

Table 6. Classification rating based on training dataset.

Criterion	$\{+1\}$	$\{0\}$	$\{-1\}$
$Sens$	0.82	0.83	0.72
$Spec$	0.85	0.91	0.94
CA	0.80	0.80	0.80
AUC	0.97	0.97	0.97

Subsequently, constructed classifier has been tested based on totally different data taken from n6 platform. The results are presented in the table 6 and the table 7. It should be noted that in this case the accuracy equals 76% and area

Table 7. Confusion matrix for test dataset.

Classes	{+1}	{0}	{−1}
{+1}	67.8%	28.8%	3.4%
{0}	3.8%	93.4%	2.7%
{−1}	7.0%	34.2%	58.8%

under ROC curve (AUC) equals 89%.

Table 8. Classification rating based on testing dataset.

Criterion	{+1}	{0}	{−1}
$Sens$	0.68	0.93	0.59
$Spec$	0.95	0.69	0.97
CA	0.76	0.76	0.76
AUC	0.89	0.89	0.89

In the next step we evaluate the influence of training dataset size on the quality of classification. The results are presented in the figure 7. It is easy to see that 100 000 samples in dataset size is sufficient to learning of these classifiers.

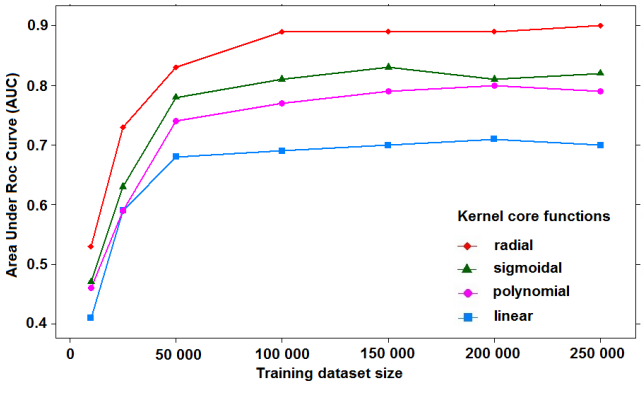

Fig. 7. Results of the selection of the optimal training dataset size and kernel core function.

5.4 Classification graph

Another technique of classification is the author's method described in subsection 4.5. It utilizes the training dataset constructed by the FP-Growth algorithm. The table 9 shows the distribution of samples into three classes assigned as $\{+1\}$, $\{0\}$ and $\{-1\}$. Furthermore it contains numbers of unique values of IP addresses and domain names.

Table 9. Results of classification using correlation graph.

Class $\{+1\}$	41%
Class $\{0\}$	37%
Class $\{-1\}$	22%
Number of unique IP addresses	5 356
Number of unique domain names	7 045

The method is presented in detail in the paper [21].

5.5 Evaluation

In the last subsection we present final result of classification process. Table 10. Now we should note that the resultant distribution of classes is quite similar

Table 10. Final results of classification.

Data participating in campaigns ($\{+1\}$)	37%
Suspicious data ($\{0\}$)	39%
Data unrelated with any campaigns ($\{-1\}$)	24%

to the training dataset. It proves the correctness of the investigated method. Furthermore the results of classification based on testing dataset using holistic approach for malware campaigns identification are compared with single SVM classifier in the table 11. It is easy to notice the holistic approach significantly

Table 11. Comparative analysis of classification for malware samples.

Criteria	SVM	Holistic approach
CA	0.76	0.83
AUC	0.89	0.92

improve identification of malicious campaigns. The classification accuracy equals

83%, but authors claim that these preliminary results can be improved in the future.

6 Outlook

For further analysis we are planning to perform long-term experiments (several months). Subsequently, the values of many properties (training dataset size, selection of classifiers parameters, values of decision variables, etc) can be optimized. On the other hand the long term analysis can provide interesting knowledge about propagation of malicious campaigns in time as a dynamic, changeability, etc.

7 Summary

In this article, we propose the method for identification of malicious campaigns based on cross-layer comprehensive analysis of malware datasets. The main idea is to compare values of selected attributes related to the collected malicious software and discover frequently occurring values of these attributes. The results of the analysis are presented in based on defined evaluation criteria. The existence of the same values of attributes in most datasets suggests the massive attack – a campaign.

The case study for datasets taken from the n6 platform confirms that the effective analysis of a malware campaign needs comprehensive, cross-layer analysis of data from the Web. We claim that the presented technique based on analysis of datasets containing data related to various types of malicious software can be implemented in the intrusion detection systems. It can be successfully used to perform an analysis of dynamic, heterogeneous, unstructured and imbalanced network data to preliminary detection of campaigns.

Acknowledgements

This research was partially supported by the "Nippon - European Cyberdefense-Oriented Multilayer threat Analysis - (NECOMA)" FP7 grant agreement number 608533 and research fellowship within Project "Information technologies: Research and their interdisciplinary applications", agreement number UDA POKL.04.01.01-00-051/10-00.

References

1. Calais, P., Pires, D., Neto, D., Meira, W., Hoepers, C., Steding-Jessen, K.: A campaign-based characterization of spamming strategies. In: CEAS'08. (2008) 1–6
2. Rieck, K., Holz, T., Willems, C., P.Dussel, Laskov, P.: Learning and classification of malware behavior. 5th International Conference, DIMVA 2008 (2008) 108–125

3. Wagner, C., Wagener, G., R. State, T.E.: Malware analysis with graph kernels and support vector machines. Malicious and Unwanted Software (MALWARE) (2009) 63–68
4. Glowacka, J., Parobczak, K., Amanowicz, M.: On mechanism supporting situational awareness of a tactical ad-hoc network nodes. In: Military Communications and Information Technology: Recent Advances in Selected Areas. (2013) 165–180
5. Kozakiewicz, A., Felkner, A., Kijewski, P., Kruk, T.: Application of bioinformatics methods to recognition of network threats. Journal of Telecommunications and Information Technology (2007) 23–27
6. Kruczkowski, M., Niewiadomska-Szynkiewicz, E.: Support vector machine for malware analysis and classification. In: Proc. of IEEE/WIC/ACM Inter. Conf. on Web Intelligence. (2014) 1–6
7. de Oliveira, I.L., Ricardo, A., Gregio, A., Cansian, A.: A malware detection system inspired on the human immune system. Computational Science and Its Applications ICCSA 2012 (2012) 286–301
8. et al., S.F.: Self-nonself discrimination in a computer. Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy (1994) 311–324
9. E. Stalmans, B.I.: A framework for dns based detection and mitigation of malware infections on a network. Information Security South Africa (ISSA) (2011) 1–8
10. Shaeffer, S.: Graph clustering. Computer Science Review (2010) 27–64
11. Park, Y., Reeves, D., Mulukutla, V., Sundaravel, B.: Fast malware classification by automated behavioral graph matching. CSIIRW '10 Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research (2010) 1–4
12. Elhadi, A., Maarof, M., Barry, B.: Improving the detection of malware behaviour using simplified data dependent api call graph. International Journal of Security and Its Applications (2013) 29–42
13. Wicherski, G.: pehash: A novel approach to fast malware clustering. Proceeding LEET'09 Proceedings of the 2nd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more (2009) 1–8
14. M. Zubair Shafiq, S. Ali Khayam, M.F.: Embedded malware detection using markov n-grams. Detection of Intrusions and Malware, and Vulnerability Assessment (2008) 88–107
15. Franklin, M., Halevy, A., Maier, D.: From databases to dataspaces: A new abstraction for information management. Sigmod Record (2005) 27–33
16. Lasota, K., Kozakiewicz, A.: Analysis of the similarities in malicious dns domain names. In: Data Management and Applications, Communications in Computer and Information Science. Volume 187., Springer-Verlag (2011) 1–6
17. Yanfang, Y., Wang, D., Li, T., Ye, D.: Imds: Intelligent malware detection system. Proceedings of KDD'07 (2007) 1043–1047
18. M.R. Faghani, H.S.: Malware propagation in online social networks. Malicious and Unwanted Software (MALWARE) (2009) 8–14
19. Thonnard, O., Dacier, M.: A strategic analysis of spam botnets operations. CEAS '11 Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (2011) 162–171
20. Tsigkas, O., Thonnard, O., Tzovaras, D.: Visual spam campaigns analysis using abstract graphs representation. VizSec'12 Proceedings of the Ninth International Symposium on Visualization for Cyber Security (2012) 64–71
21. Kruczkowski, M., Niewiadomska-Szynkiewicz, E., Kozakiewicz, A.: Cross-layer analysis of malware datasets for malicious campaign identification. Proceedings of

the International Conference on Military Communications and Information Systems (2015)
22. NASK: n6 platform. http://www.cert.pl/news/tag/n6 (2014)
23. Han, Y., Pei, Y., Yin, Y.: Mining frequent patterns without candidate generation. In: Proc. of SIGMOD. (2000) 1–12
24. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York, NY, USA (1995)