DOI 10.12694/scpe.v18i3.1300
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 18, Number 3, pp. 195–205. http://www.scpe.org © 2017 SCPE

A FRAMEWORK FOR GENERATING MALWARE THREAT INTELLIGENCE

EKTA GANDOTRA[*] DIVYA BANSAL[†] AND SANJEEV SOFAT[‡]

Abstract. Ubiquitous computing devices with network capabilities have become the critical cyber infrastructure for academia, industry and government in day-to-day life. The cyber-attacks being launched on this critical infrastructure have shifted to the pursuit of financial profit and political gains which lead to cyber warfare on various scales. The evolution of new practices like social networking, explosion of mobile devices and cloud computing have given opportunities to attackers for discovering vulnerabilities and exploiting these for creating sophisticated attacks. Malware is one of the most dreadful security threats fronting the Internet today. It is evolving and making use of new ways to target computers and mobile devices. Moreover, the exponential escalation in their volume and complexity has increased the damage caused by them. These have the capability to circumvent the earlier developed methods of detection and mitigation which clearly shows the need of shifting from traditional cyber security to cyber security intelligence. This paper purposes a design of a framework for generating Malware Threat Intelligence that can analyze, identify and predict the malware threats and can act as an Early Warning System (EWS). It also presents the real-time testing of the proposed framework which is realized by designing a prototype for providing security-as-a-service.

Key words: malware, threat intelligence, malware analysis, malware detection, malware threat level

AMS subject classifications. 68M14, 68T27

1. Introduction.

Cyberspace comprises of people, services and software linked either directly or indirectly to the Computer Networks, Internet and Telecommunications. The services and products residing on cyberspace have been adopted in almost all the sectors. Moreover, people have become habitual of services being provided by the Internet. The preservation of Confidentiality, Integrity and Availability of information and protection of critical infrastructure is the essence of secure cyberspace. During the past several years, the frequency and complexity of cyber-attacks have been changed. The evolution of new practices like social networking, explosion of mobile devices and cloud computing have given opportunities to attackers for discovering vulnerabilities and exploiting these for creating sophisticated attacks. Moreover, the new generation cyber-attacks have become more targeted, persistent and unknown. Most of these attacks are launched by people who use the Internet with wicked intentions. They make use of malicious programs (also known as malware) for this purpose.

A malware is a software program that achieves the damaging intent of an attacker [1]. According to NIST (National Institute of Standards and Technology), it is a program that has the intent of compromising the Confidentiality, Integrity, or Availability of the victim machine and its resources [2]. According to Internet Security Threat Report, Symantec [3], over 430 million new malicious specimens were discovered in the year 2015 which is about 36% more than that in 2014. Malware writers are making use of obfuscation techniques like insertion of dead code, subroutine reordering, instruction substitution etc. for creating polymorphic and metamorphic malware [4]. Moreover, the malware are becoming sophisticated, targeted, persistent, stealthy and unknown day by day. These have the capability to circumvent the earlier developed methods of detection and mitigation which clearly shows the need of shifting from traditional cyber security to Cyber Threat Intelligence (CTI). Popular security organizations providing CTI services include FireEye, LogRhythm, RSA, Symantec, and Verisign etc.

1.1. Cyber Threat Intelligence.

According to Gartner, "Threat intelligence is evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging threat to assets that can be used to take decisions" [5]. It answers the questions like what methods are being used by attackers? What is their motive? What platform they are targeting at? etc. Secureworks [6] identifies CTI as a service, which is intended to help clients by providing them with early warnings on emerging threats, vulnerabilities and consultation with the threat intelligence group to have discussion on the same. Threat

[*]Department of Computer Science and Engineering, PEC University of Technology, Chandigarh, India and Department of Computer Science and Engineering, Chitkara University Institute of Engineering and Technology, Chitkara University, India (ekta.gandotra@gmail.com)

[†]Department of Computer Science and Engineering, PEC University of Technology, Chandigarh, India (divya@pec.ac.in)

[‡]Department of Computer Science and Engineering, PEC University of Technology, Chandigarh, India (sanjeevsofat@pec.ac.in)

intelligence doesn't only provide general threat information, but also aid in prediction of a particular threat to a specific organization in future. The security community requires the methods that have the capability to provide the situational awareness and would have the ability to proactively detect and respond to new malware threats. Therefore, it is imperative to obtain CTI about the evolving malware internally as well as from the external sources. The intelligent information so obtained can help the organizations and other stakeholders to focus and prioritize the defensive methods used to deal with future threats [7].

This paper proposes the design of a framework for generating CTI (specifically targeting the malware threat), which on the other hand can act as EWS. It provides the detailed description of the framework explaining all its components and various modules along with their functionality. It describes various tools and methodologies used for the framework design. The framework, when deployed in a cloud environment can analyze and distinguish malicious traffic from the benign one and can provide security-as-a-service. It also presents the real-time testing of the proposed framework which is realized by designing a prototype for providing security-as-a-service. This type of intelligence can help the security analysts to take the preventive measures to stop the future attacks or to minimize the risks posed by them.

2. Framework Design and Architecture. This section provides an overview of the design and architecture of the framework for generating malware threat intelligence. Investigation of the incidents or changes caused by a malicious program on the victim machine can definitely help in its detection and take preventive measures to avoid similar attacks in future. Table 2.1 briefly describes the steps involved in the process of malware detection and prediction which is the significant part of malware threat intelligence framework.

Table 2.1
Steps for generating malware threat intelligence

Step 1: Incident Response
Step 1.1 Data Capturing
Step 1.2 Malware Detection
Step 1.3 Incident Reporting
Step 2: Conduct In-depth and Conclusive Analysis
Step 2.1 Data Acquisition and Preparation
Step 2.2 Malware Analysis
Step 2.3 Malware Classification and Threat Assessment
Step 2.4 Visualization and Reporting

Each step is further comprised of several sub-steps. The detailed process is described by using a flow diagram shown in Fig. 2.1. The process begins by the definition of the ***Scope and Purpose*** of the task. The purpose of performing malware analysis could be to identify or detect zero-day malware and to get prepared to prevent their future occurrences and/or to identify the source of malware attack and to study the behavioral trends to take the preventive measures to stop such attacks in future or to minimize the risks posed by them.

In the ***Groundwork*** step, the sources from where data is to be captured/collected are identified. Proper permission and rights are obtained from the concerned authorities and administrators.

Data Acquisition is the step where network data is monitored to identify and capture malicious data or it is acquired from various Virus collection databases which are available for download in the public domain after free registration. The clean files can be acquired manually from System32 directories of Windows 2000, Windows 2003, Windows XP and Windows 7. After collecting random malicious samples from diverse sources, these are scanned using an Anti-Virus (AV) tool in the ***Data Preparation*** step to confirm them as malicious and then filtered to keep only the unique malware (based on their MD5 hashes). ***Automated Malware Analysis*** is performed by executing the acquired malicious and clean files in the sandboxed environment to produce the analysis reports which are preserved in the ***Data Preservation*** step. These reports are then studied and analyzed to identify useful features. The irrelevant features are removed in ***Feature Extraction*** step. ***Data Mining*** involving machine learning and statistical modeling is used for knowledge discovery in the data. Supervised machine learning algorithms are used in our present work to build the classification models

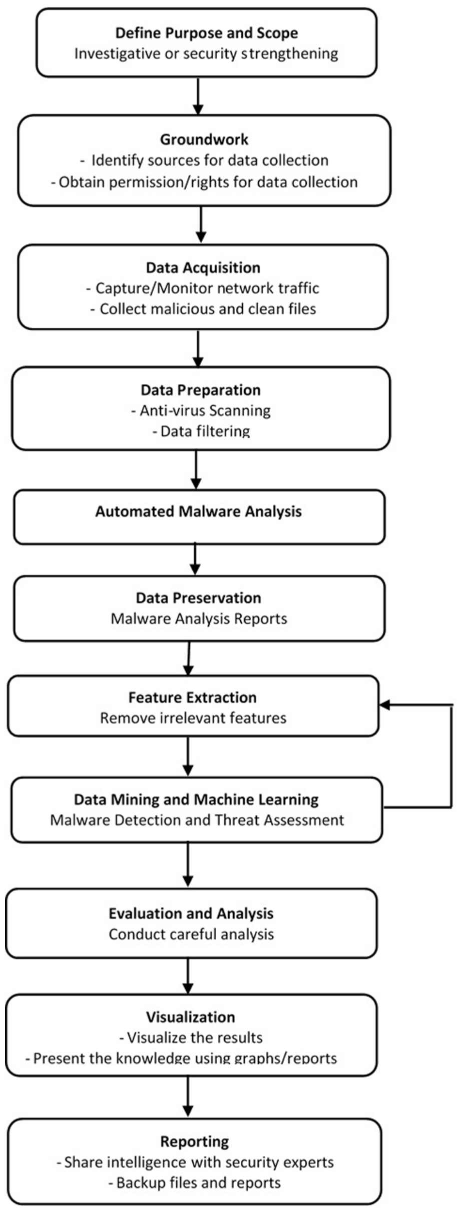
Define Purpose and Scope
Investigative or security strengthening
Groundwork
- Identify sources for data collection
- Obtain permission/rights for data collection
Data Acquisition
- Capture/Monitor network traffic
- Collect malicious and clean files
Data Preparation
- Anti-virus Scanning
- Data filtering
Automated Malware Analysis
Data Preservation
Malware Analysis Reports
Feature Extraction
Remove irrelevant features
Data Mining and Machine Learning
Malware Detection and Threat Assessment
Evaluation and Analysis
Conduct careful analysis
Visualization
- Visualize the results
- Present the knowledge using graphs/reports
Reporting
- Share intelligence with security experts
- Backup files and reports

Fig. 2.1. Malware detection and prediction process.

for zero-day malware detection which are then compared to finalize the most suitable one. An unsupervised Fuzzy modeling is used to assess the threat level of a malware on the basis of which it could be prioritized for performing manual analysis. ***Evaluation and Analysis*** is the key step of the process. The machine learning algorithms are evaluated for their performance. The evaluation and analysis results are presented in ***Visualization*** step. Data visualization is the graphical representation of data which is helpful for obtaining an overall view and locating important aspects within the dataset. ***Reporting*** is the concluding step of the process in which the generated intelligent information is shared with security experts and other stakeholders. On the basis of this information, they issue early warnings and corresponding remedial actions to be taken to deal with emerging malware threats.

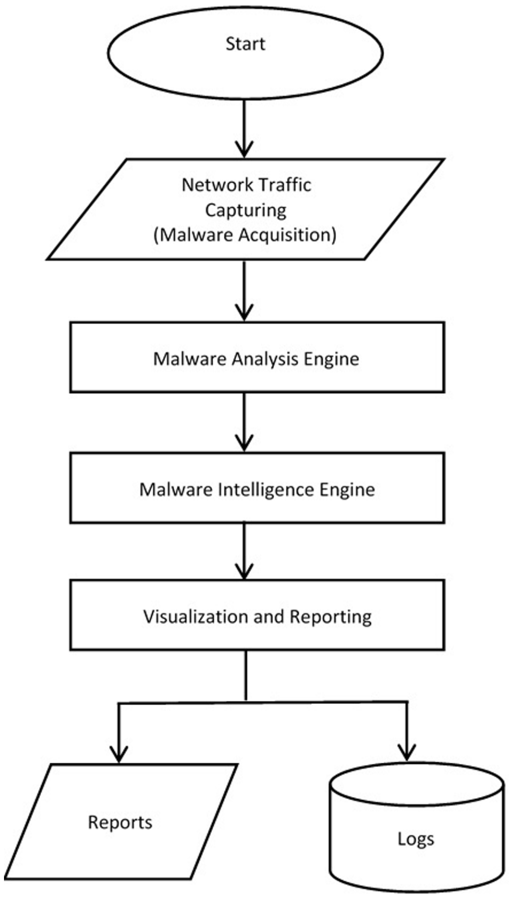
Start
Network Traffic Capturing (Malware Acquisition)
Malware Analysis Engine
Malware Intelligence Engine
Visualization and Reporting
Reports
Logs

FIG. 2.2. Design overview.

2.1. Design. The proposed framework has been designed while keeping in mind the objectives like extensibility, scalability, quick incident reporting and efficiency. Fig. 2.2 shows an overview of the design of the proposed framework. Broadly the design can be classified into three phases. After acquiring the malware samples, these are scanned with AV engines in the first phase so as to filter out according to the family to which they belong or on the basis of the operating system they target. Thereafter, these are made to execute in the sandboxed environment to generate malware analysis reports. These reports are then analyzed to gain insight into the malware behavior. In the 2nd phase, pattern matching is done on the malware behavior data and classification model is built to detect zero-day malware and also their threat level is assessed so that these specimens could be prioritized for allocating resources in order to perform closer manual analysis. This phase raises alerts and creates logs containing information about the detected malware and their threat level. For the newly detected malware samples, signature generation is done and is stored in the database repository. For the next phase, malware behavioral trend analysis is performed and visualized to have an insight into the malware behavior. The information so obtained can be shared with security agencies and other stakeholders so that they can issue advisories and preventive measures to be taken to deal with future malware threats.

2.2. Architecture. This section gives the detailed description of architecture of the complete system. The proposed framework has the capability to provide a greater awareness of emerging threats and can deal with malicious programs in a proactive way. It combines the malware analysis data consisting of behavioral attributes, their classification into the existing families or identifying those which don't fall under any class, signature generation, malware detection, their behavioral trend analysis etc. The main components of the

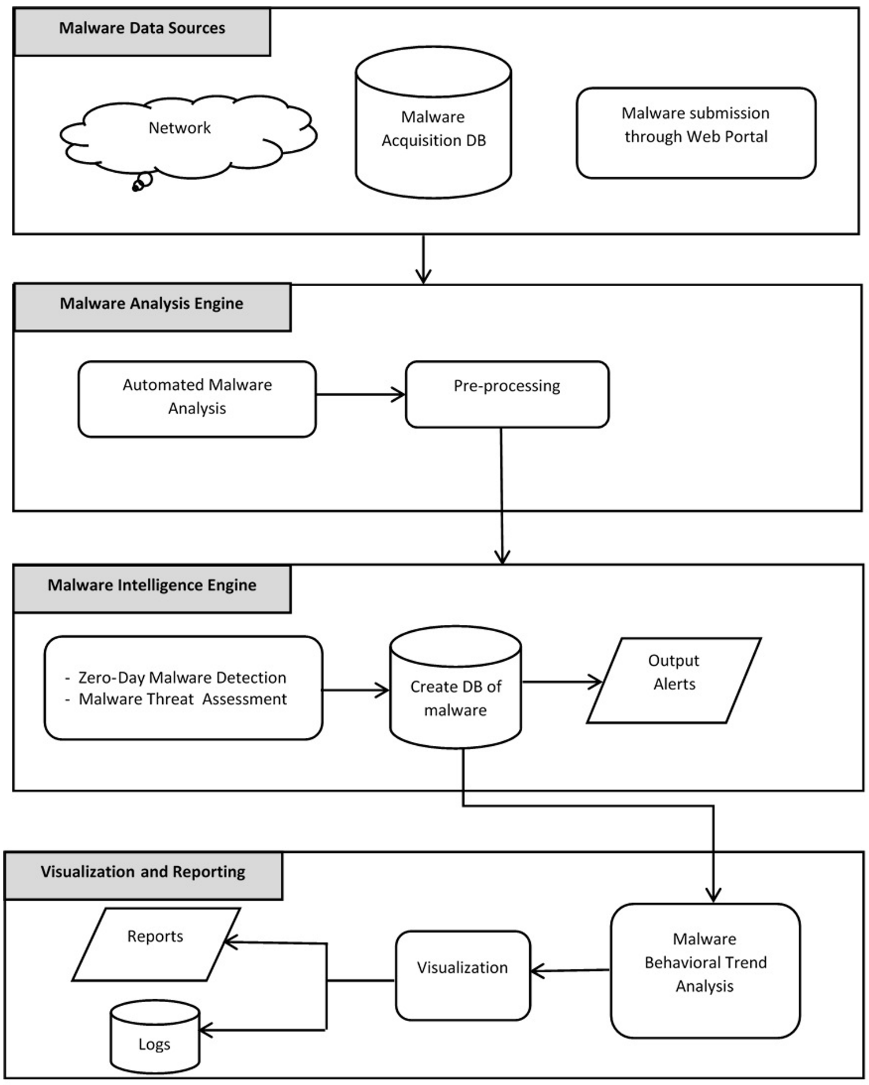
Malware Data Sources
Network
Malware Acquisition DB
Malware submission through Web Portal
Malware Analysis Engine
Automated Malware Analysis
Pre-processing
Malware Intelligence Engine
- Zero-Day Malware Detection
- Malware Threat Assessment
Create DB of malware
Output Alerts
Visualization and Reporting
Reports
Logs
Visualization
Malware Behavioral Trend Analysis

Fig. 2.3. Architecture of proposed framework.

architecture are shown in fig. 2.3 and are discussed in what follows.

2.2.1. Malware Data Sources. Random malware specimens can be acquired from various Virus collection repositories after free registration. These can also be collected by designing a web-portal where the users are permitted to submit suspicious/malicious samples. Another way to collect malicious files is to capture and monitor user's network traffic.

2.2.2. Malware Analysis Engine. Malware analysis is the skill of dissecting malicious programs for understanding their functionalities and potential impact on the victim machine. It is performed using two broad methods: Static malware analysis (code analysis) and dynamic malware analysis (behavioral analysis). Static malware analysis is performed by examining the binary file without executing it. Dynamic malware analysis, on the other hand, is performed by monitoring its behavior while it is being executed in the virtual environment [8]. Academicians and Industrialists working on malicious programs use these techniques to have an understanding of their functionalities & motives and thus the menace level posed by them. With the intention of resolving the challenges raised by the large volume of malware samples, automated malware analysis is

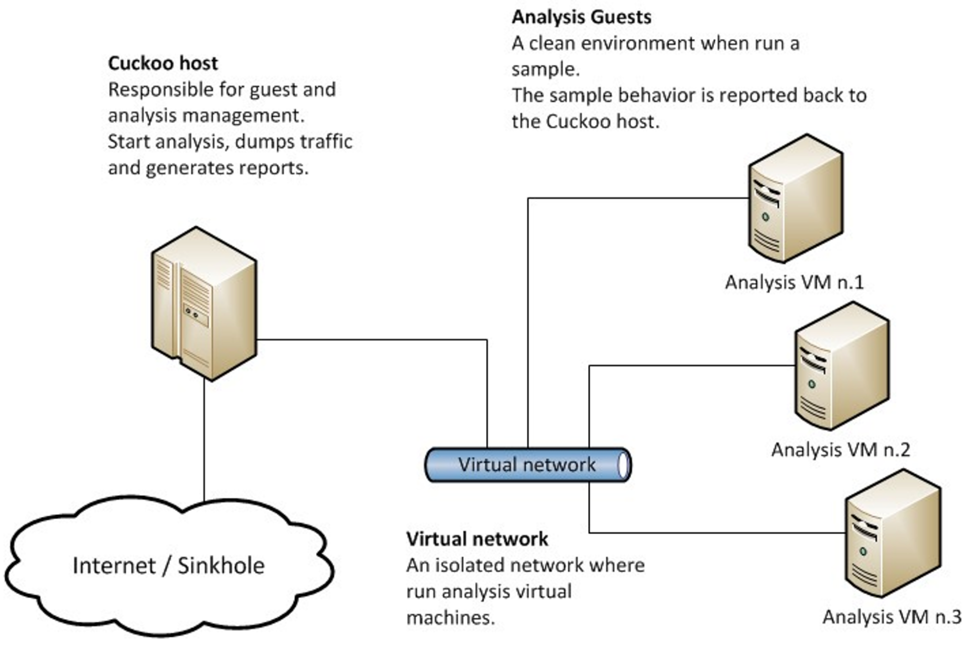
Analysis Guests
A clean environment when run a sample.
The sample behavior is reported back to the Cuckoo host.
Cuckoo host
Responsible for guest and analysis management.
Start analysis, dumps traffic and generates reports.
Analysis VM n.1
Analysis VM n.2
Analysis VM n.3
Virtual network
Internet / Sinkhole
Virtual network
An isolated network where run analysis virtual machines.

Fig. 2.4. Architecture of Cuckoo sandbox [9].

used. It automatically executes the malicious binaries in a sandboxed environment and generates the reports containing both static and dynamic features of malware programs. A comparison of tools and techniques for performing static, dynamic and automated malware analysis is carried out in [9]. It also provides the examples from the literature that use different malware analysis techniques for extracting their behavioral attributes and use machine learning algorithms for discriminating malicious programs from benign ones.

In order to develop the malware intelligence component, malicious binaries targeting Windows operating systems are acquired from Virusshare[1]. These specimens are then scanned using AVAST[2] AV and filtered to keep only the unique MD5 hash files. Subsequently, these are executed automatically (using a python script) in a sandboxed environment set up at Cyber Security Research Centre (CSRC), PEC University of Technology, Chandigarh, India. We used Brad Accuvant[3], a modified version of Cuckoo sandbox[4] for this purpose which offers numerous improvements over regular Cuckoo. It is an open source automated malware analysis system developed in python. It consists of central management software for handling the task of malware analysis. The main constituents of Cuckoo's architecture are: host machine and a number of guest machines as explained in [10]. The host machine runs the core component of the sandbox whose function is to manage the complete analysis process and the guest machines are the isolated environments where malware specimens get executed safely. Fig. 2.4 depicts the high level design of Cuckoo's architecture.

Cuckoo provides a series of Python scripts to hook into and examine malware activities while it is being executed. The system uses CuckooMon as the core element that provides it the capability to monitor a malicious sample by intercepting its execution flow. It is configured on a server with Ubuntu LTS 14.04 as the host machine and Windows XP, SP3 as the guest machine using Oracle VirtualBox[5]. It is available as open-source and has the ability to create multiple isolated virtual machines running different operating systems (shown in fig. 2.5). It has the ability to create current snapshot of complete virtual hard-disk that can be restored to its saved stage at any time. It also includes command line management interface which can perform everything that

[1]www.virusshare.com
[2]www.avast.com
[3]www.github.com/brad-accuvant
[4]www.cuckoosandbox.org
[5]www.virtualbox.org/

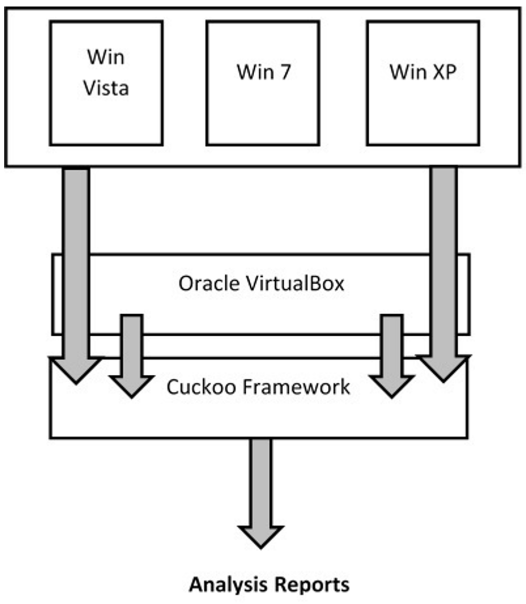

Fig. 2.5. VirtualBox as virtualization software.

can be done using Graphical User Interface (GUI). Due to all these functionalities, VirtualBox is used for the proposed system. It contains command line interface Vboxmanage, which provides the commands to manage virtual machine easily. For instance, to start and stop virtual machine and restore it to clean stage, which is very important while dealing with malware specimens.

After executing the files in the virtual environment, Cuckoo generates malware analysis reports in different formats like Java Script Object Notation (JSON), Hypertext Markup Language (HTML), Portable Document Format (PDF) etc. to make these user friendly. The reports generated are then parsed to obtain the various malware features such as MD5, file name, file size, timestamp of malware creation, packer used, suspicious section, dropped files, mutex creation, file system activities, registry activities, network activities and services created etc. These malware features are then used as a basis for generating malware intelligence.

2.2.3. Malware Intelligence Engine. Providing intelligence of the cybercrime and investigating complex threats is very crucial. Malware intelligence includes samples' analysis report, information regarding detection of new malware specimens, their signatures, behavioral profile and other informative knowledge about the malicious program which can be used for generating early warnings. The following techniques (as a part of this component) have been proposed for generating malware threat intelligence.

- **Zero-Day Malware Detection:** The traditional defenses like AV software and IDS/IPS typically rely on signature-based methods and are unable to detect previously unseen malware to offer immediate protection as these are needed to be analyzed before creating signatures for them. In order to solve this issue, static or dynamic malware analysis is being used along with machine learning algorithms for malware detection and classification [11][12], but the main problem with these systems is that they have high false positive and false negative rate. Another problem is that the process of building classification model takes time (due to large feature set) which hinders the early detection of malware. So the challenge is to select the relevant set of features so that the classification model can be built in less time with high accuracy. A survey conducted by Gandotra et al. [13] on malware analysis and classification clearly shows that a single approach, either static or dynamic can't classify the malware specimens accurately. So, a hybrid technique integrating both static attributes and dynamic behaviors is required for better malware detection and classification. After performing malware analysis, the obtained features are pre-processed and specific features are selected along with machine learning techniques to detect and classify zero-day malware. We proposed a malware classification system [14], which uses integration of both static and dynamic features for distinguishing malware files from clean ones using machine learning algorithms with the aim of achieving excellent classification accuracy. The malware features which are used in this work include suspicious section count and function call frequency (static features) and file & network activities (dynamic features). This malware detection system is able to achieve an accuracy of 99.58% for the machine learning algorithm, Random Forest. In order to improve the classification model building time, we conducted another set of experiments [15] which considers the integration of both static and dynamic analysis features of malware binaries followed by a filter approach for selecting relevant set of features. Static and dynamic analysis features considered together provide high accuracy for distinguishing malware binaries from clean ones and the relevant feature selection process improves the model building time without compromising the accuracy of malware detection system.

- **Malware Threat Assessment:** Industries providing Anti-malware solutions need the skilled analysts and a large number of hardware resources to analyze the malicious programs more closely. The malware samples are increasing in volume day by day which makes it difficult to assign such a large number of resources for conducting a closer analysis. Thus, it is required to find a way to prioritize the newly discovered malicious programs for assigning such resources in order to conduct closer manual analysis. This component is designed to compute the threat or damage posed by a piece of malware (to a victim machine) automatically as soon as it appears in the wild [16]. The first step of this component involves a novel categorization of malware behaviors. This categorization is done using MAEC[6] (Malware Attribute Enumeration and Characterization) framework. The second step involves computing the impact of each behavior on the victim machine. It is done by assigning a weight to each behavior on the basis of its adverse effects on Confidentiality, Integrity and Availability of the target system and it's resources. The malicious programs use evasion techniques to evade their detection and to avoid the static and dynamic malware analysis. The evasion techniques used by each malware are reviewed to find their sophistication level. The third step involves designing a Fuzzy Inference System (FIS). We used Mamdani model [17], which is the most popularly used for representing human reasoning. It describes the association between fuzzy variables. The designed FIS takes malware behavioral impact and the sophistication level as the input variables and provide the threat level of a malware as the output variable. Based on these threat levels, the malicious programs can be prioritized for allocating resources for conducting a closer manual analysis.

2.2.4. Visualization and Reporting. The intelligence generated in the modules of above mentioned component is visualized and the reports generated regarding malware behavior are stored in the database repository. This behavioral information can be further analyzed to have an insight into the changing trends of malware behavior. We conducted a statistical trend analysis of behavioral attributes of about 0.1 million historical malware specimens collected from diverse sources [18]. We also highlighted the challenges evolving out of these trends and provided the future research directions to malware analysts and security researchers. The intelligent information or insight so obtained about malware threats can be shared with Computer Emergency Response Teams (CERTs), security agencies and other stakeholders near real-time and thus act as an early warning system. This information can be shared with the stakeholders using threat Information sharing Standards like Structured Threat Information Expression (STIX) [19], Trusted Automated Exchange of Indicator Information (TAXII) [20] and Cyber Observable Expression (CybOX) [21], which are the efforts to enable automated cyber threat information sharing across various security organizations and stakeholders.

3. Framework Realization. This section explains the integration of all the tools and methodologies used for testing the proposed framework. It is realized by designing a prototype capable of providing security-as-a-service. The Squid proxy server [7] has been used for capturing the network traffic while implementing the proposed framework. The complete workflow of the system has been divided into two halves: Online Mode and Offline Mode and is depicted in the fig. 3.1. The step by step description of both the modes along with their implementation is discusses as follows.

[6]https://maec.mitre.org/
[7]www.squid-cache.org

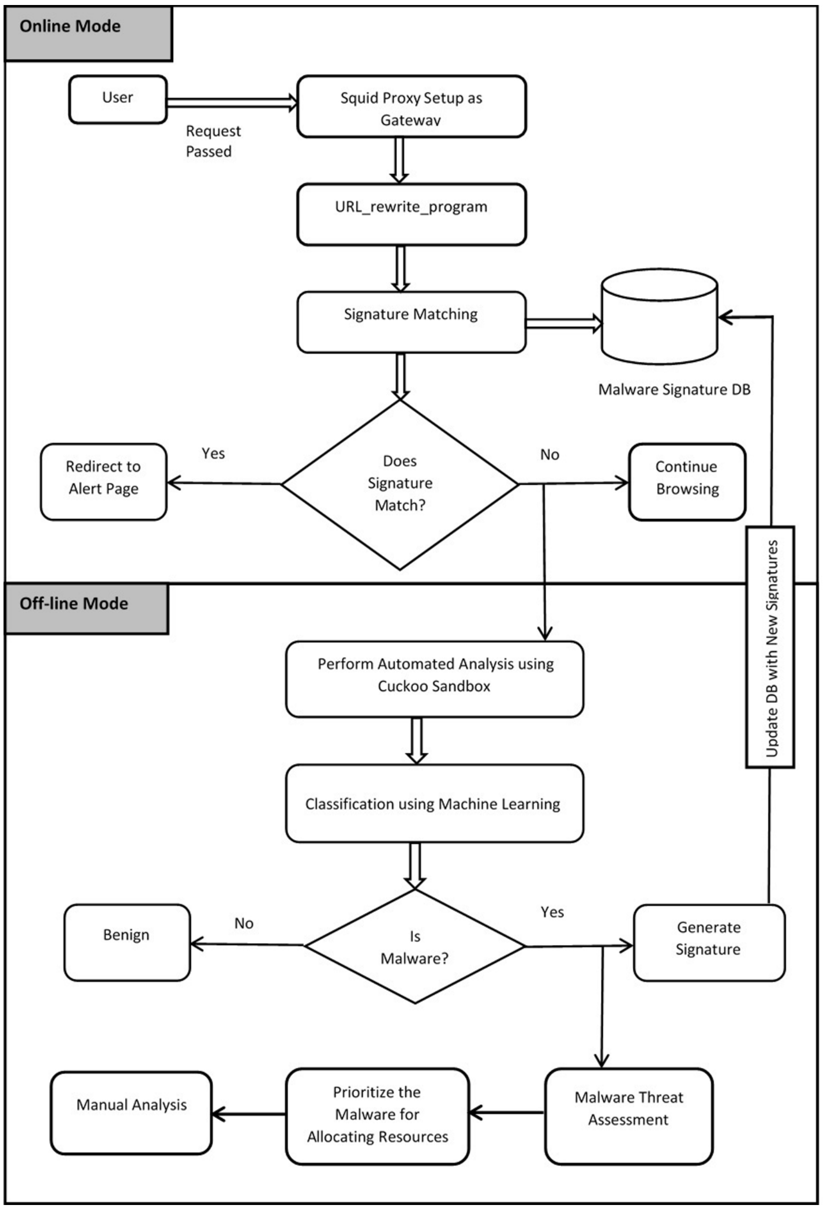

FIG. 3.1. System prototype for providing security-as-a-service.

Online Mode

The first step in the process is to capture the Internet traffic, especially the URLs the user is accessing. For this purpose, a Squid proxy server is installed in the transparent mode. The entire URL requests made by the users are captured by Squid, and forwarded to the url_rewrite_program to process the URL before request gets completed.

- After the url_rewrite_program receives the URL requested by the user, the complete content of that URL is fetched (including its css, JavaScript, images, .exe and everything else it contains). These fetched contents are then verified against set of viruses' signatures that are acquired from an open source AV engine, AVAST. If signature matches, user is redirected to an Alert Page (as explained in next step) else the user is allowed to continue browsing and for detecting zero-day malware in the suspicious traffic,

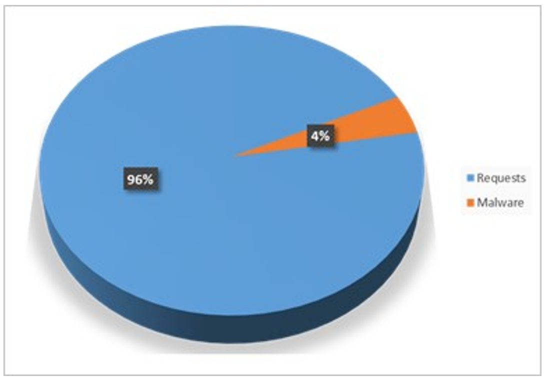

FIG. 3.2. Malicious requests against total requests.

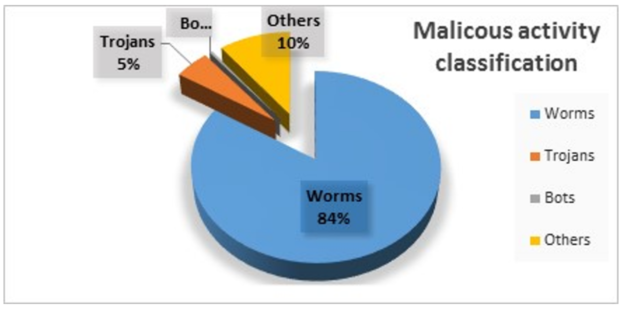

FIG. 3.3. Malware family classification.

the control goes to the offline mode.

- After it is found that the user is accessing some malicious content, instead of completing the request, user is warned by redirecting to alert page which has been hosted on the same system using Apache2 server and Python flask web development environment. It gives the details about what URL user tried to access, from which IP address and what was the kind of the malware.

Offline Mode

- In the offline mode, an automated analysis is performed for the suspicious files. In this process, every sample is made to execute automatically through a python script in a virtual environment created using Cuckoo sandbox and virtualbox which generates the comprehensive information in the form of malware analysis reports.
- These reports are then analyzed for selecting relevant set of features which are then used to predict whether the content is malicious or not using machine learning algorithms. If a new malware is detected at this stage, then its signature is generated and the existing malware signature database (in the online mode) is updated so that it can be detected in real-time next time.
- After classifying malware into different classes, the damage capability level of a malware is computed which can help in providing early warnings about it so that immediate attention could be paid to it in terms of allocating resources for performing closer analysis.

For real-time testing purpose, the system was deployed at the premises of STPI (Software Technology Parks of India), Mohali and real-time Internet traffic was routed through it for 2 hours. A total of 1,84,000 URL requests were routed through the system. These were analyzed and results were logged in a log file for further analysis and reporting purpose. It is found that out of all analyzed requests, 4% were detected to be malicious (fig. 3.2). Out of the total malicious activities, more than 80% were worms (fig. 3.3).

From the log files generated while monitoring network traffic, other type of analysis can be performed in terms of number of total malware detected, number of unknown malware detected, total number of attacks etc.

The changing trends of malware attack can be studied to have an idea about the future attacks and preventive measures to be taken to stop these attacks or to implement risk mitigation strategies.

4. Conclusion. In this era, the Internet consists of dynamic, virtualized and scalable resources which are provided as services. Due to the tremendous increase in these services over the Internet, an easy way opens up for attackers to carry out malicious activities. As a result the end-users have to deal with a flux of new malware threats, which are persistent, stealthy and advanced. The traditional anti-malware solutions are not able to deal with the current threat landscape. So, there is a need to shift from traditional cyber security to cyber threat intelligence. The intelligence must be collected from multiple sources, analyzed efficiently and shared in standard formats within no time so that it can be leveraged and included into security systems for taking preventive measures. This paper discussed the high level design and the detailed architecture of the framework for generating malware threat intelligence. The intelligent information generated can be shared with security agencies so that they can issue advisories and preventive measures to deal with future malware threats.

REFERENCES

[1] U. Bayer, A. Moser, C. Krugel, and E. Kirda, *Dynamic analysis of malicious code*, Journal of Computer Virology 2 (2006), pp. 67-77.

[2] P. Mell, K. Kent, J. Nusbaum, *Guide to Malware Incident Prevention and Handling*, National Institute of Standards and Technology (NIST), 2005.

[3] Symantec, *Internet Security Threat Report* , 21 (2016), available at: https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

[4] S. Singla, E. Gandotra, D.Bansal, and S. Sofat,*Detecting and Classifying Morphed Malwares: A Survey*, International Journal of Computer Applications, 122 (2015).

[5] Webroot, *Threat Intelligence: What is it, and How Can it Protect You from Today's Advanced Cyber-Attacks*, available at: https://www.gartner.com/imagesrv/media-products/pdf/webroot/issue1_webroot.pdf.

[6] SecureWorks, *Threat Intelligence*, available at: https://www.secureworks.com/capabilities/threat-intelligence.

[7] E. Gandotra, D. Bansal, S. Sofat, *Computational techniques for predicting cyber threats*, In Intelligent Computing, Communication and Devices, Springer (2015), pp. 247-253.

[8] M. Sikorski, and A. Honig, *Practical malware analysis: the hands-on guide to dissecting malicious software*, No starch press (2012).

[9] E. Gandotra, D. Bansal, and S. Sofat, *Tools & Techniques for Malware Analysis and Classification*, International Journal of Next-Generation Computing, 7 (2016), pp.176-197.

[10] *Cuckoo Sandbox Book*, available at: http://docs.cuckoosandbox.org/en/latest/.

[11] A.Saini, E. Gandotra, D. Bansal, and S. Sofat, *Classification of PE files using static analysis*, Proc. of 7th International Conference on Security of Information and Network, ACM, (2014).

[12] S. Singla, E. Gandotra, D.Bansal, and S. Sofat,*A Novel Approach to Malware Detection using Static Classification*, International Journal of Computer Science and Information Security, 13 (2015).

[13] E. Gandotra, D. Bansal, and S. Sofat,*Malware Analysis and Classification: A Survey*, Journal of Information Security, 5(2014), pp. 56-65.

[14] E. Gandotra, D. Bansal, S. Sofat, *Integrated Framework for Classification of Malwares*, Proc. of 7th International Conference on Security of Information and Network, ACM, (2014).

[15] E. Gandotra, D. Bansal, S. Sofat, in press *Zero-Day Malware Detection*, Proc. of 6th international symposium on embedded computing and system design, IEEE, IIT Patna, India, (2016).

[16] E. Gandotra, D. Bansal, S. Sofat, *Malware Threat Assessment Using Fuzzy Logic Paradigm*, Cybernetics and Systems, 48(2016), pp. 29-48.

[17] E. H. Mamdani and S. Assilian, *An experiment in linguistic synthesis with a fuzzy logic controller*, International Journal of Man-Machine Studies, 7(1975), pp. 1-13.

[18] E. Gandotra, D. Bansal, S. Sofat, in press *Malware intelligence: Beyond malware analysis*, International Journal of Advanced Intelligence Paradigms.

[19] S. Barnum, *Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX)*, The MITRE Corporation,(2012).

[20] J. Connolly, M. Davidson, M. Richard, M., and C. Skorupka, *The trusted automated exchange of indicator information (taxii)*, The MITRE Corporation, 2013.

[21] E. Casey, G. Back, G., and S. Barnum, *Leveraging CybOX to standardize representation and exchange of digital forensic information Digital Investigation*,Digital Investigation, 12(2015), pp. S102-S11012.

Edited by: Kavita Sharma
Received: Nov 12, 2016
Accepted: Jul 3, 2017