SEMI-SUPERVISED MACHINE LEARNING APPROACH FOR DDOS DETECTION

Mondi Surya Prabha , G.M.Padmaja, (Assistant Professor)

Department of Computer Science and Engineering , Raghu Institute Of Technology , Visakhapatnam , AP , India .

Abstract

The appearance of malicious apps is a serious threat to the Android platform. Most types of network interfaces based on the integrated functions, steal users' personal information and start the attack operations. In this project, we propose an effective and automatic malware detection method using the text semantics of network traffic. In particular, we consider each HTTP flow generated by mobile apps as a text document, which can be processed by natural language processing to extract text-level features. Later, the use of network traffic is used to create a useful malware detection model. We examine the traffic flow header using N-gram method from the natural language processing (NLP). Then, we propose an automatic feature selection algorithm based on chi-square test to identify meaningful features. It is used to determine whether there is a significant association between the two variables. We propose a novel solution to perform malware detection using NLP methods by treating mobile traffic as documents. We apply an automatic feature selection algorithm based on N-gram sequence to obtain meaningful features from the semantics of traffic flows. Our methods reveal some malware that can prevent detection of antiviral scanners. In addition, we design a detection system to drive traffic to your own-institutional enterprise network, home network, and 3G / 4G mobile network. Integrating the system connected to the computer to find suspicious network behaviors.

---

1. INTRODUCTION

Data Mining

Data mining techniques have been used to develop sophisticated intrusion detection systems for the last two decades. Artificial Intelligence, Machine Learning (ML), Pattern Recognition, Statistics, Information Theory are the most used data mining techniques for intrusion detection With the increase in dependability of the internet comes with it an important challenge: data availability. Data availability is a key requirement for a network system to be considered secure. Distributed denial of service attacks are intentional attempts by malicious users to disrupt or degrade the quality of a network or service These attacks involve a number of compromised connected online devices, The use of botnets makes it easier for attackers is to launch massive attacks due to the fact that they harness the power of a lot of devices for an attack. Attacks involving botnets also make it difficult to determine the exact source of the attack. Differentiating between flash crowds also poses a major challenge. There are two main methods to launch DDoS attacks in the Internet. The first method is for the attacker to send some malformed packets to the victim to confuse a protocol or an application running on it (i.e., vulnerability attack. The other method, which is the most common one, involves an attacker trying to do one or both of the following: Fig.1. DDOS detection life cycle There are several phases that are involved in defensing DDoS attack, Prevention.)e prevention phase focuses on protecting a system against an attack by applying appropriate security appliances at varied places. Besides that, prevention also

protects server resources and ensures that online services are ready to surf the genuine client. Mitigation.)e mitigation phase is applied when an attack occurs, and a suitable security countermeasure is executed to handle the attack or to slow down the attack. A mitigation technique operates by stopping the attack. Detection.)e detection phase requires analysis of the running system to discover malicious traffic that leads to DDoS attack. Detection involves a sophisticated approach to identify large illegal GET request traffic against a web server. Most of the detection techniques were applied to form DDoS detection known as pattern matching, clustering, statistical methods, deviation analysis, associations, and correlation. Monitoring. As for the monitoring phases, necessary information about a host or network is obtained by using tools, such as network monitoring software. Monitoring is conducted in real time as it becomes compulsory for detection of DDoS attack. A process of monitoring becomes complicated when the attacker utilized botnet that is situated at multiple locations around the world to launch DDoS attack at a minimal rate.

SYSTEM ANALYSIS

Literature survey:

[1] An empirical evaluation of information metrics for low-rate and high-rate ddos attack detection.

Distributed Denial of Service (DDoS) attacks represent a major threat to uninterrupted and efficient Internet

service. In this paper, we empirically evaluate several major information metrics, namely, Hartley entropy, Shannon entropy, Renyi's entropy, generalized entropy, Kullback–Leibler divergence and generalized information distance measure in their ability to detect both low-rate and high-rate DDoS attacks. These metrics can be used to describe characteristics of network traffic data and an appropriate metric facilitates building an effective model to detect both low-rate and high-rate DDoS attacks. We use MIT Lincoln Laboratory, CAIDA and TUIDS DDoS datasets to illustrate the efficiency and effectiveness of each metric for DDoS detection.

[2] Constructing detection knowledge for ddos intrusion tolerance

Intrusion tolerance is the ability of a system to continue providing (possibly degraded but) adequate services after a penetration. With the rapid development of network technology, distributed denial of service (DDoS) attacks become one of the most important issues today. In this paper, we propose a DDoS ontology to provide a common terminology for describing the DDoS models consisting of the Profile model(the representation of the behaviors of system and users) and the Defense model (the descriptions of Detection and Filter methodologies). Also, the Evaluation strategy based upon current statuses of users' behaviors is used to evaluate the degree of the intrusion tolerance of the proposed models during DDoS attacks. Based upon the ontology, four KCs (Profile model, Evaluation strategy, Detection methodology, and Filter methodology Knowledge Classes) and their relationships are then proposed, where each KC may contain a set of sub-KCs or knowledge represented as a natural rule format. For an arbitrarily given network environment, the default knowledge in the Profile KC and the Evaluation KC, the appropriate detection features in the Detection KC, and the suitable access control list policies in the Filter KC can be easily extracted and adopted by our proposed integrated knowledge acquisition framework. We are now implementing a NORM-based DDoS intrusion tolerance system for DDoS attacks to evaluate the proposed models.

[3] Defending against flooding-based distributed denial-of-service attacks:

Flooding-based distributed denial-of- service (DDoS) attack presents a very serious threat to the stability of the Internet. In a typical DDoS attack, a large number of compromised hosts are amassed to send useless packets to jam a victim or its Internet connection, or both. In the last two years, it is discovered that DDoS attack methods and tools are becoming more sophisticated, effective, and also more difficult to trace to the real attackers. On the defense side, cur-rent technologies are still unable to with stand large-scale attacks. The main purpose of this article is therefore twofold. The first one is to describe various DDoS attack methods, and to present a systematic review and evaluation of the existing defense mechanisms. The second is to discuss a

longer-term solution, dubbed the Inter-net- firewall approach, that attempts to intercept attack packets in the Internet core, well before reaching the victim

[4] Distributed denial of service attack and defense.

This brief provides readers a complete and self-contained resource for information about DDoS attacks and how to defend against them. It presents the latest developments in this increasingly crucial field along with background context and survey material. The book also supplies an overview of DDoS attack issues, DDoS attack detection methods, DDoS attack source trace back, and details on how hackers organize DDoS attacks. The author concludes with future directions of the field, including the impact of DDoS attacks on cloud computing and cloud technology. The concise yet comprehensive nature of this brief makes it an ideal reference for researchers and professionals studying DDoS attacks. It is also a useful resource for graduate students interested in cyberterrorism and networking

[5] Four decades of data mining in network and systems management.

How has the interdisciplinary data mining field been practiced in Network and Systems Management (NSM)? In Science and Technology, there is a wide use of data mining in areas like bioinformatics, genetics, Web, and, more recently, astro informatics. However, the application in NSM has been limited and inconsiderable. In this article, we provide an account of how data mining has been applied in managing networks and systems for the past four decades, presumably since its birth. We look into the field's applications in the key NSM activities—discovery, monitoring, analysis, reporting, and domain knowledge acquisition. In the end, we discuss our perspective on the issues that are considered critical for the effective application of data mining in the modern systems which are characterized by heterogeneity and high dynamism.

EXISTING SYSTEM:

The first phase of their approach consists of dividing the incoming network traffic into three type of protocols TCP, UDP or Other. Then classifying it into normal or anomaly traffic. In the second stage a multi-class algorithm classify the anomaly detected in the first phase to identify the attacks class in order to choose the appropriate intervention. Two public datasets are used for experiments in this paper namely the UNSW-NB15 and the NSL-KDD Several approaches have been proposed for detecting DDoS

attack. Information theory and machine learning are theThe performances of network intrusion detection approaches, in general, rely on the distribution characteristics of the under laying network traffic data used for assessment. The DDoS detection approaches in the literature are under two main categories unsupervised approaches and supervised approaches. Depending on the benchmark datasets used, unsupervised approaches often suffer from high false positive rate and supervised approach cannot handle large amount of network traffic data and their performances are often limited by noisy and irrelevant network data. Therefore, the need of combining both, supervised and unsupervised approaches arises to overcome DDoS detection issues.

DISADVANTAGES:

The datasets above are split into train subsets and test subsets using a configuration of 60% and 40% respectively. The train subsets are used to fit the Extra-Trees ensemble classifiers and the test subsets are used to test the entire proposed approach. Before fitting the classifiers the train subsets are normalized using the Min Max method

- This section presents the details of the proposed approach and the methodology followed for detecting

the DDoS attack. The proposed approach consists of five major steps: Datasets preprocessing, estimation of network traffic Entropy, online co-clustering, information gain ratio

- The aim of splitting the anomalous network traffic is to reduce the amount of data to be classified by excluding the normal cluster for the classification. For DDoS detection normal traffic records are irrelevant and noisy as the normal behaviors continue to evolve. Most of the time the new unseen normal traffic instances cause the increase of the false positive rate and the decrease of the classification accuracy. Hence, excluding some noisy normal instances of the network traffic data for classification is beneficial in terms of low false positive rates and classification accuracy. Assuming that after the network traffic clustering one cluster contains only normal traffic, a second one contains only DDoS traffic and a third one contains both DDoS and normal traffic.

- **PROPOSED SYSTEM:**
    - This sections introduces our methodology to detect the DDoS attack. The five-fold steps application process of data mining techniques in network systems discussed in characterizes the followed methodology. The main aim of combining algorithms used in the proposed approach is to reduces noisy and irrelevant network traffic data before preprocessing and classification stages for DDoS detection while maintaining high performance in terms of accuracy, false positive rate and running time, and low resources usage. Our approach starts with estimating the entropy of the FSD features over a time-based sliding window. When the average entropy of a time window exceeds its lower or upper thresholds the co-clustering algorithm split the received network traffic into three clusters. Entropy estimation over time sliding windows allows to detect abrupt changes in the incoming network traffic distribution which are often caused by DDoS attacks. Incoming network traffic within the time windows having abnormal entropy values is suspected to contain DDoS traffic. The focus only on the suspected time windows allows to filter important amount of network traffic data, therefore only relevant data is selected for the remaining steps of the proposed approach. Also, important resources are saved when no abnormal entropy occurs. In order to determine the normal cluster, we estimate the information gain ratio based on the average entropy of the FSD features between the received network traffic data during the current time window and each one of the obtained clusters. As discussed in the previous section during a DDoS period the generated amount of attack traffic is largely bigger than the normal traffic. Hence, estimating the information gain ratio based on the FSD features allows to identify the two cluster that preserve more information about the DDoS attack and the cluster that contains only normal traffic. Therefore, the cluster that produce lower information gain ratio is considered as normal and the remaining clusters are considered as anomalous. The information gain ratio is computed for each cluster as follows:

3.2.1 ADVANTAGE:

- Where *subset w*represents the received subset of network data during the time window *w*, *Ci (i* = 1, 2, 3*)* are the obtained clusters from *subset w*and |*Ci*| is the size of the *ith* cluster. *Avg H(subset)* is the average entropy of the FSD features of the input *subset* and |*subset* | represents the size

The clustering of the incoming network traffic data allows to reduce important amount of normal and noisy data before the preprocessing and classification steps. More than6% of a whole traffic dataset can be filtered

SYSTEM DESIGN

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use- case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted

INPUT AND OUTPUT DESIGN INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those

steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

Objectives

1. Input Design is the process of converting a user-oriented description of the input into a computer- based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

OUTPUT DESIGN

Output Design A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decisionmaking.

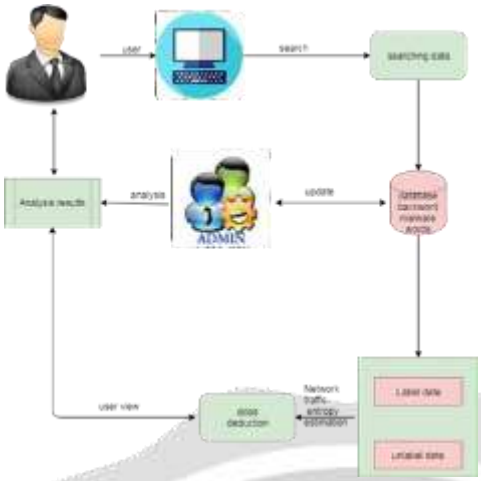
1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When 42 analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system. The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the ❖ Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

System Architecture:



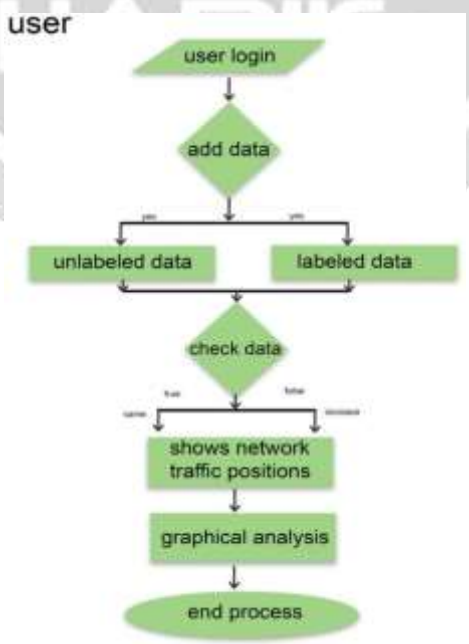
Data Flow Diagram:

1. The DFD is also called as bubble chart . It is a simple graphical formalism that can be used to represent a system in terms in terms of inputs data to the system, various processing carried out on this data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modelling tools. It is used to model the system components. These components are the systems process, the data used by the process, an external entity that interacts with the system and the information flows in the system. 3.DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts 44 information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

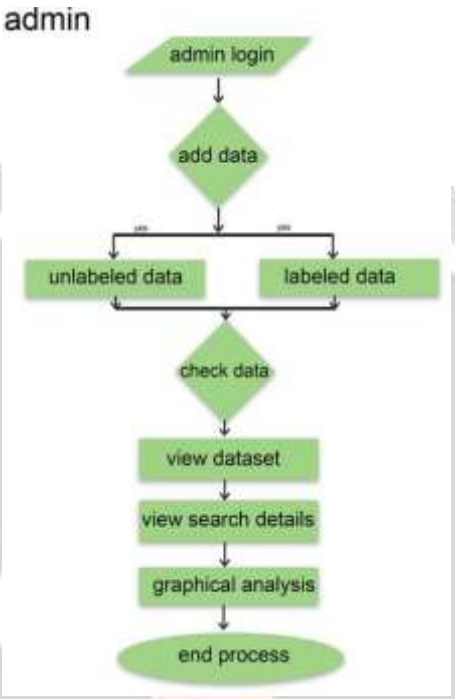
User:



Description

User will login and add some manual data and submit it, after that user will check whether it was entered labelled or unlabelled data. User will check the data. If user entered unlabelled data network traffic position is correct. If user entered labelled data network traffic position increases their position. After that shows graphical analysis and end process.

Admin:



Description

Admin will login and add some manual data and submit it, after that Admin will check whether it was entered labelled or unlabelled data. Admin will check the data and view dataset and view search details. If user entered unlabelled data network traffic position is correct. After that shows graphical analysis and end process.

SYSTEM IMPLEMENTATION PYTHON INTRODUCTION:

Python is a general purpose, dynamic, high level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development.

Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles.

Python is not intended to work in a particular area, such as web programming.
That is why it is known

as multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc.

We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable.

Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

Python 2 vs. Python 3

In most of the programming languages, whenever a new version releases, it supports the features and syntax of the existing version of the language, therefore, it is easier for the projects to switch in the newer version. However, in the case of Python, the two versions Python 2 and Python 3 are very much different from each other.

A list of differences between Python 2 and Python 3 are given below:

Python 2 uses **print** as a statement and used as print "something" to print some string on the console. On the other hand, Python 3 uses **print** as a function and used as print("something") to print something on the console.

Python 2 uses the function raw_ input() to accept the user's input. It returns the string representing the value, which is typed by the user. To convert it into the integer, we need to use the int() function in Python. On the other hand, Python 3 uses input() function which automatically interpreted the type of input entered by the user.
However, we can cast this value to any type by using primitive functions (int(), str(), etc.).

In Python 2, the implicit string type is ASCII, whereas, in Python 3, the implicit string type is Unicode.

Python 3 doesn't contain the xrange() function of Python 2. The xrange() is the variant of range() function which returns a xrange object that works similar to Java iterator. The range() returns a list for example the function range(0,3) contains 0, 1, 2.

DJANGO

Introduction

Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement.

This framework uses a famous tag line: **The web framework for perfectionists with deadlines.**

By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only.

History

Django was design and developed by Lawrence journal world in 2003 and publicly released under BSD license in July 2005. Currently, DSF (Django Software Foundation) maintains its development and release cycle.

Django was released on 21, July 2005. Its current stable version is 2.0.3 which was released on 6 March, 2018.

Features of Django Rapid Development

Secure Scalable Fully loaded Versatile Open Source
Vast and Supported Community

Rapid Development

Django was designed with the intention to make a framework which takes less time to build web application. The project implementation phase is a very time taken but Django creates it rapidly.

Secure

Django takes security seriously and helps developers to avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery etc. Its user authentication system provides a secure way to manage user accounts and passwords.

Scalable

Django is scalable in nature and has ability to quickly and flexibly switch from small to large scale application project.

Fully loaded

Django includes various helping task modules and libraries which can be used to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds etc.

Versatile

Django is versatile in nature which allows it to build applications for different-different domains. Now a days, Companies are using Django to build various types of applications like: content management systems, social networks sites or scientific computing platforms etc.

Open Source

Django is an open source web application framework. It is publicly available without cost. It can be downloaded with source

code from the public repository. Open source reduces the total cost of the application development.

Vast and Supported Community Django is an one of the most popular web framework. It has widely supportive community and channels to share and connect.

METHODOLOGIES:

There are three modules can be divided here for this project they are listed as below

- User Apps
- DDOS Attack Deduction
- Co-Clustering algorithm

- Classifications of DDOS attack
- Graphical analysis

From the above four modules, project is implemented. Bag of discriminative words are achieved

1. User Apps

User handling for some various times of smart phones ,desktops laptops and tablets .If any kind of devices attacks for some unauthorized Malware software's. In this Malware on threats for user personal dates includes for personal contact, bank account numbers and any kind of personal documents are hacking in possible.

2. DDOS Attack Deduction

User search the any link Notably, not all network traffic data generated by malicious apps correspond to malicious traffic. Many malware take the form of repackaged benign apps; thus, Malware can also contain the basic functions of a benign app. Subsequently, the network traffic they generate can be characterized by mixed benign and malicious network traffic. We examine the traffic flow header using Co- clustering algorithm from the natural language processing (NLP).

3. Co-clustering algorithm:

Co-clustering algorithm performs a simultaneous clustering of rows and columns of a data matrix based on a specific criterion. It produces clusters of rows and columns which represent sub-matrices of the original data matrix with some desired properties. Clustering simultaneously rows and columns of a data matrix yields three major benefits: Dimensionality reduction, as each cluster is created based on a subset of the original features. More compressed data representation with preservation of information in the original data. Significant reduction of the clustering computational complexity. The co-clustering computational complexity is $O(mkl+ nkl)$ which is much smaller than that of the traditional K means algorithm $O(mnk)$. Where m is the number of rows, n is the number of columns, k is the number of clusters and l is the number of column clusters.

4. Classifications of DDOS Attack: Here, we compare the classification performance of Co-clustering algorithm with other popular machine learning algorithms. We have selected several popular classification algorithms. For all algorithms, we attempt to use multiple sets of parameters to maximize the performance of each algorithm. Using Co-clustering algorithm algorithms classification for malware bag-of-words weightage.

5. Graphical analysis

The graph analysis is done by the values taken from the result analysis part and it can be analyzed by the graphical representations. Such as pie chart, pyramid chart and funnel chart here in this project.

MALWARE DETECTION METHOD

Malware detection focuses on detecting intrusions by monitoring the activity of

systems and classifying it as normal or anomalous. The classification is often based on machine learning algorithms that use rules to detect misuse, rather than patterns or signatures. One of its shortcomings is that it tends to have a high false positive rate, such that many legitimate actions are classified as intrusive, and that it requires useful training data.

TEXT SEMANTICS

Textual semantics is not a school or movement in linguistics or in literary criticism. It can be described as a continuation from and remodeling of the first European structuralism. It is not so much a *theory* as a praxeology, born from the interpretation of texts as practiced.

NLP - extracting text level

Information extraction is a powerful NLP concept that will enable you to parse through any piece of text.

N-GRAM MODEL

An ***n*-gram model** is a type of probabilistic language model for predicting the next item in such a sequence in the form of a ($n-1$)–order Markov model.[2] *n*-gram models are now widely used in probability, communication theory, computational linguistics (for instance, statistical natural language processing), computational biology (for instance, biological sequence analysis), and data compression. Two benefits of *n*-gram models (and algorithms that use them) are simplicity and scalability – with larger *n*, a model can store more context with a well-understood space–time trade- off, enabling small experiments to scale up efficiently.

AUTOMATIC FEATURE SELECTION

The goal of feature selection is to find the best set of features that allows one to build useful models of studied phenomena.

The techniques for feature selection can be broadly classified into the following categories:

Supervised Techniques: These techniques can be used for labelled data, and are used to identify the relevant features for increasing the efficiency of supervised models like classification and regression.

Unsupervised Techniques: These techniques can be used for unlabelled data.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning

properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application
.it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process,

application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.
Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent.
Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:

Valid Input:

Identified classes of valid input must be accepted.

Invalid Input:

Identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive 54 processes must be considered for testing.
Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets

requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre- driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

SYSTEM PROTOCOLS

IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers.

These are used to give a client/server model

- see later.

NTP

(NTP) is a protocol that helps the computers clock times to be synchronized in a network. This protocol is an application protocol that is responsible for the synchronization of hosts on a TCP/IP network. NTP was developed by David Mills in 1981 at the University of Delaware. This is required in a communication mechanism so that a seamless connection is present between the computers.

HTTP

HTTP is a protocol which allows the fetching of resources, such as HTML documents. It is the foundation of any data exchange on the Web and it is a client- server protocol, which means requests are initiated by the recipient, usually the Web browser. A complete document is reconstructed from the different sub- documents fetched, for instance text, layout description, images, videos, scripts, and more.

CONCLUSION:

Android is a new and fastest growing threat to malware. Currently, many research methods and antivirus scanners are not hazardous to the growing size and diversity of mobile malware. As a solution, we introduce a solution for mobile malware detection using network traffic flows, which assumes that each HTTP flow is a document and analyzes HTTP flow requests using NLP string analysis. The N-Gram line generation, feature selection algorithm, and SVM algorithm are used to create a useful malware detection model. Our evaluation demonstrates the efficiency of this solution, and our trained model

BIBILOGRAPHY :

1. Bhuyan MH, Bhattacharyya DK, Kalita JK (2015) An empirical evaluation of information metrics for low-rate and high- rate ddos attack detection. Pattern Recogn Lett 51:1–7

2. Lin S-C, Tseng S-S (2004) Constructing detection knowledge for ddos intrusion tolerance. Exp Syst Appl 27(3):379–390

3. Chang RKC (2002) Defending against flooding-based distributed denial-of-service attacks: a tutorial. IEEE Commun Mag 40(10):42–51 4. Yu S (2014) Distributed

denial of service attack and defence. Springer, Berlin

5. Wikipedia (2016) 2016 dyn cyberattack. https://en.wikipedia.org/ wiki/2016 Dyn cyberattack. (Online; accessed 10 Apr 2017)

6. theguardian (2016) Ddos attack that disrupted internet was largest of its kind in history, experts say. https://www.theguardian.com/ technology/2016/oct/26/ddos-attack-dyn- mirai-botnet. (Online; accessed 10 Apr 2017)

7. Kalegele K, Sasai K, Takahashi H, Kitagata G, Kinoshita T (2015) Four decades of data mining in network and systems management. IEEE Trans Knowl Data Eng 27(10):2700–2716

8. Han J, Pei J, Kamber M (2006) What is data mining. Data mining: concepts and techniques. Morgan Kaufinann

9. Berkhin P (2006) A survey of clustering data mining techniques. In: Grouping multidimensional data. Springer, pp 25–71

10. Mori T (2002) Information gain ratio as term weight: the case of summarization of ir results. In: Proceedings of the 19th international conference on computational linguistics, vol 1. Association for Computational Linguistics, pp 1–7

11. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63(1):3–42

12. Tavallaee M, Bagheri E, Lu W, Ghorbani A-A (2009) A detailed analysis of the kdd cup 99 data set. In: Proceedings of the second IEEE symposium on computational intelligence for security and defence applications 2009

13. Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA (2012) Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput Secur 31:357–374

14. Moustafa N, Slay J (2015) Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: Military communications and information systems conference (MilCIS), 2015. IEEE, pp 1–6

15. Moustafa N, Slay J (2016) The evaluation of network anomaly detection systems: statistical analysis of the unsw- nb15 data set and the comparison with the kdd99 data set. Inf Secur J: Glob Perspect 25:18–31s.