Digital Forensic Investigation of Access Log in Cloud Environment

Amrendra Narayan Mishra[1], Prof. Ratnesh Kumar Dubey[2], Dr. Vineet Richharia[3]
M Tech Scholar, Research Guide, HOD
Department of Computer Science and Engineering, LNCT, Bhopal

ABSTRACT- Attack surface for the attacker is enlarging with the growing popularity of the internet applications such as social networking, e-commerce, shopping, etc. It could results in the leakage of user's information, violence in digital world etc. It impacts on clients as well as reputation of the organization. Cloud is a very easy way to reach any system. If confidential data is not properly protected, then it becomes opens to vulnerable access and misuse. Cloud forensics relates to cyber-crime on the Internet. Some criminal activities like Leak of personnel images, child pornography, hacking, and identity theft can be traced and the criminals can be punished if proper evidence is found against them. Hence, detailed forensic analysis of cloud computing is required to come to a conclusion about an incident and to prove or disprove someone's guilt. This work proposed a supervised learning method to forensically investigate the abnormal behavior of web based malware through analyzing the log file entries. Also, comparative analysis of supervised machine learning approaches which includes Naïve-bayes, Random Forest, Decision Tree and Random Tree is done.

I. INTRODUCTION

Attack surface for the attacker is enlarging with the growing popularity of the internet applications such as social networking, e-commerce, shopping, etc. It could results in the leakage of user's information, violence in digital world etc. It impacts on clients as well as reputation of the organization. As well, malware to attack the web is growing swiftly, which further raise the security risk. As per the report of Kaspersky Laboratory [1], 93.01% of Web based attacks are caused by the malicious URLs.

Malware term comes from combining the two words malicious and software. It is a collective term refers to a variety of forms of hostile or intrusive software. Malware infects executable, Interpreted file, Kernel, Service, MBR, Hypervisor, etc. Malware also invades the user privacy through accessing and changing the documents and data.

"Any code added, changed, or removed from a software system in order to intentionally cause harm or subvert the intended function of the system". [2]

Classification is a machine learning technique. Machine learning utilizes the different techniques related to data mining, statistics, and computer science. It is related to the development of algorithms and methods that make computers capable of acquiring skill and integrating knowledge from data or experience, with the objective of solving a given problem in a way analogical to human learning [3].

The term cloud is a metaphor for the Internet and is a simplified representation of the complex, internetworked devices and connections that form the Internet. Cloud computing is emerging as a new paradigm for next generation computing in the field of computer science and information technology because of their attractive services such as adaptive, online, value added and pay as use scheme. Cloud can be defined in a number of ways. It is a business model, which provides the on demand hardware and software as services to the client through internet [1].

According to NIST cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources. Theses computing resources include networks, servers, storage, applications, and services. This cloud model is basically composed of five essential characteristics, three types of service models, and four deployment models.

A. Malware Intents and Attacks

Malware execute the code snippets for malicious intentions and execute the attacks. Malware intents and their related attacks are as follows-

B. Data Breaches

It refers to maintaining and ensuring the accuracy, consistency and validity of data throughout its complete lifecycle. Data integrity requires that data be protected against any types of human errors, data transmitting errors, viruses, disk crashes and natural disasters. Ensuring data integrity is the fundamental obligatory responsibility of a cloud vender.

C. Privacy Issues

The ability of cloud computing to adequately address privacy regulations is called into question. Organizations today face numerous different requirements attempting to protect the privacy of individuals' information, and it is not clear (i.e., not yet established) whether the cloud computing model provides adequate protection of such information, or whether organizations will be found in violation of regulations because of this new model.

D. Account Hijacking

Account hijacking is an approach to take over a session through acquiring the session ID of victim and masquerading as the authorized client [4]. An authentic session may be compromised by either stealing the token or guessing the token ID. Account Hijacking not only provides access to account as authentic client but also adversely affect integrity of the victim.

II. SECURITY THREATS IN CLOUD ENVIRONMENT

There are various security threats in cloud environment in which few are as follows

A. Virtual Machine Reset Vulnerabilities

Virtualization technologies enable significant flexibility in handling the state of guest systems (an operating system, user applications, and data).

In particular, virtual machine (VM) snapshots, i.e. copies of the state of the guest, can be used to replicate, backup, transfer (to another physical system), or reset (to a prior state) the guest. Snapshots are one reason virtualization is transforming numerous areas of computing.

R. Schwarzkopf, et Al. [3] suggested that, in theory, snapshots might lead to security problems due to reuse of security-critical state. Namely, reusing a VM snapshot might lead to Virtual Machine reset vulnerabilities. But no insecurities have been reported for real systems, leaving open the question of whether reset vulnerabilities are a practical problem.

B. Randomness Vulnerabilities

The Random Number Generator (RNG) is a mechanism in charge of producing pseudo random numbers. Random numbers are then used for generating the following values-

- TCP Sequence Numbers
- Secure Shell (SSH) keys
- Random PIDs for Processes, and
- Unique User Identification Number.

There are many methods for generating cryptographically strong random numbers. We do not go into significant detail regarding particular implementations [3, 4]. Randomness in operating systems is generated from non-predictable sources like user input and hardware interrupts.

There exist numerous ways in which an Random Number Generator (RNG) stack might fail or be tampered with by a dedicated attacker. For the purposes of threat modeling, however, we can loosely categorize randomness failures by the resultant quality of randomness as seen from the point of view of an application.

- **Fresh randomness:** An application is always provided new, private, uniform bits.
- **Reused randomness:** An application is provided private, uniform bits, but these bits might have been provided to the application before.
- **Exposed randomness:** An application is provided uniform bits but attackers later learn these bits.
- **Predictable randomness:** An application is provided random bits that are predictable by an adversary.
- **Chosen randomness:** An application is provided adversarial -chosen random bits.

Randomness vulnerabilities lead to applications using one of the four kinds of bad randomness.

C. Security risks due to the multi-tenancy-

In a multi-tenant environment, security dependencies on the logical separation at multiple layers are more critical rather than the physical separation of resources. Some of the cloud providers due to multi-tenancy may not allow audit and assessment by a particular tenant within their shared infrastructure. Few of the security risks due to the multi-tenancy are as follows-

D. Inadequate Logical Security Controls:

Physical resources (CPU, networking, storage/databases, and application stack) are shared between multiple tenants. That means dependence on logical segregation and other controls to ensure that one tenant deliberately or inadvertently cannot interfere with the security (confidentiality, integrity, availability) of the other tenants.

III. LOG FILES IN CYBER FORENSIC

Cyber forensics has been defined as the use of scientifically derived and proven methods towards the preservation, collection, validation, identification, analysis, interpretation and presentation of cyber evidence derived from cyber sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal or helping to anticipate the unauthorized actions shown to be disruptive to planned operations. One important Element of cyber forensics is the credibility of the cyber evidence.

In Cyber forensic, log files are like the black box on an airplane that records the events occurred within an organization’s system and networks. Logs are composed of log entries that play a very important role in evidence gathering and each entry contains information related to a specific event that has occurred within a system or a network. Log files helps cyber forensic process in probing and seizing computer, obtaining electronic evidence for criminal investigations and maintaining computer records for the federal rules of evidence.

IV. BACKGROUND AND LITERATURE

This section presents the background review of malware detection techniques and illustrates the pros and cons of these techniques. Researchers have proposed numerous approaches for detecting the ever increasing number of web sites spreading malware via drive-by downloads.

A. Static Malware Forensic Investigation

In the field of static analysis, researchers focus on deriving the libraries of particular pattern on the basis of snippets or fixed behavior.

K. Jeong and H. Lee [2] proposed an algorithm using system-call graph to develop patterns and detect the malware snippets. In order to recognize metamorphic malware snippets. Proposed system may detect and block the malware before its execution.

K. Kausal [5] presented a statistical analysis of the API call from binary executable files using similarity measurement function. It should contain sufficiently similar API calling sequence to preserve functionality of a metamorphic virus.

J. M. Borello et Al. [6] measure similarity between the behaviors of programs. Analyze the behavior of malware snippets through comparing the execution traces in terms of system calls.

Jusuk Lee et Al. [7] enhanced the system call graph technique to analyze malware snippets statically through classifying API calls in 128 clusters.

B. Dynamic Malware Forensic Investigation

This technique analyzes the malware through executing them. Detection technique first executes the malware into simulated environment, monitors the system calls and then observes its behavior. Usually virtual machines are employed as a sand box to execute the malware snippets and extract the behavioral features Dynamic analysis may detect unknown malwares. However dynamic analysis involves the risk of system infection.

In [8] malicious web pages which containing dynamic. HTML code, can be harmful for computers are detected using machine learning. Machine learning is used to classify a web page in malicious and non malicious depending on the feature extracted in this work. The aim of this work is to propose a technique which is resilient to code of the web pages.

In [9] activities on web servers and systems which are connected to internet are tracked using honey pot. These web activity logs are classified as malicious and non malicious. Supervised machine learning approaches are used to classify logs into vulnerable activity and attack. In this work 43 different features are extracted and studied. All the data collected by honey pot is having these 43 attributes.

In [10] viruses and cyber security threats and other malwares are detected using SVM. Normally virus detection mechanism use signature based approach. In this work supervised learning approach is used. Signature patterns are generated by machine learning and behavior detection methods. These patterns are compared in terms of accuracy.

In [11] Random forest, SVM Decision tree methods are used to classify malwares integrated with static and dynamic features. In this wirk static and dynamics feature extraction techniques are used to extract features. The above mentioned approaches are applied and compared in term of accuracy and time. It is concluded that Random forest machine learning approach yields the best results.

In [12] a survey is done on various machine learning algorithms and different phases in detection of malware. The three phases discussed in work is file representation technique, Feature selection method and classification. It is observed that each phase has significant effect on the accuracy.

V. ISSUES IN LOG ANALYSIS

Some early work on the subject of focused decentralize log event correlation of data from the Web was done, in the context of client-based information. This work proposed the decentralized log event correlation architecture for correlating the heterogeneous and distributed log events in real time.

A. Decentralized Log Event Correlation

They employed and adapted continuous queries for using inverted indexes to correlate log events. In addition, LEC may filter events before the correlation and may detect complex correlation that need to interrogate the history of events. Next, authors implement a log event generator to simulate and to configure a distributed log flows and they implement a wrapper that transform the generated logs to an XML schema's of their LEC architecture.

B. Centralized Log Event Correlation

Centralization of log files together with some kind of visualization is proven to be very helpful; as it will reduce the time spent searching for chains of events. For example if a user makes 3 failed login attempts, the user would probably be blocked from further login attempts, but we don't know if this is a legitimate user having problems remembering his password.

C. Rule Based Correlation on Event Logs

Over the past decade, event log correlation has received a lot of attention in the context of network fault management. A number of approaches have been proposed for event correlation, including rule-based [10], codebook based [11], Bayes network based [13], neural network based [14], and graph based methods. There are also a number of event correlation products available on the market, like HP ECS, SMARTS, Net Cool, Nerve Center, LOGEC, and Rule Core.

Burns et al. have developed the Event Browser system [7] that uses data mining and visualization techniques for finding event patterns and for creating event correlation rules. The tools and algorithms described in can also assist the analyst in discovering event correlation knowledge.

VI. PROPOSED WORK:

Proposed work contains six phases as Log File Repository, Preprocessing of Log File, Fine-grained Log Feature Selection, Feature Modeling, Classification, and Abnormal Behavior Analysis. The framework is as follows-

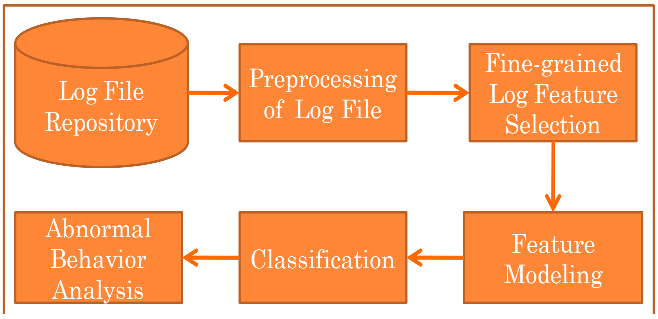
Log File Repository
Preprocessing of Log File
Fine-grained Log Feature Selection
Feature Modeling
Classification
Abnormal Behavior Analysis

Figure1: Proposed Framework for Malware Detection through Log File

A. Log File Repository

Cloud offers access logs that provides the information related to the request made by the end users. Entries in access log are recorded automatically as new request occurred in the cloud as shown in Figure 3.2. It may represent the behavior of resources used in the website. In this work, dataset consists of a total 12 Kippo Log Files in .log file format. Each log file contains approx 1 lac entries.
Database is available on-

- A labeled data set for flow based Intrusion Detection
- Simpleweb.org

B. Preprocessing of Log File

The raw log files do not arrive in a format conducive to fruitful detection. Preprocessing is performed to improve the quality of data, efficiency and effectiveness. Therefore, substantial log preprocessing must be applied. The preprocessing tasks applied in this work are

C. Data Cleaning and Filtering-

Log file contains number of raw and irrelevant entries. Remove the duplicity of the record in access logs. Access log have an ‘s_request_id field’ which is unique for each record. Preprocessing phase selects only one entry of the same ‘s_request_id’ field to detect and remove the duplicate entries.

D. Log Feature Selection

Cloud offers access logs that administrator may download and scrutinized. Access log of cloud provide the information related to the request made by the end users. Entries in access log are recorded automatically which are shown in Table 1.

Table 1: Sample Features of Log Files

Sr. No	Field	Type	Description
1.	time_micros	Integer	The time that the request was completed, in microseconds since the Unix epoch.
2.	c_ip	String	The IP address from which the request was made.
3.	cs_uri	String	The URI of the request. sc_status integer The HTTP status code the server sent in response.
4.	cs_bytes	Integer	The number of bytes sent in the request.
5.	sc_bytes	Integer	The number of bytes sent in the response.
6.	time_taken_ micros	Integer	The time it took to serve the request in microseconds.
7.	cs_bucket	String	The bucket specified in the request. If this is a list buckets request, this can be null.
8.	cs_object	String	The object specified in this request. This can be null.

Access log is to analyze the malware attacks. It may represent the behavior of resources used in the website. To simulate the behavior of web site, features of access log is extracted and categorized into two types as-

E. Feature Modeling

Selected features of the log file are parsed one by one and create the Feature Appearance Vector (FAV). The FAV in binary form only captures the types of feature appeared in the log file. The ith element value in the vector is set to be 1 if the ith message template is present in the log file. Otherwise, it is set to be 0.

VII. RESULT ANALYSIS AND DISCUSSION

Naïve Bayes, 5-fold cross-validation process is adopted due to the same standard used in different research work. Dataset is randomly divided into 5 smaller subsets. Where up to four subsets contain 15735 and fifth subset contains 15733 log entries. The four subsets are used for training and fifth subset is used for testing. The process is repeated 5 times for every combination. The required experimental setup and open source software explored in this research are illustrated in this section.

A. Evaluation through Performance Metric:

Performance metric measures how well our approach is performing on a particular dataset. Proposed approach and three other techniques are separately evaluated with labeled dataset. Labeled dataset is utilized to create the confusion matrix. False positive rate (FPR) is calculated and compared with three discrete techniques that are

- Random Forest
- Decision Tree
- Random Tree

B. Exhaustive Performance Evaluation

For cross validation initially 1st to 4th subset is used for training according to the proposed algebraic methods and 5th subset is used for testing. The 5th subset contains 15733 instances of a log file. The 5th subset has 1000 malware instances and 14733 benign instances. After validation class label returns 914 Malware and 14401 benign

From confusion matrix, Precision, Recall (True Positive Rate), Accuracy, False positive rate (FPR), and False Negative Rate are calculated for all five iteration as shown in Figure 2.

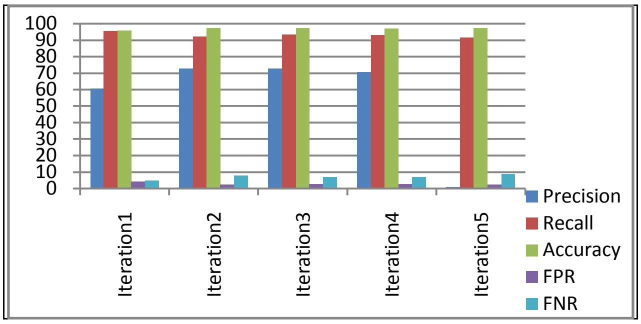

Figure 2 Performance Evaluation Matrices

The performance matrices of proposed method are compared with the three evaluated method as shown in Figure 3.

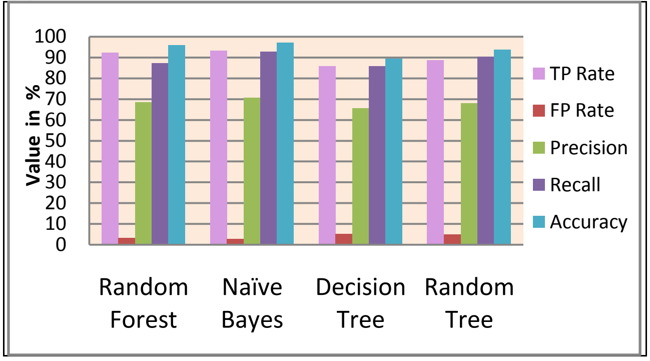

Figure 3: Comparison through Evaluated Performance Matrices.

VIII. CONCLUSION AND FUTURE WORK

This work presented static malware snippets detection using classification of log entries approach. It is based on entries of log file as features for detecting malware and malicious codes. After features selection, a sparse matrix between extracted features and each instances of log entry is created successfully. It is further applied to classify the abnormal and normal behavior.

To the result analysis, applied the 5-fold cross-validation process and datasets has randomly divided into 5 subsets of log file entries. From 5 subsets, first 4 subsets used to train and last 5th subset applied to test. Developed approach has better performances with same technique. The results analysis show that developed approach is the best detection accuracy is 96.96% with the false positive rate

2.78%. Developed approach is beneficial for cloud based systems for preventing the end users from information theft.

This work focuses on web based malware primarily on malicious ULRs. Future scope is to develop browser based plug-in on the basis of proposed approach to detect malicious URLs and other types of web based malware snippets.

IX. REFERENCES

[1]. M. Garnaeva, et Al., “Kaspersky Security Bulletin 2015”, December 2015, [online]. Available: https://securelist.com/analysis/kaspersky-security-bulletin/73038/kaspersky-security-bulletin-2015-overall-statistics-for-2015/, [Accessed: 02/11/2015].

[2]. K. Jeong and H. Lee, “Code graph for malware detection”, In: Proc. of the Int.Conf. on Information Networking, IEEE, pp. 1–5, 2008.

[3]. R. Schwarzkopf, et Al., “Increasing virtual machine security in cloud environments”, Springer Journal of Cloud Computing: Advances, Systems and Applications, Vol. 12, Issue 1, 2012, pp. 1-12.

[4]. J. Chang et Al., “Analyzing and defending against web-based malware”, ACM Computing Surveys (CSUR), Vol: 45, Issue: 4, pp. 1-35, August 2013.

[5]. K. Kausal, P. Swadas and N. Prajapati, “Metamorphic malware detection using statistical analysis”, Int. J. of Soft Computing and Engineering (IJSCE), Vol: 2, Issue:3, July 2012.

[6]. J. M. Borello, L. Me and E. Filiol, “Dynamic malware detection by similarity measures between behavioral profiles” Proc. of the 2011 Conf. on Network and Information Systems Security, IEEE, 2011, pp1-8.

[7]. J. Lee, K. Jeong and H. Lee, “Detection Metamorphic Malware using code Graph”, Proc. of the ACM Symposium on Applied Computing, 2010.

[8]. A. Kapravelos, et Al. “Revolver: An Automated Approach to the Detection of Evasive Web-based Malware” SEC'13 Proc. of the 22nd USENIX conf. on Security 2013, pp. 637-652.

[9]. V. Jain et Al., “Session Hijacking: Threat Analysis and Countermeasures”, Int. Conf. on Futuristic Trends in Computational Analysis and Knowledge Management, 2015.

[10]. M. Xu et Al., “A similarity metric method of obfuscated malware using function-call graph”, J. of Computer Virol Hack Tech, Springer, France, Vol: 9, Issue: 9, 2013, pp. 35–47.

[11]. Katerina Goseva-Popstojanova, et Al. “Characterization and classification of malicious Web traffic” in Computer and Network Security, Vol: 42, pp. 92-115, 2014.

[12]. D. R. Sahu and D. S. Tomar “DNS Pharming through PHP Injection: Attack Scenario and Investigation”, Int J of Computer Network and Information Security.

[13]. PingWang and Yu-ShihWang, “Malware behavioral detection and vaccine development by using a support vector model classifier” in Journal of Computer and System Sciences, Vol.: 81, pp. 1012–1026, 2015.

[14]. Thai-Hoang, Tran. on the methods of feature selection in text categorization. Thesis, Lunghwa University of science and technology, 2012.