International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 17
July 2018

A Novel Big Data based Security Analytics approach to detecting advanced attacks in Cloud Computing

Boddu Thirumala Rao & D Anandam
[1] *PG Scholar, Department of CSE, PACE Institute of Technology and Sciences,Vallur, Prakasam,, Andhrapradesh, India*
[2] *Assistant Professor, Department of CSE, PACE Institute of Technology and Sciences,Vallur, Prakasam, Andhrapradesh, India*

Abstract—*Cloud computing has Virtualized infrastructure become a target for cyber attackers to launch advanced attacks. This Project proposes a big data based security analytics approach to detecting advanced attacks in virtualized infrastructures. Network logs and user application logs collected periodically from the guest virtual machines (VMs) are stored in the Hadoop Distributed File System (HDFS). Then, extraction of attack features is performed through graph-based event correlation and Map Reduce parser based identification of potential attack paths. Next, determination of attack presence is performed through two-step machine learning, namley logistic regression is applied to calculate attack's conditional probabilities with respect to the attributes, and belief propagation is applied to calculate the belief in existence of an attack based on them.*

Index Terms—*Virtualized infrastructure, virtualization security, cloud security, malware detection, security analytics,event correlation, logistic regression, belief propagation*

1. Introduction

A virtualized infrastructure consists of multiple virtual machines (VMs) that are depend upon the software-defined multi-instance resources of the hosting hardware. The virtual machine monitor, also called hypervisor, sustains, regulates and manages the software-defined multi-instance architecture. The ability to pool different computing resources as well as enable on-demand resource scaling has led to the widespread deployment of virtualized infrastructures as an important provisioning to cloud computing services.

Existing security approaches to protecting virtualized infrastructures generally include two types, namely malware detection and security analytics. Malware detection usually involves two steps, first, monitoring hooks are placed at different points within the virtualized infrastructure, then a regularly-updated attack signature database is used to determine attack presence. While this allows for a real-time detection of attacks, the use of a dedicated signature database makes it vulnerable to zero-day attacks for which it has no attack signatures.

Security analytics applies analytics on the various logs which are obtained at different points within the network to determine attack presence. By leveraging the huge amounts of logs generated by various security systems (e.g., intrusion detection systems (IDS), security information and event management (SIEM), etc.), applying big data analytics will be able to detect attacks which are not discovered through signature- or rule-based detection methods. While security analytics removes the need for signature database by using event correlation to detect previously undiscovered attacks, this is often not carried out in real-time and current implementations are intrinsically nonscalable.

To overcome these limitations, in this paper we propose a novel big data based security analytics (BDSA) approach to protecting virtualized infrastructures against advanced attacks. By making use of the network logs as well as the user application logs collected from the guest VMs which are stored in a Hadoop Distributed File System (HDFS), our BDSA approach first extracts attack features through graph-based event correlation, a MapReduce parser based identification of potential attack paths and then ascertains attack presence through two-step machine learning, namley logistic regression and belief propagation.

The remainder of the paper is arranged as follows. Section 2 presents a review upon the existing security approaches. Section 3 proposes our big data based security analytics (BDSA) approach. Experimental evaluations are presented in Section 4, while Section 5 discusses our BDSA approach in contrast with the related work. Section 6 draws the conclusion.

2.Literature Review

2.1 Malware detection in virtualised infrastructure

Malware refers to any executable which is designed to compromise the integrity of the system on which it is run. There are two prominent approaches to malware detection in cloud computing, namely in-VM and outside-VM inter-working approach and hypervisor-assisted malware detec-tion.

2.1.1In-VM and outside-VM interworking approach to malware detection

In-VM and outside-VM interworking detection consists of an in-VM agent running within the guest VM, and a remote scrutiny server monitoring the VM's behaviour. When a potential malware execution is detected the in-VM agent sends the suspicious executable to the scrutiny server, which then uses the signature database to verify malware presence or otherwise and then informs the in-VM agent of the results.

CloudAV, a cloud-based malware detection system fea-turing multiple antivirus engines, employs in-VM and outside-VM interworking approach to protect the guest VMs against attacks [4]. Apparently the effectiveness of this scheme depends on the frequency at which the virus signatures are updated by the antivirus vendors.

The in-VM and outside-VM interworking approach is also used by CuckooDroid, to detect mobile malware pres-ence on Android devices [5]. It consists of an in-device agent which scans executables on the device and sends any sus-picious executable to a remote scrunity server which runs a hybrid of anomaly-based and signature-based malware detectors. The scheme first extracts malware features by using static as well as dynamic analysis on malware apps. The obtained features are then used to train a one-class SVM (Support Vector Machine) classifier for anomaly-based detection. Implemented on an emulated Android platform, CuckooDroid achieved a detection accuracy of 98.84 %.

2.1.2 Hypervisor-assisted malware detection

Hypervisor-assisted malware detection, on the other hand, uses the underlying hypervisor to detect malware within the guest VMs.

A hypervisor-assisted malware detection scheme is de-signed in [6] to detect botnet activity within the guest VMs. The scheme installs a network sniffer on the hypervisor to monitor external traffic as well as inter-VM traffic. Implemented on Xen, it is able to detect the presence of the Zeus botnet on the guest VMs.

A hypervisor-assisted detection scheme is proposed in [7] using guest application and network flow characteris-tics. This scheme first uses LibVMI to extract key process features from the processes running within VMs and then uses tcpdump together with the CoralReef network packet analysis tool from CAIDA (Center for Applied Internet Data

Analysis) to extract network flow features. The obtained features are then used to train one-class SVM classifiers to detect malware presence within guest VMs. Implemented on KVM, the scheme is able to detect well-known DDoS (Distributed Denial of Service) and botnet attacks such as LOIC (Low Orbit Ion Cannon) and Zeus.

The hypervisor-assisted detection is also used in Access-Miner [8]. Implemented as a custom hypervisor, Access-Miner monitors normal user behavior within the system and creates access activity models which are used for anomaly-based malware detection. To ensure that the underlying hardware is protected, it intercepts the guest system call requests and uses a policy checker module to determine if it should access the system resource.

2.2 Security Analytics

Security Analytics refers to the application of analytics in the context of cybersecurity [9]. Based on a variety of data collected from different points within an enterprise network, security analytics aims to detect previously undiscovered threats by use of analytic techniques.

Common techniques of security analytics include clus-tering and graph-based event correlation.

2.2.1 Clustering for security analytics

Clustering organises data items in an unlabeled dataset into groups based on their feature similarities [10]. For security analytics, clustering finds a pattern which generalises the characteristics of data items, ensuring that it is well generalized to detect unknown attacks. Examples of cluster-based classifiers include K-means clustering and k-nearest neighbors, which are used in both intrusion detection and malware detection.

Clustering is used for security analytics for industrial control systems [11] in an NCI (networked critical infrastructure) environment. First, data outputs from various network sensors are arranged as vectors and K-means clustering is applied to group the vectors into clusters. The MapReduce model is then applied to the grouped clusters to find groupings of possible attack behaviour, thus allowing the detection to be carried out efficiently.

In [12] an "attack pyramid" -based scheme is proposed to detect APTs (advanced persistent threats) in a large enterprise network environment. Based on threat tree modeling, different planes (namely hardware, user, network, application) to which an attack may be launched are placed hierarchically with the end goal placed at the top. First, outputs from all available sensors in the network (e.g., network logs, execution traces, etc) are put into contexts. Then, in terms of the contexts various suspicious activities detected at each attack plane are correlated in a MapReduce model,

which takes in all the sensor outputs and generates an event set describing potential APTs. Finally, an alert system determines attack presence by calculating the confidence levels of each correlated event.

SINBAPT (Security Intelligence techNology for Blocking APT) [13] uses big data processing such as HDFS and MapReduce together to detect the presence of APTs in an enterprise network environment. Used for anomaly-based detection, the scheme collects log data from different sources
(e.g., Netflow, application logs, etc) and applies a MapReduce model for feature extraction. Once organized into clusters, the data is then used to determine attack presence according to pre-defined rules.

2.3 Limitations of existing approaches

Existing approaches to detecting attack presence are limited in terms of their ability to detect threats in real-time as well as to scale across multiple hosts.

One of the limitations of existing security approaches stems from the use of a dedicated signature database for threat detection. This applies to approaches that feature a regularly updated attack signature database for threat detection. Typically in the in-VM and outside-VM interworking approach, an in-VM agent detects and passes any suspicious file to the remote scrunity server, which uses the signature database to determine if it is a malware. The dependence on a regularly-updated signature database makes it limited in detecting zero-day attacks. While BareCloud [17] and
CloudAV [4] attempt to get around this limitation by using multiple antivirus engines for threat detection, they are still limited in detecting previously undiscovered attacks due to the post factum data in updating the signature database. This is further exacerbated by an increased number of false positives reported by the different antivirus engines.

Security analytics removes the need for signature database by correlating events from the collected logs, but they still suffer from the post factum data in training for threat detection. Typically BotCloud [14] and Nazca [18] collect data over long periods of time (usually over a 24 hour period) and apply analytics for threat detection. While a long period of time allows for a rich collection of data, that entails a tendency in detecting threats which have already taken place over a breadth of time within the network. This makes it difficult, if not impossible, to focus on immediate events and take immediate actions against a compromised point within the network.

Another limitation of existing security approaches is the centralized execution process. For instance, SINBAPT [13] runs on a single host, collecting data from various points within the network and analysing them as a single centralized process. While centralized execution process is feasible in network environments in which there is a single centralized server responsible for monitoring all network components, it is infeasible for large network environments in which multiple guest VMs are hosted on different hosts and attack presence has to be communicated to other hosts in near real-time.

3. Proposed Approach

3.1 Overall Framework

The basic idea of our proposed approach is to detect in real-time any malware and rookit attacks via holistic efficient use of all possible information obtained from the virtualized infrastructure, e.g., various network and user application logs. Our proposed approach is a big data problem for the
following characteristics of the network and user application logs collected from a virtualized infrastructure:

• ***Volume:*** Depending on the number of guest VMs and the size of the network, the amount of the network and user application logs to be collected can range from approximately 500 MB to 1 GB an hour;

• ***Velocity:*** The network and user application logs are collected in real-time, in order to detect the presence of malware and rootkit attacks, accordingly the collected data containing its behavior needs to be processed as soon as possible;

• ***Veracity:*** Due to the “low and slow” approach that malware and rootkit take in hiding their presence within the guest VMs, data analysis has to rely upon event correlation and advanced analytics.

The design principles, which are integral in the development of our BDSA approach to protecting virtualized infrastructures, can be elaborated as follows.

Design Principle # 1 - Unsupervised classification: The attack detection system should be able to classify potential attack presence based on the data collected from the virtualized infrastructure over time.

Design Principle # 2 - Holistic prediction: The attack detection system should be able to identify potential attacks by correlating events on the data collected from multiple sources in the virtualized infrastruc-ture.

Design Principle # 3 - Real-time: The attack detection system should be able to ascertain attack presence as immediately as possible so as for the appropriate countermeasures to be taken immediately.

Design Principle # 4 - Efficiency: The attack detection system should be able to detect attack presence at a

high computational efficiency, i.e., with as little performance overhead as possible.

Design Principle # 5 - Deployability: The attack detection system should be readily deployable in production environment with minimal change required to common production environments.

Figure 1 illustrates the overall conceptual framework of our proposed big data based security analytics (BDSA) approach, with the different components highlighted in blue.

Our BDSA approach consists of two main phases, namely

• Extraction of attack features through graph-based event correlation and MapReduce parser based identification of potential attack paths, and

• Determination of attack presence via two-step machine learning, namely logistic regression and belief propagation.

Prior to the online detection of attacks, there is actually a system initialization, in which offline training of the logistic regression classifiers is carried out, that is, the stored features are loaded from the *Cassandra* database to train the logistic regression classifiers. Specifically, wellknown malicious as well as benign port numbers are loaded to train a logistic regression classifier to determine if the incoming/outgoing connections are indicative of an attack presence. Likewise, well-known malware and legitimate applications together with their associated ports are loaded to train a logistic regression classifier to determine if the behavior of an application running within the guest VM is indicative of an attack presence. These trained logistic regression classifiers are ready for online use, upon the extraction of new attack features, to determine if the potential attack paths are indicative of attack presence.

In the Extraction of Attack Features phase, first, it carriesout Graph-Based Event Correlation. Periodically collectedfrom the guest VMs, network and user application logs arestored in the *HDFS*. By assembling the information contained in these two logs, theCorrelation Graph Assembler (*CGA*) forms correlation graphs. Then, it carries out the Identification of Potential Attack Paths. A MapReduce model is used to parse the correlation graphs and identify the potential attack paths i.e., the most frequently occurring graph paths in terms of the guest VMs' IP addresses. This is based on the observation that a compromised guest VM tends to generate more traffic flows as it tries to establish communication with an attacker.

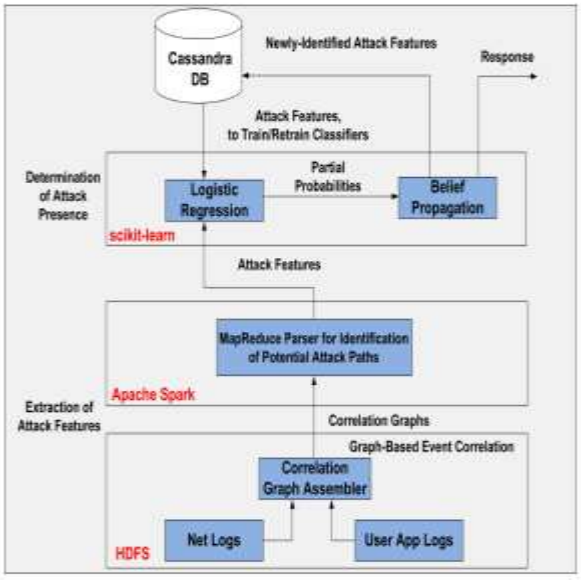

Fig. 1: Conceptual framework of the proposed big data based security analytics (BDSA) approach

In the Determination of Attack Presence phase, two-step machine learning is employed, namely logistic regression and belief propagation are used. While the former is used to calculate attack's conditional probabilities with respect to individual attributes, the latter is used to calculate the belief of an attack presence given these conditional attributes. From the potential attack paths, the monitored features are sorted out and passed into their logistic regression classifiers to calculate attack's conditional probabilities with respect to individual attributes. The conditional probabilities with respect to individual attributes are passed into belief propagation to calculate the belief of attack presence. Once attack presence is ascertained, the administrator is alarmed of the attack. Furthermore, the *Cassandra* database is updated with the newly identified attack features versus the class ascertained (i.e., attack or benign), which are then used to retrain the logistic regression classifiers.

4. Overall algorithm of BDSA

Our BDSA approach can be formulated in pseudo codes as shown in Algorithm 1.

The execution of the proposed BDSA approach begins by loading all well-known malicious as well as benign port numbers from the distributed Cassandra database. Both of these port types are then used to train a classifier using using logistic regression. This allows the proposed approach to determine on-the-fly the probability of an unknown port being malicious, before passing it to the belief propagation framework for final

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 17
July 2018

aggregation. A trained logistic classifier is used to determine if any of the attributes are malicious or benign, before passing their respective probabilities to the belief propagation process for final probability aggregation. Belief propagation process takes attack's conditional probabilities with respect to individual attributes to calculate the belief of attack presence, taking into account each conditional probability values to ensure that the value obtained is not influenced only by any conditional probability alone.
The Algorithm as follows

Algorithm 1 Security analytics in BDSA

1: **Initialize:** Obtain benign and malicious parameters of the attack features from *Cassandra* DB.
2: Train classifiers for monitored features using Logistic Regression.
3: **while** True **do**
4: Collect network and user application logs from guest VMs.
5: Filter network log entries using the guest VMs' IP addresses.
6: Form correlated_log.
7: Use correlated_log to form a correlation graph G.
8: Input G into MapReduce parser to identify potential attack paths {*attack_paths*}, which is a sub-set of all graph paths as shown in Figure 3.
9: **for** each *attack_path* in {*attack_paths*} **do**
10: $i \leftarrow 0$.
11: **for** each monitored feature $t_{feature}$ in *attack_path* **do**
12: Calculate P^{port_change}, $P^{unknown\_exect}$, $P^{in\_connect}$, and $P^{out\_connect}$
13: Pass P^{port_change}, $P^{unknown\_exect}$, $P^{in\_connect}$, and $P^{out\_connect}$ into Step. 4 of Algorithm 1.
14: End do
15: End do
16: End

4.1 Detection of userspace malware and kernel-level rootkits

The ability of the BDSA approach to detect different malware attacks is evaluated by executing the two userspace malware programs as well as the two kernel-level rootkits on the guest VMs. The malware and rootkits are taken from PacketStorm [25] as shown in Table 1. They were selected due to the availability of their source code, which enables the severity of their attacks to be modified and tested against our BDSA approach.

Detection of userspace malware

Also known as *application-level* malware, userspace malware runs at the application-level of the guest operating system alongside other legitimate applications. The ability of our BDSA approach to detect userspace malware is evaluated by executing the aforementioned userspace malware on the guest VMs. In order to run the userspace malware, a test scenario is set up, that is, one guest VM acts as an attacker while another guest acts as an attack victim. The attacker VM is then made to listen to different non-standard port numbers using netcat, and then runs the reverse shellcode on the victim VM. The same test scenario is used for creating a C & C (Command & Control) botnet, by running the server component of the botnet on the attacker VM and its client component on the guest VM. In both of the test scenarios, the user space malware is executed *as is* with only the hard-coded destination IP addresses and the port numbers modified. Both user space malware programs are executed 5 times with up to three guest VMs. In all cases, our BDSA approach is able to detect them through monitoring the communication flows between them as well as the ports which are opened on the guest VMs.

TABLE1: Malware and rootkits tested

Malware/Rootkit	Category	Execution space on guest VM	Characteristics
Reverse shellcode	Malware	User space	Establishing an external reverse shell connection
C & C botnet	Malware	User space	Creating a master-slave botnet connection
XingYiQuan	Rootkit	Kernel space	Executing in the guest VM's kernel and establishing an external connection
Azazel	Rootkit	Kernel space	Executing in the guest VM's kernel and establishing an external connection

5.Conclusion

In this paper, we have put forward a novel big data based security analytics (BDSA) approach to protecting virtualized infrastructures in cloud computing against advanced attacks. Our BDSA approach constitutes a three phase framework for detecting advanced attacks in real-time. First, the guest VMs's network logs as well as user application logs are periodically collected

from the guest VMs and stored in the HDFS. Then, attack features are extracted through correlation graph and MapReduce parser. Finally, two-step machine learning is utilized to ascertain attack presence. Logistic regression is applied to calculate attack's conditional probabilities with respect to individual attributes. Furthermore, belief propagation is applied to calculate the overall belief of an attack presence. From the second phase to the third, the extraction of attack features is further strengthened towards the determination of attack presence by the two-step machine learning. The use of logistic regression enables the fast calculation of attack's conditional probabilities. More importantly, logistic regression also enables the retraining of the individual logistic regression classifiers using the new attack features as they are obtained from attack detection.

The use of belief propagation calculates the aggregate belief of an attack presence by taking into account the conditional probabilities with respect to individual attributes, which thereby achieves a holistic view of the guest VM's behavior. The effectiveness of our BDSA approach is evaluated by testing it against well-known malware and rookit attacks.

In all cases, it has been shown that our BDSA approach is able to detect them while maintaining a consistent performance overhead with increasing number of guest VMs at an average detection time of approximately 0.06 ms. Tested against *Livewire*, our BDSA approach incurs less performance overhead in attack detection through monitoring the guest VM's behavior.

Our BDSA approach has taken advantage of the distributed processing of HDFS and real-time ability of MapReduce model in Spark to address the velocity and volume challenges in security analytics. To tackle the veracity issue posed in zero-day attacks, our BDSA approach addresses this challenge by enforcing the on-the-fly mechanism for the retraining of logistic regression classifiers.

REFERENCES

[1] D. Fisher, "'venom' flaw in virtualization software could lead to vm escapes, data theft," https://threatpost.com/venomflaw-in-virtualization-software-could-lead-to-vm-escapes-datatheft/112772/, 2015, accessed: 2015-05-20.

[2] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer *et al.*, "The matter of heartbleed," in *Proceedings of the 2014 Conference on Internet Measurement Conference*. Vancouver, BC, Canada: ACM, 2014, pp. 475–488.

[3] K. Cabaj, K. Grochowski, and P. Gawkowski, "Practical problems of internet threats analyses," in *Theory and Engineering of Complex Systems and Dependability*. Springer, 2015, pp. 87–96.

[4] J. Oberheide, E. Cooke, and F. Jahanian, "Cloudav: N-version antivirus in the network cloud." in *USENIX Security Symposium*, San Jose, California, USA, 2008, pp. 91–106.

[5] X. Wang, Y. Yang, and Y. Zeng, "Accurate mobile malware detection and classification in the cloud," *SpringerPlus*, vol. 4, no. 1, pp.1–23, 2015.

[6] P. K. Chouhan, M. Hagan, G. McWilliams, and S. Sezer, "Network based malware detection within virtualised environments," in *Euro-Par 2014: Parallel Processing Workshops*. Porto, Portugal: Springer, 2014, pp.335–346.

[7] M. Watson, A. Marnerides, A. Mauthe, D. Hutchison *et al.*, "Malware detection in cloud computing infrastructures," *IEEE Transactions on Dependable and Secure Computing*, pp. 192 –205, 2015.

[8] A. Fattori, A. Lanzi, D. Balzarotti, and E. Kirda, "Hypervisorbased malware protection with accessminer," *Computers & Security*, vol. 52, pp. 33–50, 2015.

[9] T. Mahmood and U. Afzal, "Security analytics: big data analytics for cybersecurity: a review of trends, techniques and tools," in *Information assurance (ncia), 2013 2nd national conference on*. Rawalpindi, Pakistan: IEEE, 2013, pp. 129–134.

[10] C.-T. Lu, A. P. Boedihardjo, and P. Manalwar, "Exploiting efficient data mining techniques to enhance intrusion detection systems," in *Information Reuse and Integration, Conf, 2005. IRI-2005 IEEE International Conference on.* Las Vegas, Nevada, USA: IEEE, 2005, pp. 512–517.

[11] I. Kiss, B. Genge, P. Haller, and G. Sebestyen, "Data clusteringbased anomaly detection in industrial control systems," in *Intelligent Computer Communication and Processing (ICCP), 2014 IEEE International Conference on*. Cluj-Napoca, Romania: IEEE, 2014, pp. 275–281.

[12] P. Giura and W. Wang, "Using large scale distributed computing to unveil advanced persistent threats," *Science J*, vol. 1, no. 3, pp. 93–105, 2012.

[13] H. Kim, I. Kim, and T.-M. Chung, "Abnormal behavior detection technique based on big data," in *Frontier and Innovation in Future Computing and Communications*. Springer, 2014, pp. 553–563.

[14] J. Francois, S. Wang, W. Bronzi, R. State, and T. Engel, "Botcloud:detecting botnets using mapreduce," in *Information Forensics and Security (WIFS), 2011 IEEE International Workshop on*. Foz do Iguacu, Brazil: IEEE, 2011, pp. 1–6.

[15] L. Aniello, A. Bondavalli, A. Ceccarelli, C. Ciccotelli, M. Cinque,F. Frattini, A. Guzzo, A. Pecchia,

A. Pugliese, L. Querzoni *et al.*, "Big data in critical infrastructures security monitoring: Challenges and opportunities," *arXiv preprint arXiv:1405.0325*, 2014.

[16] L. Chen, T. Li, M. Abdulhayoglu, and Y. Ye, "Intelligent malwaredetection based on file relation graphs," in *Semantic Computing(ICSC), 2015 IEEE International Conference on*. Anaheim, California,USA: IEEE, 2015, pp. 85–92.

[17] D. Kirat, G. Vigna, and C. Kruegel, "Barecloud: bare-metal analysis-based evasive malware detection," in *23rd USENIX Security Symposium (USENIX Security 14)*, San Diego, California, USA, 2014, pp. 287–301.

[18] L. Invernizzi, S. Miskovic, R. Torres, S. Saha, S. Lee, M. Mellia, C. Kruegel, and G. Vigna, "Nazca: Detecting malware distribution in large-scale networks," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, California, USA, 2014, pp. 23–26.

[19] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[20] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.

[21] SANS, "Intrusion detection faq: What port numbers do wellknown trojan horses use?" https://www.sans.org/securityresources/idfaq/oddports.php, 2001, accessed: 2015-09-30.

[22] IANA, "Service name and transport protocol port numberregistry,"http://www.iana.org/assignments/service-namesport-numbers/service-names-port-numbers.xhtml, 2015, accessed: 2015-09-30.

[23] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[24] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 498–519, 2001.

[25] PacketStorm, "Packetstorm security," http://tinyurl.com/qhygrsu, 2013, accessed: 29-10-2014.

[26] T. Garfinkel, M. Rosenblum *et al.*, "A virtual machine introspection based architecture for intrusion detection," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, vol. 3, San Diego, California, USA, 2003, pp. 191–206.

[27] T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda, "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," in *Proceedings of the 29th Annual Computer Security Applications Conference*. New Orleans, Lousiana, USA: ACM, 2013, pp. 199–208.

[28] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, "Big data analytics framework for peer-to-peer botnet detection using random forests," *Information Sciences*, vol. 278, pp. 488–497, 2014.

Author's Profile

Mr. Boddu Thirumala Rao pursuing M. Tech in Computer Science and Engineering from PACE Institute Of Technology and Sciences affiliated to the Jawaharlal Nehru technological University,Kakinada.

Mr.D.Anandam has received his B.Tech and M.Tech PG. He is Dedicated to Teaching Field from the last 10 Years. He has Guided 10 P.G Students and 20 U.G Students. At Present He is Working as Asst.Professor in PACE Institute Of Technology and Sciences, Vallur, Prakasam(Dt), AP, India.He is Highly Passionate and Enthusiastic about his Teaching and Believes that Inspiring Students to Give of his best in order to Discover what he Already knows is better than Simply Teaching.