RESEARCH ARTICLE OPEN ACCESS

Privacy and Security of Big Data Mining Issues

Pasupuleti Nagendra Babu[1], Prof.S.Rama Kirshna[2]
[1]Research Scholar,M.Tech.,Ph.D ,Rayalaseema University ,Kurnool - 518007 Andhra Pradesh
[2]Department of Computer Science, Sri Venkateswara University ,Tirupati – 517502. INDIA

----------------------------------*********************------------------------------

Abstract:

Today the main crucial task is one of the most important concept is to store and preserve the data in a safest place and retrieving the data in a efficient and intelligent method even then today we are seeing the information technology is drastic growth at the same time there is not having security for data. Making some changes in security point of issue this research revises the most important aspects in how computing infrastructures should be configured and intelligently managed to fulfill the most notably security aspects required by Big Data applications. One of them is privacy. It is a pertinent aspect to be addressed because users share more and more personal data and content through their devices and computers to social networks and public clouds. So, a secure framework to social networks is a very hot topic research. This last topic is addressed in one of the two sections of the current topic with case studies. In addition, the traditional mechanisms to support security such as firewalls and demilitarized zones are not suitable to be applied in computing systems to support Big Data. SDN is an emergent management solution that could become a convenient mechanism to implement security in Big Data systems, as we show through a second case study at the end of the topic. This also discusses current relevant work and identifies open issues.

Keywords — **Big Data, Security, Privacy, Data Ownership, Cloud, Social Applications, Intrusion Detection, Intrusion Prevention.**

----------------------------------*********************--------------------------------

INTRODUCTION

The Big Data is an emerging area applied to manage datasets whose size is beyond the ability of commonly used software tools to capture, manage, and timely analyze that amount of data. The quantity of data to be analyzed is expected to double every two years (IDC, 2012). All these data are very often unstructured and from various sources such as social media, sensors, scientific applications, surveillance, video and image archives, Internet search indexing, medical records, business transactions and system logs. Big data is gaining more and more attention since the number of devices connected to the so-called “Internet of Things” (IoT) is still increasing to unforeseen levels, producing large amounts of data which needs to be transformed into valuable information. Additionally, it is very popular to buy on-demand additional computing power and storage from public cloud providers to perform intensive data-parallel processing. In this way, security and privacy issues can be potentially boosted by the volume, variety, and wide area deployment of the system infrastructure to support Big Data applications.

As Big Data expands with the help of public clouds, traditional security solutions tailored to private computing infrastructures, confined to a well-defined security perimeter, such as firewalls and demilitarized zones (DMZs) are no more effective. Using Big Data, security functions are required to work over the heterogeneous composition of diverse hardware, operating systems, and network domains. In this puzzle-type computing environment, the abstraction capability of Software-Defined Networking (SDN) seems a very important characteristic that can enable the efficient deployment of Big Data secure services on-top of the heterogeneous infrastructure. SDN introduces abstraction because it separates the control (higher) plane from the underlying system infrastructure being supervised and controlled. Separating a network's control logic from the underlying physical routers and switches that forward traffic allows system administrators to write high-level control programs that specify the behavior of an entire network, in contrast to conventional networks, whereby administrators (if allowed to do it by the device manufacturers) must codify functionality in terms of low-level device configuration. Using SDN, the intelligent management of secure functions can be implemented in a logically centralized controller, simplifying the following aspects: implementation of security rules; system (re)configuration; and system evolution. The robustness drawback of a centralized SDN solution can be mitigated using a hierarchy of controllers and/or through the usage of redundant controllers at least for the most important system functions to be controlled.

CONCEPTS ON BIG DATA CHALLENGES TO INFORMATION SECURITY AND PRIVACY ISSUES

With the proliferation of devices connected to the Internet and connected to each other, the volume of data collected, stored, and processed is increasing everyday, which also brings new challenges in terms of the information security. In fact, the currently used security mechanisms such as firewalls and DMZs cannot be used in the Big Data infrastructure because the security mechanisms should be stretched out of the perimeter of the organization's network to fulfill the user/data mobility requirements and the policies of BYOD (Bring Your Own Device). Considering these new scenarios, the pertinent question is what security and privacy policies and technologies are more adequate to fulfill the current top Big Data privacy and security demands (Cloud Security Alliance, 2013). These challenges may be organized into four Big Data aspects such as infrastructure security (e.g. secure distributed computations using MapReduce), data privacy (e.g. data mining that preserves privacy/granular access), data management (e.g. secure data provenance and storage) and, integrity and reactive security (e.g. real time monitoring of anomalies and attacks).

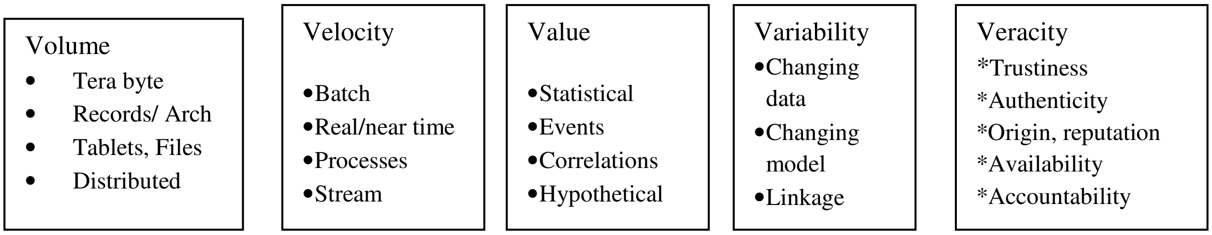


Figure 1. The five V's of Big Data (adapted from ("IBM big data platform - Bringing big data to the Enterprise," 2014))

Cloud Secure Alliance (CSA), a non-profit organization with a mission to promote the use of best practices for providing security assurance within Cloud Computing, has created a Big Data Working Group that has focused on the major challenges to implement secure Big Data services (Cloud Security Alliance, 2013). CSA has categorized the different security and privacy challenges into four different aspects of the Big Data ecosystem. These aspects are Infrastructure Security, Data Privacy, Data Management and, Integrity and Reactive Security. Each of these aspects faces the following security challenges, according to CSA:

* **Infrastructure Security**

1. Secure Distributed Processing of Data
2. Security Best Actions for Non-Relational Data-Bases

* **Data Privacy**

3. Data Analysis through Data Mining Preserving Data Privacy
4. Cryptographic Solutions for Data Security
5. Granular Access Control

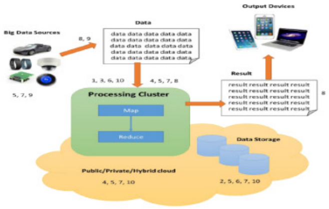
* **Data Management and Integrity**

6. Secure Data Storage and Transaction Logs
7. Granular Audits
8. Data Provenance

***Reactive Security**

9. End-to-End Filtering & Validation
10. Supervising the Security Level in Real-Time

These security and privacy challenges cover the entire spectrum of the Big Data lifecycle (Figure 2): sources of data production (devices), the data itself, data processing, data storage, data transport and data usage on different devices.



HP recently conducted a study on market-available IoT solutions and concluded that 70% of those contain security problems. These security problems were related with privacy issues, insufficient authorization, lack of transport encryption, insecure web interface and inadequate software protection (HP, 2014). Based on some of these findings, HP has started a project at OWASP (Open Web Application Security Project) that is entitled "OWASP Internet of Things Top Ten" (OWASP, 2014) whose objective is to help IoT suppliers to identify the top ten security IoT device problems and how to avoid them. This project, similar to the OWASP Top 10, identified the following security problems:

Face book recently conducted a study for unexpected or unnamed data attacks in social media and that company also notified some security concerns.

- Insufficient Authentication/Authorization: can allow an attacker to exploit a bad password policy, break weak passwords and access to privileged modes on the IoT device.
- Insecure Network Services: which can lead to an attacker exploiting unnecessary or weak services running on the device, or use those services as a jumping point to attack other devices on the IoT network.
- Lack of Transport Encryption: allowing an attacker to eavesdrop data in transit between IoT devices and support systems.
- Privacy Concerns: raised from the fact the most IoT devices and support systems collect personal data from users and fail to protect that data.
- Insecure Cloud Interface: without proper security controls an attacker can use multiple attack vectors (insufficient authentication, lack of transport encryption, account enumeration) to access data or controls via the cloud website.
- Insecure Mobile Interface: without proper security controls an attacker can use multiple attack vectors (insufficient authentication, lack of transport encryption, account enumeration) to access data or controls via the mobile interface.
- Insufficient Security Configurability: due to the lack or poor configuration mechanisms an attacker can access data or controls on the device.
- Insecure Software/Firmware: attackers can take advantage of unencrypted and unauthenticated connections to hijack IoT devices updates, and perform malicious update that can compromise the device, a network of devices and the data they hold.
- Poor Physical Security: if the IoT device is physically accessible than an attacker can use USB ports, SD cards or other storage means to access the device OS and potentially any data stored on the device.

It is clear that Big Data present interesting opportunities for users and businesses, however these opportunities are countered by enormous challenges in terms of privacy and security (Cloud Security Alliance, 2013). Traditional security mechanisms are insufficient to provide a capable answer to those challenges. In the next section, some of these solutions/proposals are going to be addressed.

SOLUTIONS / PROPOSALS TO ADDRESS BIG DATA MINE SECURITY AND PRIVACY CHALLENGES

As previously referred, traditional encryption and anonymization of data are not adequate to solve Big Data problems. They are adequate to protect static information, but are not adequate when data computation is involved (MIT, 2014). Therefore, other techniques, allowing specific and targeted data computation while keeping the data secret, need to be used. Secure Function Evaluation (SFE), Fully Homomorphic Encryption (FHE) (Gentry, 2009) and Functional Encryption (FE) (, and partition of data on non-communicating data centers, can help solving the limitations of traditional security techniques.
Homomorphic encryption is a form of encryption which allows specific types of

computations (e.g. RSA public key encryption algorithm) to be carried out on cipher text and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintext .

An important security and privacy challenge for Big Data is related with the storage and processing of encrypted data. Running queries against an encrypted database is a basic security requirement for secure Big Data however it is a challenging one. This raises questions such as a) is the database encrypted with a single or multiple keys; b) does the database needs to be decrypted prior to running the query; c) do the queries need to be also encrypted; d) who as the permissions to decrypt the database; and many more. Recently a system that was developed at MIT, provides answers to some of these questions. CryptDB allows researchers to run database queries over encrypted data. Trustworthy applications that intent to query encrypted data will pass those queries to a CryptDB proxy (that sits between the application and the database) that rewrites those queries in a specific way so that they can be run against the encrypted database. The database returns the encrypted results back to the proxy, which holds a master key and will decrypt the results, sending the final answer back to the application. CryptDB supports numerous forms of encryption schemes that allow different types of operations on the data (RA Popa & Redfield, 2012). Based on CryptDB, Google has developed the Encrypted Big Query Client that will allow encrypted big queries against their BigQuery service that enables super, SQL-like queries against append-only tables, using the processing power of Google's infrastructure.

Fully homomorphic encryption has numerous applications, as referred. This allows encrypted queries on databases, which keeps secret private user information where that data is normally stored (somewhere in the cloud – in the limit an user can store its data on any untrusted server, but in encrypted form, without being worried with the data secrecy). It also enables private queries to a search engine - the user submits an encrypted query and the search engine computes a succinct encrypted answer without ever looking at the query in the clear which could contain private user information such as the number of the national healthcare service. The homomorphic encryption also enables searching on encrypted data - a user stores encrypted files on a remote file server and can later have the server retrieve only files that (when decrypted) satisfy some boolean constraint, even though the server cannot decrypt the files on its own. More broadly, the fully homomorphic encryption improves the efficiency of secure multiparty computation.

PREFERRED STUDY IN A SECURE SOCIAL APPLICATIONS IN ELECTRONIC MEDIA;

Social networks are one of the key-applications for a large number of users. Millions and millions of persons are connected to some kind of social network – e.g. Face book according to its own accounting has more than 829 million daily active users on average (654 million with mobile access). Currently, social network platforms already present a set of pre-defined but limited content privacy and security sharing controls Major social network platforms offer the possibility for users to share content under specific privacy rules, which are defined by the social platform and not by the end-user. Most of the times, these rules are extremely permissive and differ from platform to platform. Also, on social networks, content is shared in a non-protected manner, making it easier for unauthorized usage and sharing. Users are also bound by subsequent privacy policies changes that threaten more and more the user right to protect its personal information and personal content.

The other problem that is most of the times associated with the security and privacy of content shared on social networks, is related to the security of the social network platform itself The exploitation of the social network infrastructure can lead to security and privacy threats. On the other hand, recently on the media there have been some allegations about the cooperation of some of the most important IT suppliers (including some major social platforms) with governmental agencies to allow the unauthorized access to user's information and content. This latter fact is quite relevant, because, in theory, the social network service supplier has unlimited access to the information and content of all its customers.

This is an increasing serious problem, not only for end-users but also for organizations. More and more, organizations rely on social network services as a mean to disseminate information, create relations with and between employees and customers, knowledge capture and dissemination. The privacy and security challenges presented by these new ways of communication and interaction are very pertinent topics for both end users and organizations.

The continuous growing proliferation of mobile devices (mostly smart phones and tablets, but soon more devices will enter this scenario) with capabilities of producing content (mainly audio recordings, videos and pictures) at the palm of every user's hand, following them everywhere and anytime is also a serious threat to their content privacy and security. This user generated content creates cultural, symbolic, and affective benefit including personal satisfaction, enhanced skill or reputation, improved functionality for existing games or devices, community building or civic engagement. In more simplistic terms, user generated content creates value, economic or not.

Having all of this into consideration, it seems clear that it is necessary to have a clear separation among the social network platform providers, their social functionalities, and the user generated content that they hold. It is important to create mechanisms that transfer part of the security and sharing control to the end-user side. Having this into consideration, in this section, it is proposed and presented a paradigm shift that implies a change from the current social networks security and privacy scenario based on a social network platform centric, to another paradigm that empowers social networks users' on the control and safeguard of its privacy, passing the user generated content sharing control to the end-user side, using rights management systems Also, the entity that is responsible for the storage and protection of the user generated content is independent of the social network platform itself.

This new approach creates a mechanism that protects the shared user generated content on the social network platform while it provides the content sharing and access control to the end-user.

OVERALL SYSTEM MODEL

As referred on the previous section, the novel approach that is followed is based on open rights management systems – in particular, and for this sake, it is based on OpenSDRM. OpenSDRM is an open and distributed rights management architecture that allows the implementation of different content business models. Moreover, OpenSDRM was created having into consideration interoperability aspects that permit that the different modules that compose the system to be decoupled and re-integrated to allow interoperability with other non-OpenSDRM components, using an open and well-defined API (Figure 3).

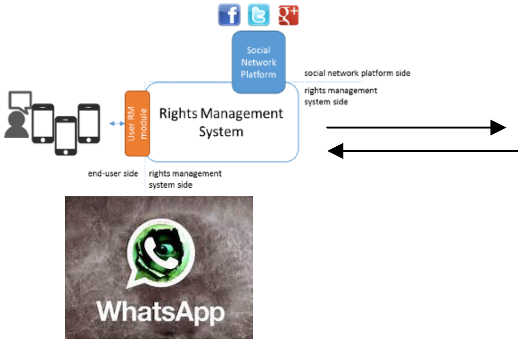


Figure 3 - Overview of integrated with the rights management system by using whats app and Face book and social media mutually affected some data threats.

For the proposed scenario, the social network platform can be integrated with the rights management system, using different methods. If the social network implements a development API or if it is open-source, a much tighter integration scenario can be achieved. If not, it is possible to use other publicly available mechanisms on the platform (or out of the platform) to enable a lesser integrated scenario, but that maintains the privacy and security characteristics sought. Using mechanisms on the platform is the most common scenario and therefore is the approach that will be reflected here.

In this architecture there are some elements that cooperate in order to provide the necessary functionalities to both the end-users and the social network platform, in order to implement the necessary mechanisms to provide security and privacy to user generated content.

On the server-side, is where a large part of the rights management responsibility lies. A set of decoupled components with a well-defined API that allows an integration between the necessary ones to implement the specific content business model. These services are the following:

- ✓ Content storage and distribution service: this service is responsible for the storage and distribution of user generated content in a protected manner in an appropriate storage data basing technology;
- ✓ Content protection service: the service is responsible for the protection of the content. The content is protected by specific protection tools and specific protection mechanisms that may change according to the content and the business model that is going to be implemented;
- ✓ Content registration service: this service is responsible for registering the content on the platform that will be used to uniquely identify the content on the system. This unique identifier is used to identify the user generated content throughout the entire content lifecycle;
- ✓ Payment service: if the business model includes the possibility to trade content, this payment service is responsible to communicate with a payment gateway that implements the necessary mechanisms to process payments;
- ✓ Protection tools service: this service is responsible for the registration of content protection tools on the system and for making those tools available for the content protection service to use when implementing the content protection schemas (such as encryption, scrambling, watermarking and others);
- ✓ Authentication service: handles the registration of users and services on the system as well as the requests for authenticate users on behalf of other services;
- ✓ Licensing service: this is one of the most important services of the rights management framework, responsible for creating license templates, define and produce new content licenses (that represent the type of rights, permissions and/or restrictions of a given user, or group of users, over the content) and provide licenses, upon request, to specific users.

SHARING PLATFORM : Today the latest technology for information sharing to many number of platforms. The other important functionality on the system is the sharing of user generated content (UGC) on the social network. This sharing mechanism is performed through the rights management platform, and the content is stored securely on a configured location (it can be on a specific storage location, on the social platform or on the rights management platform). When the user uploads user generated content, the content is protected and the rights, permissions and restrictions about the content can be defined by the user.

In a brief discussing way, the user has generated content is uploaded to the rights management platform, the access rights and permissions are defined by the user, the content is protected, and a URI is returned to be shared on the social network platform.

The novel content sharing process, using the mechanisms described in this topic, can be now defined in the following steps:

1. The user sends the user generated content (UGC) that it expects to share on the social network. This UGC is uploaded through the content rendering service (CRS). This service requires the user to enter its credentials (email and password), if the user is not yet authenticated. These credentials are used to access the secure storage: SkSStorage = SHA1[email+password];

2. The CRS contacts the AUTS, which reads from the secure storage the user rights management system credentials: CASUUID;

3. The CRS uploads to the content protection service (CPS) the UGC and sends the user credentials, obtained in the previous step: UGCUUID, CASUUID;

4. The CPS, after retrieving some metadata information about the UGC (such as the type, the

format, the encoding, among others), contacts the protection tools service (PTS), requesting a list of available protection tools, that can be suitable to protect the UGC. The PTS sends its credentials and some information about the content: CASCPS, UGC_info;

5. The PTS also returns a list of protection tools that match the request made by the CPS. This information is signed by PTS: KprivPTS{protection_tools_list};

6. The CPS returns the list of protection tools to the CRS, and presents it to the user. The user selects the most appropriate protection tools, adjusting the parameters of applicability of the tools to the UGC and submits its request about the necessary protection tools;

After this process is completed, the UGC shared by the user is shared on the social network platform. The user can also use the social network sharing mechanisms as a way to control how the UGC is propagated on the social network. But, in order to have a fine grained control over the UGC, the user needs to use the rights management system to produce specific licenses with the conditions under which the UGC can be used. These licenses are produced in multiple formats (either in ODRL or MPEG-21 REL). In addition, these licenses are used to support the expression of rights over the UGC. Therefore, when the user uploads user generated content to the rights management system, and after the process that was described previously, the subsequent steps are the following:

1. The CPS contacts licensing service to obtain the appropriate license template for the specific UGC, which was previously created: LICTPL [UUID]. The license template is an XML-formatted document that contains parameterized fields that can be adapted to specific rights situations;

2. A typical license template for user generated content would be composed by following elements:
a. User unique identifier (UUID), multiple users (UUID1, UUID2,…, UUIDn) or a group identifier (GUUID): these fields represent the unique identifiers of the users or groups to whom the user generated content is going to be shared;

b. The unique identifier of the content: UGCUUID;

c. List of permissions (Permission1…Permission n);

d. List of restrictions (Restriction1…Restriction n);

e. Validity date (validity);

3. The license is stored on the licensing service, where it can be accessed by legitimate users.

ACCESSING CONTENT ON THE PLATFORM

Finally the last process in this case-study is to present how the users can access user generated content that was shared by other users on the social network platform. In order to do that, the user needs to be registered on the social network platform and on the rights management system.
When navigating through the timeline of the social network platform, user generated content that was shared over the social network platform, is presented in the form of a special URI, that, when clicked, is intercepted by the rights management platform, and the access process is started.
The referred process is described in the following steps:
1. The CRS, while trying to render the content that is shared on the social network platform, detects that it is protected content, and contacts the authorization service to access the appropriate information to try rendering the content;

2. The user authenticates to the system using the authorization service, supplying its credentials (email and password) to unlock the secure storage and retrieve the user information;
3. The authorization service, using the UGCUUID embedded on the URI, checks if a license for this UGC already exists on the secure storage. If a license already exists:
a. The authorization service checks the license contents, validating the license digital signature and verifying the UGCUUID;

b. If the UGCUUID is the right one, the Validity is checked and the list of permissions and restrictions are evaluated;

c. If the conditions are met, the content can be deciphered and rendered by the content rendering service. The content encryption keys can be retrieved from the license, and used to decipher the content: KprivU(KpubU(CEK[1] … CEK[n]))= CEK[1] … CEK[n];

After this process is executed, the access to the CRS can be granted or not, depending on the conditions expressed on the license. For simplicity sake, there are several other processes that were not included in this description, such as, for instance, the verification of the protection mechanisms that were applied to the content, and the download of the appropriated mechanisms to allow the local temporarily unprotected version of the user generated content to be rendered.

A CASE STUDY FOR AN BIG DATA MINE INTRUSION DETECTION / PREVENTION SYSTEM ON A SOFTWARE-DEFINED NETWORK

This section presents and discusses a case study about an intelligent Intrusion Detection/Prevention System (IDS/IPS) belonging to a software-defined network. In this case study, the IDS/IPS behavior is controlled by a Kinetic module). The Kinetic language is an SDN control framework where operators can define a network policy as a Finite State Machine (FSM). The transitions between states of a FSM can be triggered by different types of dynamic events in the network, (e.g. intrusion detection, host state). Based on different network events, operators can enforce different policies to the network using an intuitive FSM model. Kinetic is implemented as a Pyretic controller module written in Python. In order to acquire more details related to Pyretic and Python, consult respectively

In this case study, an implementation of an IDS/IPS security module will be developed, which should behave as follows:

- ✓ If a host is infected and is not a privileged host then it is dropped;
- ✓ If a host is infected and is a privileged (exempt) host then the traffic from that host is automatically redirected to a garden wall host, where some corrective security actions could be issued over that infected host (e.g. clean and install security patches for trying to recover it);
- ✓ If a host is not infected then the traffic from that host is forwarded towards its final destination.

Each time a new packet arrives to the system, the IDS/IPS initially processes that packet and defines the policy to be applied to that packet (i.e. drop | redirect | forward). This policy is then delivered to a second module that implements further MAC functionality, namely the learning algorithm of MAC addresses to enhance the L2 packet forwarding. This second module is the one that effectively forwards or redirects the packet (otherwise if the packet is to be drooped, this second module will not receive any packet at all because it was already discarded by the first IDS/IPS module).

The Finite State Machine (FSM) (see Figure 8) used in the current scenario associates the transition functions previously defined with the appropriate state variables. The FSM definition consists of a set of state variable definitions. Each variable definition simply specifies the variable's type (i.e., set of allowable values), initial value, and associated transition functions. The infected variable is a boolean whose initial value is FALSE (representing the assumption that hosts are initially not infected), and transitions based on the infected function defined previously. Likewise, the policy variable can take the values *drop* or *identity*, initially starts in the *identity* state, and transitions based on the policy function defined previously. The FSMPolicy that Kinetic provides automatically directs each incoming external event to the appropriate *lpec* FSM, where it will be handled by the exogenous transition function specified in the FSM description (i.e. the function *self.fsm_def*). In this way, it is ensured that the FSM works as expected.

EVALUATION SHORT PARADIGM

The network topology used in the current evaluation made with a network emulator is shown in Figure 9. All the evaluation was performed in a single Linux virtual machine (Ubuntu linux).

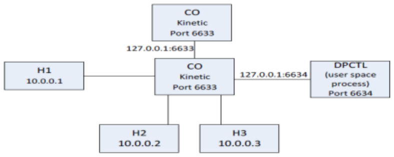


Figure 9 – Network Topology under test

We now initiate the evaluation, opening a Linux shell, and run our Kinetic controller application with the following commands:

```
$ cd ~/pyretic
$                              pyretic.py
pyretic.kinetic.examples.gardenwall
```

As shown in Figure 10, the kinetic controller prints out some results from a verification of network policies using the NuSMV symbolic model checker Kinetic automatically generates a NuSMV input from the program written by the programmer/operator, and verifies logic statements written in CTL (Computation Tree Logic).

BIG DATA SECURITY: FUTURE DIRECTIONS IN DATA MINING TECHNIQUE

Throughout this topic it was possible to present some of the most important security and privacy challenges that affect Big Data projects and their specificities. Although the information security practices, methodologies and tools to ensure the security and privacy of the Big Data ecosystem already exist, the particular characteristics of Big Data make them ineffective if they are not used in an integrated manner. This topic also presents some solutions for these challenges, but it does not provide a definitive solution for the problem. It rather points to some directions and technologies that might contribute to solve some of the most relevant and challenging Big Data security and privacy issues.

Next, two different use cases were presented. Both of the use-cases present some directions that contribute to solving part of the large Big Data security and privacy puzzle. In the first use-case it was presented an approach that tries solving security and privacy issues on social network user generated content. In this approach, an open an interoperable rights management system was proposed as a way to improve the privacy of users that share content over social networks. The processes described show how the rights management system puts the end-users on the control of their own user-generated content, and how they prevent abuses from either other users or the social network platform itself. The second use-case presented the capabilities offered by SDN in increasing the ability to collect statistics data from the network and of allowing controller applications to actively program the forwarding devices, are powerful for proactive and smart security policy enforcement techniques such as active security.

REFERENCES

Advantech. (2013). *Enhancing Big Data Security*. Retrieved from http://www.advantech.com.tw/nc/newsletter/whitepaper/big_data/big_data.pdf

Agrawal, D., Das, S., & El Abbadi, A. (2011). Big data and cloud computing. In *Proceedings of the 14th International Conference on Extending Database Technology - EDBT/ICDT '11* (p. 530). New York, New York, USA: ACM Press. doi:10.1145/1951365.1951432

CTL. (2014). *Computation tree logic*. Retrieved July 17, 2014, from http://en.wikipedia.org/wiki/Computation_tree_logic

DARPA. (2014). *MINING AND UNDERSTANDING SOFTWARE ENCLAVES (MUSE)*. Retrieved August 03, 2014, from http://www.darpa.mil/Our_Work/I2O/Programs/Mining_and_Understanding_Software_Enclaves_(MUSE).aspx

De Cristofaro, E., Soriente, C., Tsudik, G., & Williams, A. (2012). Hummingbird: Privacy at the time of twitter. In *Security and Privacy (SP), 2012 IEEE Symposium on* (pp. 285–299).

Demchenko, Y., Ngo, C., Laat, C. de, Membrey, P., & Gordijenko, D. (2014). Big Security for Big Data: Addressing Security Challenges for the Big Data Infrastructure. In W. Jonker & M. Petković (Eds.), *Secure Data Management* (pp. 76–94). Springer International Publishing. Retrieved from http://link.springer.com/topic/10.1007/978-3-319-06811-4_13

ACKNOWLEDGEMENT :

- At the outset, I wish to acknowledge my heartfull gratitude to the Lord Almighty who is the foundation of the wisdom, prudence and astuteness. God has given me ample resilience, guidance, exhortation and patronage throughout my dissertation.
- Momentous thanks to my beloved Father and Mother Dr.P.Nageswara Rao, Neeraja, my wife Smt.Prabhavathi Nagendra, my children Saranya and Lohith and my godmother Prof.Dr.Indira.S Ph.D. Dean Narayana Nursing Institutions for their love, support and encouragement towards carrying out this project.
- I extend my whole hearted gratitude to our propagator and harvester, My guide, Prof. S. Rama Krishna Ph.D. Professor and HOD SVU-Tirupati who has given treasured advice, herald inspiration, and highly instructive suggestions to accomplish my goal.
- I extend my thank to Dr.K.Vijaya Bhaskar Reddy, MTech, M.B.A., Ph.D. Principal Priyadarshini Engineering and Technology, Nellore. who has given advice, herald inspiration, and highly instructive to accomplish my goal.