

Computer Security with Big Data Analytics

Hugo Virgen

CONTENTS

I	Introduction	1
II	Survey	1
II-A	Early Principles of Computer Security .	1
II-B	Early Principles of Big Data	1
II-B1	Coining Big Data	1
II-B2	Hadoop	2
II-C	Computer Security with Big Data Today	2
II-C1	Zions	2
II-C2	HP Labs	2
II-C3	Identifying Botnet with Net-flow Data	2
II-C4	Anomaly Detection through Event Log Analysis	2
II-C5	Threat Prediction using Big Data	2
III	Technical Underpinnings	3
III-A	Host-Domain Access Graph	3
III-A1	Algorithm for Construction .	3
III-A2	Partite Graph	3
III-B	Belief Propagation	3
III-B1	Algorithm to apply Belief Propagation	3
III-B2	Efficiency Analysis of Belief Propagation	3
III-C	Malware Communication Structure . . .	4
IV	Future Trends	4
IV-A	Scalability	4
IV-B	Heterogenous Data	4
V	Conclusion	5
	References	5

I. INTRODUCTION

The analytics of big data has changed numerous fields. With computer security becoming more important and an interest in computer security; I studied how big data analytics is changing paradigms in computer security. I began by researching how the two topics have developed. After gaining an understanding for the current technologies in computer security employing big data analysis. I learned about the challenges the intersection of these two topics face and possible avenues for how they can improve in the future.

II. SURVEY

A. Early Principles of Computer Security

John Von Neumann outlined Self-replicating automaton in 1949 and Robert Thomas created the first virus in 1970. The Creeper virus ran on Tenex OS and was written in PDP-10, the main purpose of the Creeper was to prove possible a self-replicating automaton. All the virus did was print "I'M THE CREEPER: CATCH ME IF YOU CAN". Creeper demonstrated its ability to self-replicate then connected to another machine on the network and move itself. Ray Thomson received reports of Creeper and immediately created the Reaper program. The program was similar in structure to creeper virus, except as it moved through machines it would remove occurrences of Creeper. Although not widespread, this was the initial battle between computer threat/defense. [5] This event demonstrated a lot of computer security principles that would mature. Despite early computer limited use, the lack of security for computers is highlighted. This initial battle shaped the paradigm for dealing with computer threats, repair and clean the damages after an attack has occurred.

In 1977 scientists Ron Rivest, Adi Shamir, and Leonard Adleman at MIT released a paper describing RSA Algorithm. In 1983 they received the first patent for a computer technology having to do with security. Their algorithm generates public/private keys, the public key is used to encrypt the information, and then the information is decrypted with a private key. [9] At this point people are starting to incorporate isolation as part of the computer security paradigm. Computer security is based around fortifying what they want to defend.

Initially viruses came at a slow rate and so the first antivirus programs to come out only dealt with a small number of viruses. In the early 90s "toolkits" were released which would search a computer for an occurrence of a virus. These toolkits could also remove the virus or create a checksum of the clean file to prevent future infections. In the later 90's anti-virus developers created real-time scanners that would monitor the system and check for virus presence. Developers began applying heuristic analysis to look for viruses by identifying similar characteristics to known viruses. Initially computer security approaches focused on cleaning up threats. This approach of fortify/clean-up quickly developed into real-time scanners. The security paradigm shift is key for current solutions/approaches.

B. Early Principles of Big Data

1) *Coining Big Data*: In 2001, Doug Laney outlined the struggles emerging with managing larger data sets. Laney explained companies would have to adapt. Obviously condensing data increase "operational, analytical, and collaborative consistencies", but "changing economic conditions have made this job more difficult. [7] He outlined volume (the depth of

data available), velocity (how fast data is being collected), and variety (inconsistent semantics) as the 3 principles to deal with for effective data management. These 3 principles are what would be known as big data. He suggests using tiered storage systems, limiting/monitoring data collection, and collecting unique data for dealing with volume; minimizing latency by making data accessible where it needs to be and making the transfer of data as direct as it can be and periodic extraction, integration, and reorganization for effective management of velocity; and profiling data to resolve inconsistencies, making adapters for acquiring and delivering data, using universal translators, and indexing techniques to relate data will limit inconsistencies in data types. These principles become what we know as big data and dealing with these 3 allows for effective analytics.

2) *Hadoop*: In 2004, Doug Cutting began working on Hadoop. Hadoop is a system with Distributed Filesystem and MapReduce. Apache Hadoop stores large data sets with Distributed Filesystem(DFS) and processes large data with MapReduce. It is a project built on Apache Nutch. Apache Nutch is a search engine which indexes web pages using node clusters. Hadoop has the same idea using node clusters, but for data storage and data processing. In 2006 Hadoop became a standalone project with development of its DFS and MapReduce system. [2] Becoming an independent project allowed for the processing of big data because it deals with a lot of the consideration laid out by Laney. Therefore, Hadoop set the precedence for big data because it allowed for people to begin managing and controlling large data sets that would not be manageable otherwise.

C. Computer Security with Big Data Today

1) *Zions*: Zions bancorporation in 2012 reported its results with big data applications. They wanted a security system based on data and SIEM was too slow. Machine Learning models did not help as they take too long to train. With Hive and Hadoop they loaded logs for processing in near linear time, while previously on their old security system it could potentially take a day. Previously searching could take up to 20 minutes, but now it happens nearly 20x faster. [8] With the new environment, they can gather meaningful data from firewalls, security devices, website traffic, business processes, and day-to-day transactions to ensure their networks are secure. This was one of the first examples where a corporation took in a variety of voluminous data to identify threats in real time using big data architecture.

2) *HP Labs*: HP Labs designed algorithms to provide information that was more actionable. They introduced a large-scale graph interference approach to identify malware-infected hosts and malicious domains. [3] A belief propagation was used to determine the likelihood of a malicious host/domain. By testing on HTTP requests data set from a large enterprise, DNS requests data set from an ISP, and an intrusions detection system alert data set from enterprises world-wide the researchers were able to establish a high probability of detection.

3) *Identifying Botnet with Netflow Data*: Researchers used Hadoop's MapReduce was used to identify infected hosts participating in a botnet by analyzing large amounts of NetFlow data. [6] Researchers used BotTrack to examine relationships between hosts to track command-and-control channels. The detection works by creating a dependency graph, running the PageRank algorithm, and DBScan clustering. The PageRank algorithm is the most intensive part so it was implemented using a cluster of 12 commodity nodes. They analyzed a data set containing 16 million hosts and 720 million records. The cluster reduced time by a factor of seven since scores are propagated through edges.

4) *Anomaly Detection through Event Log Analysis*:

a) *Detecting APT Attacks*: APT, Advanced Persistent Threat, attacks target high-value assets or physical streams. APTs are carried out by highly skilled and well-funded attackers. Because of how specialized these attacks are, they require a lot of individual assets and are not scalable. [3] RSA Labs has observed that the attacker's action causes deviation from the affected user's normal patterns. At RSA they are using behavioral deviations to act as anomaly sensors where they can track abnormal behavior leading to an investigation or reports of a given user's activity. Beehive can process 1 billion logs in about an hour and identify policy violations/malware infections that would not have been noticed.

5) *Threat Prediction using Big Data*:

a) *Discover Model*: Sapienza and a group of researchers developed Discover, a framework to predict cyber-threats. They noted attackers usually discuss vulnerabilities, choose targets, recruit participants, and plan/execute attacks using on-line forums. Another signal for potential threats is professionals discussing vulnerabilities, threats, and defense measures on online forums. Sapienza and her team compiled a list of cyber-security experts and mined data from their social media. They then mined information like date published, URL, and contents from the blogs of security experts. Lastly, they mined dark web forums. Sapienza and her group mined from 263 dark web forums that relate to the information security experts' social media and blog. Then they designed 4 dictionaries to filter out words they know will not be beneficial for analysis. They filtered out common English words, technical terms that does not relate to emerging threats, general security threats, like hack or malware, and common Italian words since some experts are italian. From the remaining words they test for recurring words that occur with a term from the threat dictionary, and this is how they generate errors to possible attacks. According to their results they reached an average threat prediction of 84%. The Discover framework has proven to effectively generate warning for threats like wannacry or cloudpets.

b) *Analyzing North Korean Cyber Attacks*: North Korea has been systematically improving their cyber warfare capabilities since the 1990s. Their attacks are well-coordinated attack usually requiring 3 to 7 months of preparation. These attacks are carried out in neighboring countries and target financial, press, or government institutions. South Korea's deals with these attacks by minimizing the damage after the occur. Since these attacks can be sector specific and sometimes

not completely traceable, analysis of attacks is limited and restricts learning about North Korea approaches. Lee proposes a countermeasure system where sharing information and subject of attacks for sharing malicious code, origin of attack, and hacking info. Currently big data analysis in cyber security is focused around large-scale log analysis, abnormal transactions and actions, and detecting malicious. Lee proposes real-time information sharing based on a standardized format. With vast amounts of information collected in real time, Lee proposes the use of analysis methodologies such as machine learning and cluster analysis to detect changes in cyber-attacks. The infrastructures would prevent more North Korea cyber-attacks because resources can be concentrated on predicted points.

III. TECHNICAL UNDERPINNINGS

To learn about the technical underpinnings, I focused on algorithms that detect malicious nodes based on a probability through event log analysis. To do this a graph is constructed from event logs recorded by a company. The graph shows connections between hosts and domains. A host will be a node on the known network and a domain will be any external node while an edge will indicate a connection between host and domain. Then a belief propagation is applied on the graph to calculate the probability of a node being malicious. The purpose of the belief propagation is to uncover malware communications that would normally go unnoticed. A belief propagation is used compared to calculating the probability distribution for every node.

A. Host-Domain Access Graph

1) *Algorithm for Construction:* The host-domain graph will represent who queried what. Therefore, to construct the graph the domain and source IP address is extracted from event logs. The source IP represents a host node and the domain requested represents a domain node. The host nodes are added to a set V_1 , the domain nodes are added to a set V_2 , and any connection between a host to domain node is added to an edge set E . Therefore, the resulting graph is $G = (\{V_1\}, \{V_2\}, \{E\})$, where V_1 is the set of hosts, V_2 is the set of domains, and E is the edges between them. [10] A graph can be constructed from DNS records or HTTP record as both have a source IP attached and destination which can be mined.

After construction based on the message requests, the nodes need to be initialized. There are three main kinds of nodes; malicious, benign, and unknown. Malicious nodes are initialized using publicly available blacklists or privately acquired blacklists. Likewise, benign nodes are initialized with a whitelist of nodes that are known to be not malicious. There are online rankings like popular domain lists or the Alexa ranking which can be used to initiate benign nodes. The remaining nodes are initialized with values of malicious as .5, and benign as .5 which would indicate an unknown state. The edges are bi-directional since the intent is to identify malicious domains alike which could be command and control servers and communication would occur both ways. The connections in a graph can be unique, but in the case of a host-domain graph this is nearly impossible due to the volume of events.

$$m_{ij}(x_j) \leftarrow \sum_{x_i} \phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki} x_i$$

Fig. 1. This equation is the mathematics showing a message is the product over all messages going into node x except for the message from node j

$$b_i(n_i) = k \phi_i(x_i) \prod_{j \in N(i)} m_{ji}(x_i)$$

Fig. 2. This equation is the mathematics showing the belief at a node is the product of current belief at that node and all messages coming into it

2) *Partite Graph:* The resulting graph from construction is a bi-partite graph. A graph is called R-partite if it partitions every end of a connection into separate R classes. A Key characteristic is the lack of odd cycles. [4] A bipartite graph is complete because when it is constructed a node is added to the set when it appears at the end of a connection.

B. Belief Propagation

1) *Algorithm to apply Belief Propagation:* The belief propagation algorithm is used to approximate the probability distribution of a node.

In the belief propagation algorithm, a node i sends its neighbour j a message about what state it is in. A message is a vector of the same dimensions. The belief at node n is proportional to all messages being received at node i and the current belief at that node Fig. 2. The belief at a node has a constant to normalize the sum of incoming beliefs to 1. Messages sent out from a node are created at that node where the message is the product of all messages coming in Fig. 1. Belief propagation gives the exact marginal probabilities for all nodes in a singly-connected graph. To practically compute marginal probabilities, we start with the edges of the graph for which we know and iterate through from there. [11] The time to compute all beliefs for the nodes is proportional to number of links in the graph which is less than exponential time when calculating marginal probabilities naively. Starting with a principle set of messages, one iterates through until the values converge. Once the values converge, we take the individual component of interest, the malicious belief, and compare it to a threshold value. Nodes that are over the threshold are identified as malicious. Figure demonstrates a graph's initialized states, and Figure shows how the host nodes would be updated after the first set of messages are sent out by the domain nodes.

2) *Efficiency Analysis of Belief Propagation:* The Belief propagation algorithm is iterative and so, first it must be identified that the input size is the number of edges. This is because a message is created for every connection between nodes. Then the basic operation is the summation of the marginal belief at a node and the messages received. The basic operation is dependent only on the input of messages from surrounding nodes. Therefore, the number of times the basic operation executes is the number of connections at every node. This shows that the summation is

$$\sum_{i=1}^n i = \frac{1}{2} n^2$$

which indicates an order of growth that is at worst n^2 .

C. Malware Communication Structure

Zhou and researchers confirmed out of 1,172 samples, 93 percent were turned into bots for remote control. These bots use HTTP-based traffic to receive commands. Most command and control servers are controlled by the attackers. Although sometimes they are cloud based. The researchers also confirmed that 138 samples relay phone numbers, 563 samples gather and relay phone numbers, and 43 collect user accounts. These malwares communicate the information gathered to a command and control server. In some cases, the server address is stored in a plain-text file. A malware that turns a host into a bot must receive commands from a command server. There is some bot malware that have a secondary command server which utilizes large scale blogs to push out encrypted updates for the direct-action command server. Researchers also confirmed some malware intentionally cause financial charges. An attacker can cause victims to subscribe to a premium-rate service by sending an SMS message.[13] When confirmation is required, these malwares will respond in the background. 55 of the samples were confirmed to send messages to premium-rate numbers. Identifying that a portion of malware require constant communication is important for identifying malware using a domain-access graph. The more hosts communicate to a suspected malicious website or a website with a confirmed malicious ability, the higher probability the node gets marked as malicious. Understanding the communication structure of malware helps prevent data breaches, data loss, and other security risks.

IV. FUTURE TRENDS

A. Scalability

The paradigm for scaling big data is changing. Previously, analysts and researchers would deal with scaling issues through the concurrent development of processing power because of Moore's law. Although data volume is increasing faster than the rate of processing power. Therefore, scaling has become a serious issue. Scientists have been dealing with decreasing processor speeds by adding more cores. While this may seem like a solution, it only adds to the complexity of processing big data because now the analyst must consider the parallelism occurring inside of the node. Parallelism within the node coupled with parallel nodes leads to complex systems that do not make the use of big data feasible.

One of the ways scalability is being dealt with is with changing input and output subsystems. Data processing engines were formatted around Hard Disk Drives (HDDs) which are slow when it comes to matching computer processing speeds. HDDs are being replaced by new memory technologies. [1] A lot of these new memory technologies are significantly faster therefore subsystems for data processing systems can be designed differently.

The improvement of hardware allows for scaling big data by modifying concurrency control (CC) methods. Zhou and his group of researchers identify the code of concurrency control is tied together with a Data Base Management System

(DBMS). This causes the architecture of the database to be monolithic. Monolithic databases cause difficulty when scaling because they cause the DBMS to be rigid since its operation is designed for efficiency due to being modeled around HDDs. These researchers propose separating the CC from the DBMS. This separation is possible and theoretically would allow the database to be more adaptable because it could be used for various applications and platforms. [12] This would increase scalability because the DBMS could be more easily adaptable to volume and variety of data. Theoretically this seems like a fantastic opportunity, but the researchers are quick to note that the CC would be deprived of data semantics which could potentially cause slow performance.

Although it will be an ongoing process, within the following four years researchers will leverage new memory technologies to restructure key components. Restructuring how components work will mainly improve dealing with variety in data and increasing the volume of data that can be handled. This restructuring of key components will focus around remodeling how components work like the CC which will open new opportunities to scale big data for applicable use.

B. Heterogenous Data

Data that is collected from humans is very heterogenous by the nature of how humans interact with their devices. Algorithms that analyze data often require homogenous data sets because computers do not understand nuance. Creating multiple designs that are structured data will improve one of the challenges posed when working with big data, variety, since the type of data will have to be processed less to be actionable.

Incomplete data is another challenge analysts will have to overcome. Incomplete data possess challenges because it leads to skewed results. [1] Analysis that depends on the complete data will lead to inaccurate results. The analyst may be able to account for variations and inconsistent data, but this only complicates the analysis process.

Analysts often deal with heterogenous data or incomplete data by pre-processing it and setting it up so that it can be analyzed. During pre-processing analysts will have to extract required information and expense it into a structured form. Due to the variety of data the information one would want to extract, the data is dependent on the application of the data.

One example of this is the system proposed by Lee, where countries create a uniform system for reporting North Korean cyber-attacks. Lee believes that the lack of information on North Korean cyber-attacks is leading to a knowledge gap. This knowledge gap causes vulnerabilities in cyber defense strategies. He proposes a hypothetical system where any organization can report a cyber-attack to a large-scale data base, and because this data base will be accessible people will be able to analyze the large sets and produce actionable data. In the coming years, organizations will begin to work towards developing a model for uniform data when it comes to architectures that provide common utility. An example would be a hospital doing research on heart disease because having common data to compare and use would greatly benefit everyone. These uniform structures that provide common utility

will be universal and user across organizations because they provide a common service to all working with the data.

V. CONCLUSION

Computer Security is moving towards real-time detection and prevention of threats with there being a possibility of prediction soon. This paradigm is being enabled by big data analytics and big data tools. Big data implementations for computer security focus on organizations and large-scale implementations. Currently, efforts focus on managing known data, but there is uncertainty in how data variety is going to increase new devices requiring differently structured data. Challenges researchers face is dependent architectures or technologies reaching their theoretical limit or slowing their rate of growth. Future trends that can make real-time threat detection will focus on refining memory structures, data structures, and data processing to make these techniques more scalable and adaptable which will theoretically remove immediate bottlenecks limiting growth.

REFERENCES

- [1] Divyakant Agrawal, Philip Bernstein, Elisa Bertino, Susan Davidson, Umeshwas Dayal, Michael Franklin, Johannes Gehrke, Laura Haas, Alon Halevy, Jiawei Han, et al. Challenges and opportunities with big data 2011-1. 2011.
- [2] Marko Bonaci. The history of hadoop, Nov 2018.
- [3] Alvaro A Cardenas. Big data for security intelligence. page 49, Sep 2013.
- [4] Reinhard Diestel. *Graph Theory*. Springer-Verlag, 2000.
- [5] Alberto Dominguez. Creeper and reaper, the first virus and first antivirus in history, October 2018.
- [6] Jérôme François, Shaonan Wang, Thomas Engel, et al. Bottrack: tracking botnets using netflow and pagerank. In *International Conference on Research in Networking*, pages 1–14. Springer, 2011.
- [7] Doug Laney. Application delivery strategies. Feb 2001.
- [8] Person. A case study in security big data analysis, Mar 2012.
- [9] Paul Tarau. Rsa(cryptosystem). Oct 2014.
- [10] Xin Xie, Weina Niu, XiaoSong Zhang, Zhongwei Ren, Yuheng Luo, and Jiangchao Li. Co-clustering host-domain graphs to discover malware infection. In *Proceedings of the 2019 International Conference on Artificial Intelligence and Advanced Manufacturing*, page 49. ACM, 2019.
- [11] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003.
- [12] Ningnan Zhou, Xuan Zhou, Kian-lee Tan, and Shan Wang. Transparent concurrency control: Decoupling concurrency control from dbms. *arXiv preprint arXiv:1902.00609*, 2019.
- [13] Yajin Zhou and Xuxian Jiang. Dissecting android malware: Characterization and evolution. In *2012 IEEE symposium on security and privacy*, pages 95–109. IEEE, 2012.