Computer Security with Big Data Analytics

Hugo Virgen

***Abstract*—Big Data is a buzzword and computer threats are everywhere. I developed an understanding for how big data is changing the computer security field. I sought current computer security technologies employing big data. I unified what I learned about these technologies and big data challenges. The challenges lies with system architecture and processing of raw data. Improvements in methods utilizing big data are reliant on unifying data like event logs and restructuring sub systems. Unifying data used in analysis before it is processed and restructuring sub systems will lead to better and faster security solutions.**

CONTENTS

LIST OF FIGURES

List of Technical Terms

1) **Monolithic Data Base:** A large and uniform database
2) **Automaton:** A machine performing functions by predetermined steps
3) **Encryption:** Encoding data to limit access
4) **Security Information and Event Management (SIEM):** A solution analyzing activity from multiple sources from an IT structure
5) **Belief Propagation:** Algorithm that passes messages for performing inference
6) **Domain Name System (DNS):** A decentralized naming system for computers
7) **Botnet:** A number of Internet-connected devices
8) **NetFlow:** Cisco developed network protocol for collecting and monitoring network traffic
9) **PageRank:** Search engine algorithm to rank web pages
10) **Density-Based Spatial Clustering of Application with Noise (DBSCAN):** Algorithm to group close points
11) **wannacry:** A windows ransomware which demands ransom payments

12) **cloudpets:** A data breach that lost customer voice recordings
13) **Command and Control Server:** A server operated by an attacker to send commands to bots on a botnet
14) **Moore's law:** The expectation that computation speeds will double every couple of years
15) **Parallelism:** Parallel processing in computer systems
16) **Concurrency Control (CC):** Rules that dictate correct results for concurrent operations
17) **Data Base Management System (DBMS):** Software handling storage, retrieval, and update of data
18) **checksum:** a value compared to the sum of stored values

I. Introduction

Big Data Analytics is changing fields daily. Since computer security continuous growing in importance and a personal interest in computer security, I studied how big data is changing computer security. I studied the evolution of these topics, then I researched computer security technologies employing big data analysis. I learned the challenges big data faces and possible ways to overcome them. Understanding the current state of big data, I made a prediction of future events in big data which will improve computer security.

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 proof-of-concept for self-replicating automaton. The virus simply printed "I'M THE CREEPER: CATCH ME IF YOU CAN". It demonstrated its ability to self-replicate, then connect to another machine on the network and moved. Ray Thomlinson 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. The lack of security for computers is highlighted. This initial battle shaped the paradigm for dealing with computer threats, repair and cleaning damages after an attack.

In 1977 scientists Ron Rivest, Adi Shamir, and Leaonard Adleman at MIT released a paper describing tbe RSA Algorithm. RSA is an encryption intended to secure data. In 1983 they received the first patent for a computer technology connecting 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. [11] At this point people begin incorporating 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 a virus. These toolkits could also remove the virus or create a checksum of the clean file to prevent future infections. In the late 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 and clean-up quickly developed into real-time scanners. The security paradigm shift is key for current methodologies.

Fig. 1. A timeline showing the beginning and major events in big data also the events and paradigm shifts in computer security. The events converge in the early 2010's where the computer security began implementing big data

B. Early Principles of Big Data

1) Coining Big Data: In 2001, Doug Laney outlined the struggles with managing larger data sets. Laney explained companies would have to adapt. Obviously, condensing data increases "operational, analytical, and collaborative consistencies", but "changing economic conditions have made this job more difficult. [8] He outlined volume (the depth of data available), velocity (how fast data is being collected), and variety (inconsistent semantics) as the principles for effective data management. These principles are what is known as big data. He suggests using tiered storage systems, limiting and monitoring data collection, and collecting unique data for dealing with volume. He suggests minimizing latency by making data accessible, making the transfer of data direct, and periodic extraction. For velocity: 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. It stores large data sets with Distributed FileSystem(DFS) and processes large data with MapReduce. It is a project built on Apache Nutch, a search engine which indexes web pages using node clusters. Hadoop also uses node clusters, but for data storage and data processing. In 2006 Hadoop became a standalone project with development of its DFS and MapReduce functionalities. [2] Becoming an independent project allowed for the processing of big data because it begins dealing with the principles laid out by Laney. Hadoop set the precedence for big data because it allowed managing and controlling large data sets not manageable otherwise.

C. Computer Security with Big Data Today

1) Zions: In 2012 Zions bancoporation reported its results utilizing big data. 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 on their old system it could potentially take a day. Searching could take up to 20 minutes, but now it happens nearly 20x faster. [10] With the new environment, they gather meaningful data from firewalls, security devices, website traffic, business processes, and day-to-day transactions helping ensure their networks security. 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 more actionable information. They introduced a large-scale graph interference approach to identify malware-infected hosts and malicious domains. [3] They determined the probabilty of a node being malicious with a belief. They used HTTP requests data sets from large corporations, DNS requests data sets from an ISP, and intrusions detection system data sets from enterprises world-wide; the researchers were able to establish a high probability of detection.

3) Identifying Botnet with Netflow Data: These researchers used Hadoop's MapReduce to analyze NetFlow data and detect infected hosts participating in a botnet. [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 attacker's actions cause deviations from the affected user's normal patterns. At RSA they are using behavioral deviations as anomaly sensors detect abnormal behavior which lead to an investigation or reports of a given user's activity. Beehive can process 1 billion logs in 1 hour and identify policy violations or malware infections that would go undetected.

5) Threat Prediction using Big Data:

a) Discover Model: Sapienza and a group of researchers developed, Discover, a framework to predict security threats. They noted attackers discuss vulnerabilities, choose targets, recruit participants, and plan and execute attacks using online forums. Security professionals are a signal source because they discuss vulnerabilities, threats, and defense measures on online forums. Sapienza and her team compiled a list of security experts and mined data from their social media. They mined information like date published, URL(Uniform Resource Locator), and contents from these blogs. Lastly, they mined dark web forums. Sapienza and her group mined 263 dark web forums that relate to information security experts' would discussed on social media and blogs.They designed 4 dictionaries to filter out words not beneficial for analysis. They filtered out common English words, technical terms that do 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 occurring with a term from the threat dictionary, and they generate warnings 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 improved their cyber warfare capabilities. Their conduct well-coordinated attacks usually requiring 3 to 7 months of preparation. These attacks are carried out in neighboring countries and target financial, editorial, or governmental institutions. South Korea deals with these attacks by minimizing the damage after the occur. Since these attacks can be sector specific and not completely traceable, analysis of them is limited and restricts learning about North Korea approaches. Lee proposes a countermeasure system where sharing information like subject of attacks, sharing malicious code, origin of attack, and other information. Currently big data analysis is focused around large-scale log analysis, abnormal transactions and actions, and detecting malicious intentions. Lee proposes real-time information sharing based on a standardized format. With vast amounts of information collected in real time, Lee believes the use of analysis methodologies such as machine learning and cluster analysis to detect changes in attack methodologies. The infrastructures would prevent more North Korean attacks because resources can be concentrated on predicting attacks.

III. Technical Underpinnings

To learn about the technical underpinnings, I focused on algorithms that detect malicious nodes based on a probability

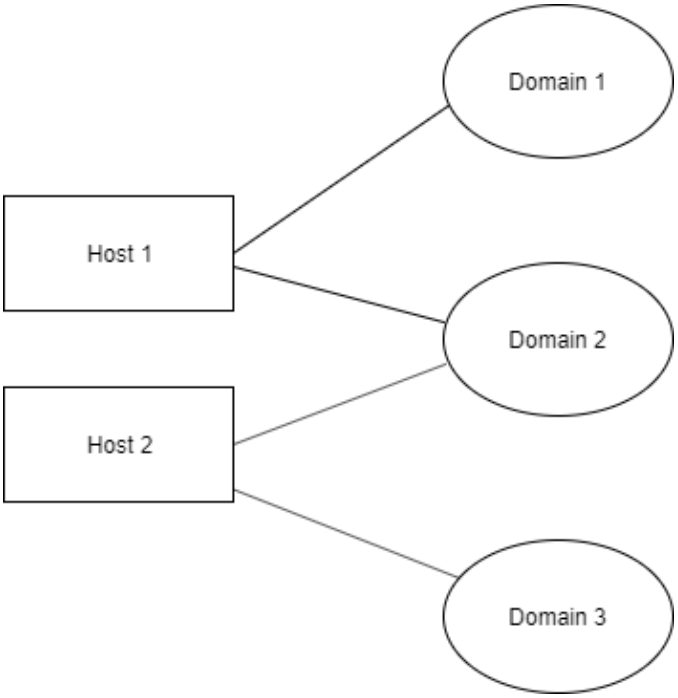

Fig. 2. A graph showing the basic structure of a host-domain graph. Host nodes that are pictures as squares, domain nodes that are pictures as ovals, and lines between nodes are the connections indicating a communication between the two nodes.

through event log analysis. To do this a graph is constructed from event logs. The graph shows connections between hosts and domains. A host will be a node on the known network and a domain will be external nodes while edges will indicate a connection between hosts and domains. 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 communication between nodes. Therefore, to construct the graph the domain and source IP addresses are 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. [12] A graph can be constructed from DNS records or HTTP record as both have a source IP attached and destination which can be mined. [7] Figure 2 demonstrates what a host-domain graph will look like after contruction.

After construction based on the message requests, the nodes are initialized. There are three main kinds of nodes; malicious, benign, and unknown. Malicious nodes are initialized using public 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. 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.

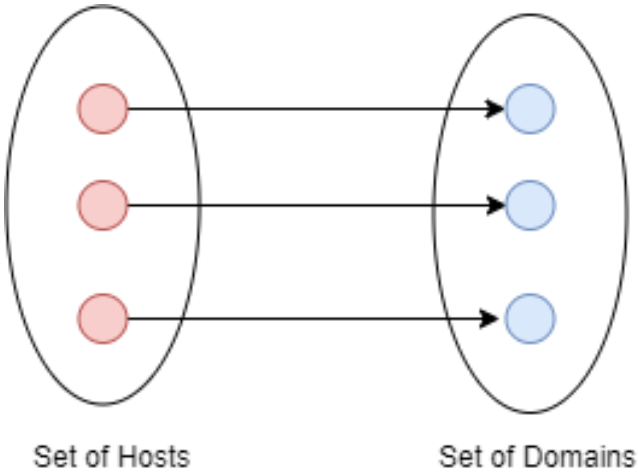

Fig. 3. A visual representation of a parsed host-domain graph. The large oval on the left, with circles inside being host nodes, is the set of host nodes. The large oval on the right is the set of domain nodes where the circles inside are the domains. Lastly, the arrows are all part of a set indicating a node connection between sets.

2) Partite Graph: The resulting graph 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. Figure 3 is a visual representation of a partite graph where every node connection is unique.

B. Belief Propogation

1) Algorithm to apply Belief Propogation: The belief propagation algorithm is used to approximate the probability distribution of a node. The belief at node n is proportional to all messages being received at node i and the current belief Fig. 4. The belief at a node is normalized 1.

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

Fig. 4. The equation showing the belief at a node is the product of the node's current belief and all messages coming into it

In the belief propagation algorithm, a node i sends its neighbour j a message about its state. A message is a vector

of the same dimensions. Messages are sent and produced at a node where the message is the product of all messages coming in Fig. 5.

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

Fig. 5. 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

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. [13] 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 6 shows the application of a Belief Propagation algorithm by demonstrating the changing node states through a single iteration of messages being sent.

2) Efficiency Analysis of Belief Propogation: To determine the efficiency for an iterative algorithm, we must:

1) determine input size
2) identify algorithm's basic operations
3) check if number of times basic operation executes is dependent on only size of input
4) sum the number of times algorithm executes for a basic operation
5) find closed-form formula or establish order of growth

[9] The input size for a belief propagation is the number of edges. This is because a message is created for every connection between nodes. 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

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

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 confirmed that 138 samples relay phone numbers, 563 samples gathered and relayed phone numbers, and 43 collected user accounts. These malware 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

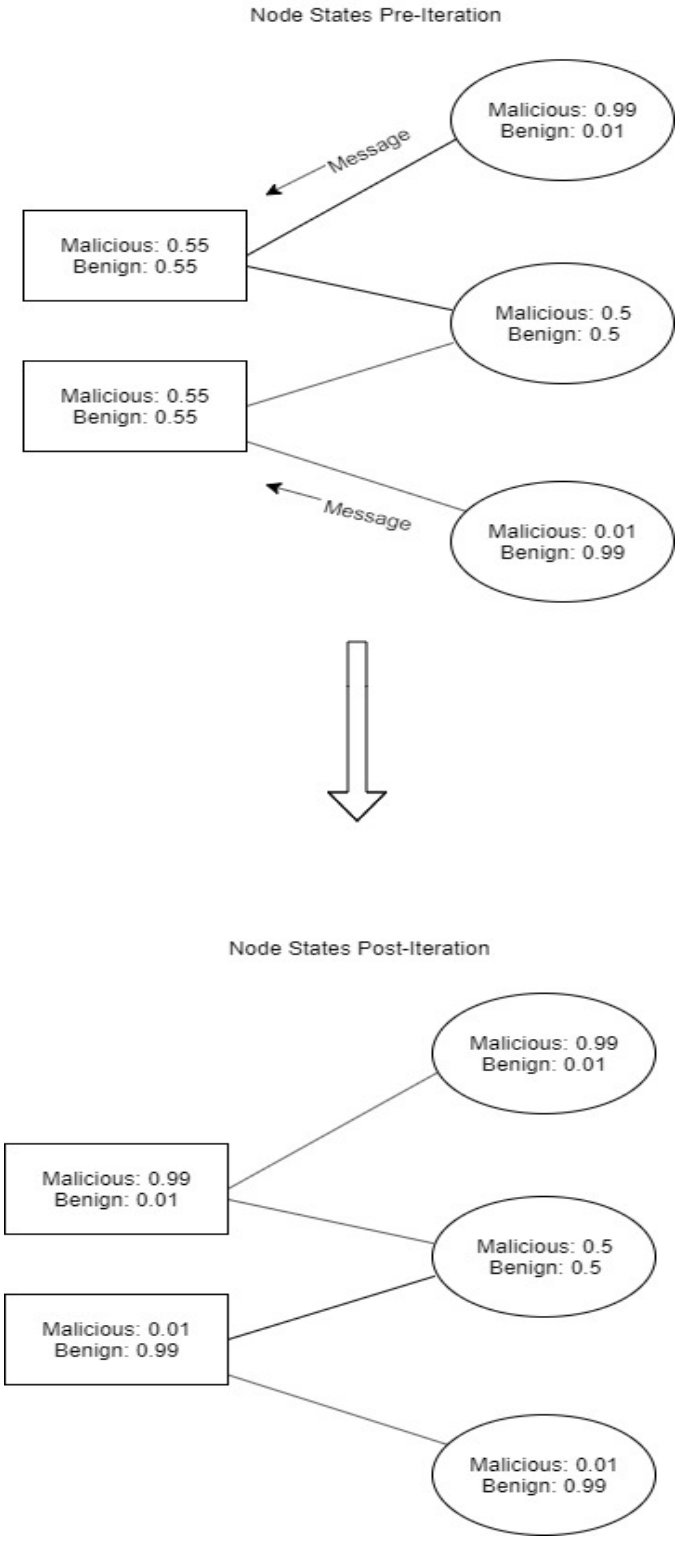

Fig. 6. An iteration of the Belief Propagation algorithm. The top version of the host domain graph is the initial node values and a message passing between connected nodes. The bottom version of the graph is after an iteration with updated node values.

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.[15] 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 dealt 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 improving 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 since it could be used for various applications and platforms. [14] This would increase scaling 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 subsystems. Restructuring how subsystems work will improve managing 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. Heterogeneous Data

Data that is collected from humans is very heterogeneous by the nature of how humans interact with their devices. Algorithms that analyze big data require homogeneous data sets because computers do not understand nuance. Creating multiple designs for 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 heterogeneous 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 large-scale implementations. Currently, efforts focus on managing known data, but there is uncertainty in how data variety is going to increase with new devices producing differently structured data. Challenges researchers face is outdated architectures and technologies

reaching their theoretical limit causing processing speeds to slow. Researchers will focus on making real-time threat detection a possibility by 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. 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] James Kurose and Keith Ross. *Computer Networking*. Pearson, 2017.

[8] Doug Laney. Application delivery strategies. Feb 2001.

[9] Anany Levitin. *Introduction to The Design and Analysis of Algorithms*. Pearson, 2012.

[10] Person. A case study in security big data analysis, Mar 2012.

[11] Paul Tarau. Rsa(cryptosystem). Oct 2014.

[12] 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.

[13] 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.

[14] Ningnan Zhou, Xuan Zhou, Kian-lee Tan, and Shan Wang. Transparent concurrency control: Decoupling concurrency control from dbms. *arXiv preprint arXiv:1902.00609*, 2019.

[15] Yajin Zhou and Xuxian Jiang. Dissecting android malware: Characterization and evolution. In *2012 IEEE symposium on security and privacy*, pages 95–109. IEEE, 2012.

Appendix

My understanding for this topic was built on what I learned at College of Saint Benedict and Saint John's University. While a lot of information applied directly, the skills I learned in CSCI 200, 230, 312, and 338 were crucial.

In, CSCI 200, the Data Structures class, we developed a deeper understanding of java and we were introduced to data structures. Having to implement the data structures increased my problem-solving skills and ability to translate my ideas into code. This directly applied to my project because I implemented a simple belief propagation on a small-scale graph based on an input file. The data structures directly applied because I used them to implement the graph. The problem solving applied because a lot of the principles for how to apply a belief propagation were not explicit. I was also able to analyze how the structure of data can improve or deter from a program. In the end, this class gave me the basis for my project.

In Data Network, CSCI 312, we focused on understanding how networks communicate and the structure of protocols. In learning how networks communicate I learned about protocols like DNS and HTTP. These protocols and communications directly applied because they were how I analyzed an approach that utilized big data analytics for computer security. The structure of the protocols helped me because I understood the location and production of essential data.

Algorithms and Concurrency, CSCI 338, was one of the most useful classes for understanding my topic. I learned most of my topic through the technical analysis and implementation. This class provided me with the ability to analyze and compare competing algorithms and how they differ. This is also where I learned how to construct algorithms, therefore I was able to construct an algorithm to apply a basic belief propagation.

CSCI 230, Software Development, class gave me a very basic view at databases. With this basis, I was able to understand where challenges lie in their use for big data methods. With this basis I was able to make a prediction because I understood how people are improving and experimenting on databases.

These current technologies have deepened my understanding of data structures because I was introduced to new ways people are managing data. I developed my problem-solving skills because I learned about new tools being developed and methods people are using to solve problems. I also improved my problem-solving because I implemented a current technology on a small-scale. My ability to analyze was sharpened because I constantly analyzed algorithms efficiency and actual application. I increased my understanding of networks because I was introduced to communication structures I did not know about. While there is still I do not know about data bases, I learned more basics about data bases improving my overall understanding.