FinalSOTF

Hugo Virgen

***Abstract*—**

I. Introduction

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 ran on Tenex OS and was written in PDP-10, the main purpose of the Creeper was to prove a self-replication automaton possibility. 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 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. [1] 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. The paradigm of cleaning up the aftermath of a security threat was also showcased.

In 1977 scientists Ron Rivest, Adi Shamir, and Leaonard 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. [8] At this point people are starting to focus on computer security by making information and devices difficult to access.

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 data. 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. [6] 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 by 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 through 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. [7] 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: Enterprises are gathering data at rates they cannot handle, therefore HP Labs focused on designing 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 a 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: Hadoop's MapReduce was used to identify infected hosts participating in a botnet because there are large amounts of NetFlow data. [5] 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 is not scalable. [3] RSA Labs has observed that the attacker's action causes the affected user's actions to deviate from their normal pattern. 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.

b) Anomaly Detection Software Defined Networks: Cyber-physical systems are systems that tightly integrate physical and computing processes by monitoring/controlling data through underlying networks. The data these networks used is sensed is used to compute and determine actions. These networks are complex causing problems with scalability; therefore, software defined networks have been more warranted. Software defined networks are nice because they provide flexibility, customizability, and lower processing expenses. Although, these networks are more susceptible to security threats/attacks. Two common attacks are Link Discovery Attack where a link is falsified which gives an attacker control over the traffic and an ARP attack where the attacker pretends to be a specific gateway an creates an invalid address mapping. Both of these attacks are the entry to other attacks. A lot of techniques that deal with these attacks are complex rendering it practically useless or require a subjective threshold which can be unreliable. In this example a predictive model is built to predict the presence of attacks by leveraging 4 machine learning algorithms (Regression, BayesNet, Decision Tree, and Decision Table). The use of big data analytics overcomes the obstacle of monitoring multiple critical measures of communication. The author uses Mininext data to emulate the virtual SDN networks. In the Link Discovery Attack a recipient node sends a message to a controller of a packed using a sender's ID and address. In the ARP spoofing attack, the attacker finds a compromised node and sends it a message pretending to be a legitimate node leading the target node to leak its information. In the experimental setup there was a small network of 13 nodes, a medium network of 21 nodes. In each attack the attack was simulated near the source and near the destination. On a small network all the machine learning algorithms performed over 99

5) Threat Prediction using Big Data:

a) Discover Model: Sapienza and a group of researchers are researching a method to use big data to anticipate cyber-attacks. They named their method Discover. They note that cyber attackers typically identify vulnerabilities, acquire expertise to use the vulnerabilities, choose targets, recruit participants, and plan/execute the attack using online forums. Other signals for potential cyber-attacks such as professionals discussing vulnerabilities, threats, and defense measures also reside on online forums. To mine the data they would analyze, Sapienza and her team compiled a list of cyber-security experts and collected data from their social media. This data is collected every hour and stored to be later recovered. They then formed a list of cyber security experts with blogs. From these blogs they extracted related data with a focus on date published, URL, and contents of the blog. The last place they collected from was dark web forums. Sapienza and researchers collected a list of 263 sites that are forums or marketplaces for hacking. Discover is set up to collect data from the dark web forums relating to the information form the cyber-security experts' social media/blogs and possibly emerging threats. They collected from these forums relating to malicious hacking three times per week. Then they designed 4 dictionaries to filter out words they know will not be beneficial for analysis. The first words filtered out are common English words that would not relate to cyber threats. Then they removed English words that do not mean anything like "to, on, a, for", etc. Technical terms are filtered out because they do not represent emerging threats. Then general cyber threat terms are filtered out because they do not stand for an individual threat themselves. Since some of the expert's tweet in Italian, common Italian words were filtered out. 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

b) Analyzing North Korean Cyber Attacks: North Korea has been systematically improving their cyber warfare capabilities since the 1990s. Most of their cyber attacks are well-coordinated attack that require 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 North Korean Cyber attacks by minimizing the damage caused by these attacks. Since these attacks can be sector specific and sometimes not completely traceable to north Korea, than analysis of the attacks is limited and restricts learning about the attacks. A countermeasure system where sharing the information and subject of an attack is initially proposed for sharing malicious code, origin of attack, and hacking info. Lee explains developed countries in surveillance have set up cross established surveillance systems. Currently big data analysis in cyber security is large-scale log analysis, abnormal transactions/actions, and detection of malicious code. 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 underpinings, 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. Constructing the Graph

1) How to Construct the Graph : The host-domain graph will represent who queried what. Therefore, to construct the graph the domain and source IP address is extracted from the resource records. The source IP represents a host node and the domain requested represents a domain node. The host nodes are added to a set V1, the domain nodes are added to a set V2, and any connection between a host to domain node is added to an edge set E. Therefore, the resulting graph is G = (V1, V2, E), where V1 is the set of hosts, V2 is the set of domains, and E is the edges between them. [9] A graph can be constructed from DNS records or HTTP record as both have a source IP attached and destination which can be mined.

Now that the graph has been constructed 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. Although the construction of the graph only considers the request messages, the edges are bi-directional since the intent is to identify malicious domains alike which could be command and control servers and send instructions to the infected host. 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.

2) 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. A note about bi-partite graphs is the possibility of all unique connections.

B. Belief Propogation

1) Algorithm to apply Belief Propogation: 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 i is proportional to all messages being received at node i and the current belief at that node. 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. 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. [10] 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 the threshold value. Nodes that are over the threshold are identified as malicious. Figure 1 demonstrates a graph's initialized states, and Figure 2 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 Propogation: 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 n(n+1)/2 which indicates an order of growth that is at worst n squared.

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. Some malware families encrypt their URLs to the command and control server. 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.[11] 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. 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. 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. 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

References

[1] First computer virus.

[2] Marko Bonaci. The history of hadoop, Nov 2018.

[3] Alvaro A. Cárdenas, Tudor Dumitras, and Thomas Engel. Big data analytics for security intelligence. page 49. CSA, 2013.

[4] Reinhard Diestel. *Graph Theory*. Springer-Verlag, 2000.

[5] 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.

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

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

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

[9] 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.

[10] 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.

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