Cyber Threat Hunting:
A Cognitive Endpoint Behavior Analytic System

Muhammad Salman Khan, Canadian Nuclear Laboratories, National Innovation Center for Cyber Security, Canada
Rene Richard, Digital Technologies, National Research Council Canada, Canada*
https://orcid.org/0000-0002-1342-6225

Heather Molyneaux, Digital Technologies, National Research Council Canada, Canada
Danick Cote-Martel, Canadian Nuclear Laboratories, National Innovation Center for Cyber Security, Canada
Henry Jackson Kamalanathan Elango, Digital Technologies, National Research Council Canada, Canada
Steve Livingstone, Canadian Nuclear Laboratories, National Innovation Center for Cyber Security, Canada
Manon Gaudet, Digital Technologies, National Research Council Canada, Canada
https://orcid.org/0000-0002-2119-9149

Dave Trask, Canadian Nuclear Laboratories, National Innovation Center for Cyber Security, Canada

ABSTRACT

Security and information event management (SIEM) systems require significant manual input; SIEM tools with machine learning minimize this effort but are reactive and only effective if known attack patterns are captured by the configured rules and queries. Cyber threat hunting, a proactive method of detecting cyber threats without necessarily knowing the rules or pre-defined knowledge of threats, still requires significant manual effort and is largely missing the required machine intelligence to deploy autonomous analysis. This paper proposes a novel and interactive cognitive and predictive threat-hunting prototype tool to minimize manual configuration tasks by using machine intelligence and autonomous analytical capabilities. This tool adds proactive threat-hunting capabilities by extracting unique network communication behaviors from multiple endpoints autonomously while also providing an interactive UI with minimal configuration requirements and various cognitive visualization techniques to help cyber experts quickly spot events of cyber significance from high-dimensional data.

KEYWORDS

Cognitive Analysis, Cognitive Command and Control, Cognitive Machine Learning, Cyber Security Operation Center, Cyber Threats, Endpoint Behavior, Prediction, SARIMA, Streaming, Time Series, Training

1. INTRODUCTION

A Cyber Security Operations Center (CSOC) is a centralized operational facility to continually monitor, identify, analyze, and defend against cyber-attacks and threats. A CSOC should have clear visibility into the data and situational awareness (SA) to enrich cyber analysis with local and global contextual information for identification and detection of threats (Carson Zimmerman, 2014). Cyber adversaries have acquired machine intelligence capabilities to deploy state-of-the-art sophisticated

DOI: 10.4018/IJCINI.20211001.oa9

*Corresponding Author

and autonomous tools to launch and deploy threats (Omid E. David & Nathan S. Netanyahu, July 2015) (Kevin M. Peters, March 2019) (Konstantinos Demertzis, Lazaros Iliadis, April 2015). A continuous war of attrition for both defenders and attackers (James P. Farwell & Rafal Rohozinski, August 2012) has reached a state in which attack objects such as malware are becoming self-aware and smart and are able to successfully penetrate defenses, as demonstrated by recent breaches and attacks (Sana Siddiqui, Muhammad Salman Khan, Ken Ferens, & Witold Kinsner, March 2016) (Sana Siddiqui, Muhammad Salman Khan, Ken Ferens, & Witold Kinsner, July 2017) (Kate O'Flaherty, December 2018) (Sana Siddiqui, May 2017). One of the main problems lies in keeping up with the ever-changing Tactics, Techniques, and Procedures (TTPs) of attacks that are mutating and using advanced intelligent techniques to hide their patterns; these attacks remain beyond state-of-the-art defense tools such as firewalls, Intrusion Detection/Protection Systems (IDS/IPS), and anti-malware technologies (Muhammad Salman Khan, December 2018).

In the current landscape of rapidly evolving cyber threats, a CSOC must be equipped with an advanced suite of tools and technological products that provide complete visibility into the environment and ensure the required security posture of the organization based on risk analysis and processes by a qualified security team. Required defense technologies should be identified based on a combination of the current skillset of the Security Operation Center (SOC) team as well as planned future training requirements. A CSOC should have a capability maturity improvement model to continually enhance the security capabilities. At a minimum, a CSOC should have four capabilities (Babu Veerappa Srinivas, n.d.): (1) Protection and Detection Technologies such as Firewalls, Antivirus, Intrusion Detection System, Intrusion Prevention System, Honeypots, Sandboxes, Endpoint Threat Detection and Response, Malware Analysis, and Forensics, (2) Analytical and Correlation Platforms such as Security Analytics, SIEM, and Visualization Tools, (3) Orchestration Tools such as Workflow Management, Response Orchestration, and Case Management, and (4) Threat Hunting and Intelligence.

2. CYBER THREAT HUNTING

Cyber threat hunting is gaining popularity as the cyber landscape is becoming more complex and dynamic. Threat hunting is a proactive cyber defense methodology that employs searching for threats with little to no knowledge of particular threat objects. In a way, threat hunting can be described as an exploratory cyber data analysis to find events of cyber significance. Furthermore, threat hunting can be defined as iteratively searching through data for either threats that have evaded the underlying cyber defenses or an indicator of a threat that may happen soon (such as any sign of the first stage of a kill chain, i.e. phishing emails or illegitimate port scanning for Reconnaissance stage) (Theodor Liliengren, Paul Lowenadler, May 2018). In a typical SOC, threat hunting commences with a search for threats that have evaded the rule-based cyber defenses but are known through either their behavior or their signatures (Lyndsey Franklin, Meg Pirrung, Leslie Blaha, Michelle Dowling, & Mi Feng, October 2017). Therefore, in this case, threat hunting requires threat intelligence to extract indicators of threats that can then be searched by human cyber experts manually using available tools and technologies. This is different from cyber Incidence Response (IR) methodologies, which are dependent on tools such as firewalls, anti-malware tools, and IDS/IPS. All these tools require configuring rules, writing queries, or updating signatures to detect known threats and generate cyber events. IR processes start event/incidence analysis by triaging the events for which an alert was raised by the defense tools already configured for threat detection (Tim Bandos, June 2019). Conversely, cyber threat hunting aims to uncover new patterns and evidence for threats that are not known or were not captured previously by any cyber defense tools. Threat intelligence does enrich threat hunting tasks but may not be required to start threat hunting (Jai Vijayan, April 2016). Therefore, threat hunting methodology involves four fundamental iterative steps (Robert M. Lee & David Bianco, July 2019) (Chiheb Chebbi, June 2018): (1) creating a hypothesis, (2) investigating by using tools and techniques, (3) uncovering new patterns or signatures, and (4) informing and enriching analytics.

As evidenced by literature surveys (Yousra Aafer, et al., October 2015) (Nitin Naik, Paul Jenkins, Nick Savage, & Longzhi Yang, April 2019), threat hunting is still heavily dependent on manual processes and the hunting methodologies are heavily driven by human skills. Factors of automation and cognitive intelligence are missing (Muhammad Salman Khan, December 2018) (Eric Cole, October 2015) (Steven Schmitt, December 2018). According to SANS, a world-leading organization in cyber security training courses for cyber professionals (Eric Cole, June 2019), cyber threat-hunting processes are still being developed and have not achieved the maturity level to that of reactive incidence handling and response processes. SANS conducted a survey (Eric Cole, June 2019) to assess the status of cyber threat hunting maturity, revealing that less than 3% of organizations follow formal threat hunting processes, while approximately 26% have defined their own internal threat hunting processes. More than 52% of organizations state that they do threat hunting, but these hunting operations are based on ad-hoc processes. Mostly, these organization use already known signatures or knowledge of threats such as indicators of compromise to initiate threat hunting. According to SANS, this is a reactive methodology and is insufficient for true threat hunting. The SANS report (Eric Cole, June 2019) provides three important factors when conducting threat hunting: (1) how long a threat should dwell in the network, (2) the extent of the damage caused by the threat such as lateral movement, and (3) the reinfection frequency, which is defined as how many times the same threat has caused successful damage to the same network. Therefore, to cope with advancing threats, proactive threat hunting should be adopted as a continuous process and should be well integrated with the ongoing reactive cyber security practices of the organization. The same SANS report suggests that increasing the frequency of the threat hunting process will largely decrease the probability of a successful attack or compromise. This is because threat actors are able to introduce mutations in the threat objects for which new rules and signature updates are required in the existing defense mechanisms (Muhammad Salman Khan, Sana Siddiqui, & Ken Ferens, April 2017). Unless the damage has occurred, it is not possible to know the signatures in advance, except if proper hunting is performed that reveals significant evidence of an existing threat evading the cyber defenses (Muhammad Salman Khan, Ken Ferens, & Witold Kinsner, 2015).

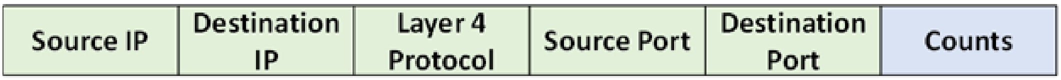
3. FEATURE ENGINEERING

In this work, preliminary feature engineering was applied to extract features from the raw packet capture data to address sufficiency and uniqueness of information representation that can then be used for reliable event analysis.

Machine learning applications emphasize the significance of choosing the right features with acceptable accuracy and precision (David Lopes Pegna, July 2015) (Muhammad Salman Khan, Ken Ferens, & Witold Kinsner, July 2015) (Muhammad Salman Khan, May 2019). It is very important to choose the right features from the raw data that represent and characterize the data as completely as possible (Sana Siddiqui, Muhammad Salman Khan, Ken Ferens, & Witold Kinsner, July 2017). With incorrect features, a reliable machine intelligence model is not possible. In cyber security, features are chosen such that they represent the semantics of the data analysis properly. Mathematical models and guidelines are available for selecting reasonable features from the raw cyber data such that the factors of unique and complete representation, sufficiency, and reliability are achieved within an acceptable confidence margin (Muhammad Salman Khan, December 2018).

In this work, raw data were composed of packet captures formatted in a PCAP data structure and containing Open Systems Interconnection (OSI) layered information starting from layer 2. This work considered the raw packet traffic details from layer 3 to layer 5 that includes information such as the source and destination IP addresses, layer 3 protocol such as IPv4, layer 4 protocols such as TCP or UDP, and layer 5 session ports for the source and destination. Using this information, a virtual packet flow object was extracted showing traffic flow from the source to the destination in layers 3, 4, and 5 as shown in Figure 1.

Figure 1. Flow features



Source IP	Destination IP	Layer 4 Protocol	Source Port	Destination Port	Counts

This flow object was based on the information communication channel between the source and the destination, which is independent of the direction. Therefore, the octets of source and destination IPs and source and destination ports were not important in terms of the direction of the packet flow. A simple statistical count was maintained for each unique flow. For example, if there was a communication between 192.168.0.1:52543 and 10.10.3.1:80 over TCP with 10 packets going outward and 20 packets coming inward, then the unique flow of [192.168.0.1**:**52543**:**TCP**:**10.10.3.1**:**80] was recorded as having a count of 30 packets. The details of the Ethernet protocols were followed from IANA RFC 7042 (Andrew Cherenson, Ashwani Singhal, & et. al., 2019). Likewise, the details of IP protocols are followed through IANA RFC 5237 and RFC7045 (Barry Boehm, Barry Howard, & et.al., 2017). If a new protocol was observed that had not been defined in the RFC, then it was recorded in a separate database and was considered an alert event for the threat hunters.

4. REAL-TIME STREAMING FOR THREAT HUNTING

Big Data is a term that encompasses systematic data engineering and analysis frameworks in a reliable, scalable, and optimized fashion. Various big data processing tools are available such as Hadoop, Spark, Kafka, and ELK (Fairuz Amalina, et al., June 2019), to name a few. Acquiring packet captures from many endpoints in a network can be modelled as a big data problem for an organization (Tom Obremski, July 2016). A big data problem is considered a classical problem of 4 V’s: high **v**elocity, high **v**ariety, high **v**eracity, and high **v**olume (Muhammad Salman Khan, December 2018). Typically, threat hunting on packet capture data using manual analysis requires analysis of the historical data. Threat hunters must always strike a balance in deciding how much historical data to process, going back in time. For successful and reliable threat hunting, it is important to capture packets at line rate, index them in a real-time database, and then write everything to the disk (Tom Obremski, July 2016). However, the main challenges are computing speed, storage options, and database limitations in searching through the historical data quickly. For example, over a 1 Gbps network, a storage capacity of at least 320 TB is required for 30 days, assuming that indexing is done at an optimal level; otherwise, more storage is required (Tom Obremski, July 2016). With the increase in network complexity and scalability, it is common for a medium-to-large organization to have multiple 10 Gbps networks, which is posing a challenge for computing, storage and searching the events. For instance, having four 10 Gbps networks requires at least 12.4 PB storage capacity for 30 days (Tom Obremski, July 2016). Existing threat hunting and analysis methodologies store data for less than 2 months of network activity, ideally (Shannon Kempe, June 2013). Otherwise, the time period is typically only 2 to 3 weeks. With the massive intelligence acquired by the threat actors (Muhammad Salman Khan, Sana Siddiqui, & Ken Ferens, April 2017) and high-frequency mutating malware (Muhammad Salman Khan, December 2018), packet communications must be analyzed for longer time periods, sometimes on the scale of multiple years, to hunt the symptoms of dangling threats in the network that are waiting for the right time to extract data and damage the network or that have already slowly started the compromise. For example, in July 2016, Yahoo disclosed that it had suffered a massive data breach during 2013 and 2014, although it presumably had state-of-the-art cyber defense and hunting tools and capabilities (Taylor Hatmaker, 2017). Although the exact cause of the breach is still unknown, the ability to correlate traffic behaviors and patterns over long time periods to detect anomalies would have been advantageous for the company to detect the symptoms of possible breach at the early stage and hence could have controlled the damage.

As available literature surveys reveal (Zhijiang Chen, et al., April 2016) (Zhijiang Chen, Hanlin Zhang, William G. Hatcher, James Nguyen, & Wei Yu, June 2016) (Duygu Sinanc Terzi, Ramazan Terzi, & Seref Sagiroglu, October 2017) (Riyaz Ahamed, et al., April 2019), researchers have started using streaming mechanisms in real-time monitoring and threat detection to cope with big data challenges. However, these applications of streaming for threat detection are applied to time series modeling in a limited sense, and do not address the challenge of reducing the complexity of multidimensional data and dynamic time series modeling and the correlation of events for threat hunting. Static time series models are available as are various visualization techniques. However, a near real time and machine intelligent threat hunting module that can mimic the cognitive process of establishing a hypothesis, examining data in a time-series fashion, extracting features of cyber significance, correlating them temporally from the past history, and finally comparing the predicted traffic patterns with actual traffic is still at the infancy stage.

A big data streaming analytic model is therefore needed that can correlate data on larger time scales in a real-time fashion to validate the hypothesis for threat hunting. Therefore, in this work, the authors implemented a Kafka-based big data streaming model using time scale correlation configured by the threat hunter (Shuai Zhao, Mayanka Chandrashekar, Yugyung Lee, & Deep Medhi, March 2015) (Chun Xiao, Shenghua Zhang, Qianxiang Zeng, & Xiaofei Cao, August 2018). Kafka is an Apache project that provides a scalable, fault-tolerant, and publish–subscribe messaging system to develop distributed applications for real-time streaming. Kafka was chosen because of its speed: it presents the data structures by offsets of the logs and does not add new message IDs, and is therefore light in the transaction which in turn optimizes the speed of message delivery (Chengwei Wang, Infantdani Abel Rayan, & Karsten Schwan, Dec. 2012). Also, instead of using its own memory cache for writing and reading from the storage medium, it leverages the operating system's cache for file paging, thereby improving time and resource efficiencies. With the big data challenges of packet captures, optimizing the resource consumption for faster message delivery is important (Christian Posta).

5. PROBLEM STATEMENT

Threat hunting is the method or process of proactively searching the network and systems for threats that have evaded existing security measures. Threat hunting is different from reactive cyber incidence management where a particular alert based on pre-defined threat rules is triggered and the incidence response (IR) team looks for the answers to what, why, who, and when questions during the forensic analysis. Threat hunting requires a shift away from a post-attack mentality or approach; it also requires a set of tools for data collection such as Endpoint Detection and Response (EDR), User and Entity Behavior Analytics (UEBA), and logs, and analysis tools such as SIEM correlators and machine learning. Available threat hunting tools require manual analysis whereas the introduction of machine-learning-based tools can automate most of the manual repetitive tasks. An autonomous threat-hunting mechanism uses an advanced suite of machine learning tools that automate repetitive analyses of various heterogeneous logs; furthermore, it provides intelligent data mining and extraction of hidden patterns by adopting both a macro- and micro-view for further decision making using past history, existing data logs, threat intelligence feeds, and situational awareness.

An example in (Mohit Kumar, June 2018) shows a typical threat hunting process. Raw data logs are classified in a two-dimensional space of clients and the ranking of the destination machine the clients are trying to connect over time. These rankings can be considered a threat intelligence database where the cyber status of each destination IP or server is recorded. Information such as how many times the IP has been compromised and what vulnerabilities it is open to is collected. This information is usually required for asset classification but can be concurrently used for hunting as well. Afterwards, destination-based target information is extracted and correlated with the source (client) to provide a map of the communications and activities. This information requires analysis based on threat intelligence feeds (such as if the client and destination communication is considered ex-filtration as per the threat intelligence) and historical analysis of the data logs to attach a level of confidence (local intelligence). This level can be a popularity score or security score based on a multitude of metrics including but not limited to user activities, dynamics, whitelisting, or any situational awareness-related meta data/information. For example, if there is a bot communication, then there are indicators of uniformness in the communication and therefore the score can be reduced for creating alerts. Mathematically, the model can be translated into a flow map time series between the source and the destination weighted by the ranking of both the flows and destination scores.

Another example of threat hunting from an industrial platform is given in (Mohit Kumar, 2018): an endpoint on a network tries connecting to 150 different domains where 90% domain requests remain unresolved. These domain names appear as though an algorithm had generated them. A threat hunter analyzes the history and infers that this event may be an indicator of a Bot traffic as it happens every three hours. Also, it is found out that a few of the domains are resolved successfully, which in turn creates HTTP sessions. Therefore, the threat hunter can find the client address from this session. A temporal machine learning based supervised classification analysis on the data can reveal this pattern easily. Further, the client is not known to the company. Therefore, it can be safely deduced that an unknown Bot is communicating with a low ranked website. Now, the threat hunter communicates with the user of the machine and after analysis of the machine it is found out that it is infected with a malware. This example shows that a Bot threat is detected without the need of any external threat intelligence or malware signatures. This discovery is based on merely analyzing the network flows. Further, there is no need of additional hardware or software to collect the data, as it all comes from the network packet flows collected at the Network Operation Center (NOC). The organization did not invest lot of resources to hunt this threat.

Based on these examples, the authors state the formal problem statement as follows: Is there a way to reduce the cognitive load on human threat hunters by mimicking their mental analytical model using machine intelligence and dynamic and interactive visualization techniques for complex packet data in a real-time fashion?

6. AUTONOMOUS THREAT HUNTING USING COGNITIVE TIME SERIES MODELING

In CSOC facilities, cyber analysts hunt threats by applying various mental models of correlation to validate a hypothesis. In Section V, the authors provide two examples of how a manual threat hunter analyzes events and then validates the hypothesis. In most of the threat-hunting methodologies, a time-series-based mental model (Diego Vidaurre, Stephen M. Smith, & Mark W. Wool, October 2017) is used because an event in the present is correlated with an event in the past to find some similarities. To introduce machine intelligence, it is necessary to transform the problem of mental modeling of data in a temporal fashion to a representative time series. To create a meaningful analysis on time series to mimic mental correlation, it is also necessary to apply some cognitive function to the raw data such that the time series is comprehensible. In this work, it is addressed by applying feature engineering, which behaves like a mathematical function that translates the raw data into a feature as shown in Figure 1. As the feature represents a mental model of individual logical flows between the source and destination in a unique and complete manner, the time series of the counts of these features will represent a virtual threat hunter who is hunting for cyber events while seeing data on a CSOC console temporally. Furthermore, this time series modeling requires autonomous correlation mechanisms at different time instances, and therefore, in this work, a "Seasonal Auto Regressive Integrated Moving Average" (SARIMA) model (Josef Arlt & Peter Trcka, June 2019) was used to mimic the correlation of the count number of the feature between different time instances.

Cognitively, SARIMA model uses an auto-regression mechanism (AR) to regress the feature count from the present state to the previous states using time lags. It is an indicator of the evolution of the count feature based on the historical evolution. Moving Average (MA), on the other hand, represents the deviation of the regressed (or predicted value) from the mean of the data series, and is called the error. Therefore, the combined AR and MA are Integrated (hence the "I" in SARIMA) to cognitively predict a time series in the future based on historical data and to evolve the series as closely as possible to the average for a meaningful prediction. Seasonality refers to the process of removing seasonal trends in the data such that actual evolution can be extracted for prediction. For example, if a seasonal trend is such that the data become double every 3 months (hence a season of 3 months), then the data should be normalized by differencing the next season values from the previous one to get the actual difference. This is cognitively equivalent to removing cumulative effects from the data to show the actual growth rather than the aggregated behavior.

7. CONTRIBUTIONS

In this work, a new threat-hunting mechanism is proposed using cognitive machine intelligence techniques to address the problem statement of Section V. As mentioned in the previous section, typically threat hunters use their mental analysis to correlate data available from various cyber security tools such as SIEM and then deduce the validity of hypothesis. Therefore, in this work, the authors took the same approach of mental analysis and developed a data streaming and analysis framework for raw packet captures to extract flow feature, and then used the framework to learn the behavior and predict the future behavior using time series modeling. In this work, the authors did not take the approach of evaluating the detection performance of the proposed technique, but instead developed a framework aiming to reduce the cognitive load on cyber threat hunters and provide them with cues for validating their hypothesis.

In particular, the following are a few major mechanisms used in building the proposed model and contributing toward a cognitive threat-hunting framework that is real-time and fault–tolerant, and that applies stochastic time series analysis to predict anomalies without requiring labeled datasets for training:

1) Flow features are extracted from the raw data to represent the dynamics of raw PCAP data in a unique manner. As explained in the previous section, flow features are extracted to sufficiently represent the dynamics of the packet flows using the statistical count of flows in either direction and thereby represent the cyber communication in a more meaningful sense. Flow features have been used significantly in reactive threat detection approaches. However, for this work, authors considered flow features as a contribution toward proactive threat hunting methods as this is a natural mental model a threat hunter would apply in validating the hypothesis. Typically, threat hunters use the packet data from NetFlow (SolarWinds) integration with SIEM products to analyze various flows through mental analysis and find correlations. However, using statistical counting, which is a simple yet powerful feature, this work considers a cognitive meaning toward detecting anomalies for threat hunting. The real claim is not developing the flows themselves, but using the flows in a more mental approach toward hunting using transformed time-series-based analytical modeling.
2) In this work, a real-time streaming tool is employed to acquire raw packets to store them in a database for subsequent feature extraction, and then a dynamic time series model is developed both statistically and visually. A real-time streaming data acquisition framework is needed to model cyber events on a temporal scale and to introduce cognitive intelligence similar to how the human brain recognizes events and tries to find a correlation.
3) Combining the SARIMA time series prediction model with the proposed feature engineering for threat hunting is a new topic, and little to no research is available. Therefore, the authors

believe that this approach is a new advancement toward a cognitive threat-hunting mechanism that uses a dynamic interactive time series analysis to address the cognitive aspects of correlation on various time scales concurrently.

4) Using the open-source Dash visualization interactive web-based framework, the authros developed a cognitive time criticality threat-hunting visualization not only to give the threat hunter the required control to correlate time series data using a SARIMA prediction model (Simon Duque Anton, Lia Ahrens, Daniel Fraunholz, & Hans Dieter Schotten, November 2018), but also to present complex multidimensional data using a Sankey diagram in a fast, effective, and substantial way. Therefore, simplifying the complex data representation on a two-dimensional screen with sufficient feature statistics helps the threat hunter validate the hypothesis and then interactively select or reject the results quickly. This improvement in cyber defense tools is instead of relying on the current methodologies of correlating large volumes of packet capture data manually with limited interactive configuration capabilities by existing state-of-the-art SIEM tools.

8. DATA SET

In this work, streaming was simulated using two different PCAP data files from data infected by Bubble Dock and taken from the Stratosphere Research Laboratory at the Czech Technical University (CTU) (Sebastian Garcia, Martin Grill, Jan Stiborek, & Alejandro Zunino, 2014) (Sebastian Garcia, November 2014) (Frantisek Strasak, May 2017). The specific PCAP files used were “2015-07-28_mixed.before.infection.pcap” and “2015-07-28_mixed.pcap” (Czech Technical University, 2015). These files were used to simulate the streaming from two different endpoints offline for the following reasons:

1) This work simulated streaming two endpoints, one infected with Bubble Dock adware malware packet capture and the other without any infection. As the end goal of this project was to read packet captures in a real-time streaming fashion, it was convenient to write a streaming script that could send packets from offline PCAP files at line rate to the streaming module without having to worry about live packet capture agents in each endpoint. This helped author’s focus more on the streaming and the load-balancing part of the project after the packet captures are streamed out from each endpoint. In future, it would be easier to write an endpoint agent pipeline that can be integrated with the streaming pipeline developed in this work. Furthermore, it is also possible to scale up the number of endpoints based on the load-balancing module used in this work.
2) The authors of the CTU data set (Czech Technical University, 2015) mentions that a significant amount of data pre-processing and data cleaning was done before uploading the PCAP files on the CTU repository. For example, some artefact flows were cleaned of certain IPs that were redundant and may have created either noise or bias in the data analysis. Also, all broadcast and multicast packets were cleaned. In the case of a live endpoint agent capturing traffic, a significant amount of analysis and coding was required but was beyond the scope of this work.
3) To capture the traffic data for malware command and control communication over PCAP captures, labeling of the malicious packet flows and validation of those flows for any unnecessary artefact remaining in the files were required. The CTU team also provided a detailed description of the timeline to indicate specific packet flow instances from the endpoint. Live packet capture would have taken considerable work involving development of an endpoint agent, and was therefore considered to be beyond the scope of this publication. However, as per (Muhammad Salman Khan, December 2018), the development of a sandbox environment is planned to capture end-to-end data for a full visibility and analysis.

The endpoint used by CTU for PCAP capture was the Windows 7 operating system. It captured normal traffic flows from July 15, 2015, 17:51:07 CEST until July 26, 2015, 14:41:32 CEST. The following provides a brief overview of the PCAP information from both files (for a detailed description, please refer to (Czech Technical University, 2015)):

1) Capture Start: July 15, 2015, 17:51:07 CEST.
2) Capture and Infection Stop: July 28, 2015, 08:17:45 CEST.
3) Total packets captured (after pre-processing): 1,437,980.
4) Clean packets (after pre-processing): 541,043.
5) Infected packets (after pre-processing): 896,937.
6) Size of "2015-07-28_mixed.before.infection.pcap": 408 MB.
7) Size of "2015-07-28_mixed.pcap": 660 MB.

Therefore, in 1,088,798 seconds, this Windows 7 endpoint created 1.438 million packets of which ~0.897 million packets were infected, or 62.37% infected packets in 12 days, 14 hours, and 27 seconds. As mentioned in (Czech Technical University, 2015), CTU researchers performed various legitimate browsing and other activities to simulate the natural behavior of a user at the endpoint, and they then infected the computer with Bubble Dock malware. Furthermore, as per the above data, the size of each packet data is estimated at ~458 bytes with 1.32 pps (packets per second) rate.

Bubble Dock (Nathan Bookshire, n.d.) is considered an adware malware program that displays pop-up advertisement and other links for marketing and sales purposes. It links itself with all the major browsers. Currently, Bubble Dock is an adware infection designed with the sole purpose of monetizing internet traffic by collecting sales leads from any website. For this reason, Bubble Dock is being used by dubious and malicious websites to infect the target computers in the context of advertisement. It is not a malware object itself, but its use makes the computer vulnerable and sometimes opens it to infection. Bubble Dock installs rootkit into the operating system and is considered a potentially unwanted program (PUP) (Stelian Pilici, 2014). Bubble Dock infection traces can be found both at the endpoint and at the network packet capture level. As it communicates to the online websites for advertisement purpose, it produces packets with the webservers. Furthermore, it adds itself to the Windows 7 operating system process tree and runs internally through a BubbleDock.exe file. Bubble Dock affects both Windows 7 and Windows 10 operating systems.

9. SYSTEM ARCHITECTURE

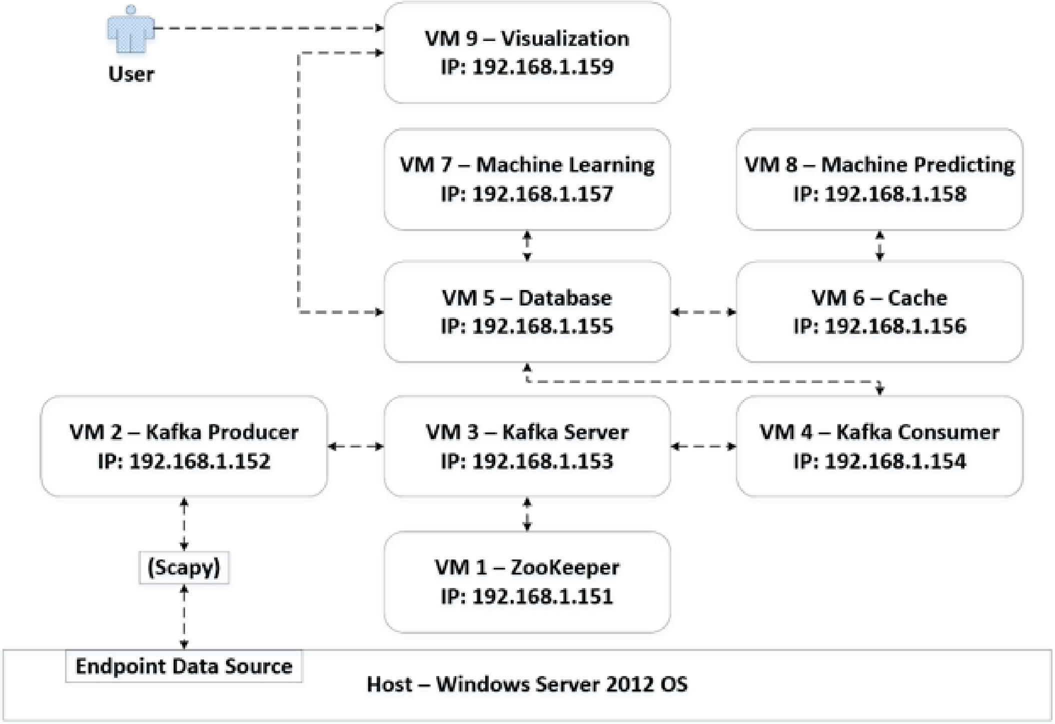
Figure 2 shows a high-level system overview of the experiment setup. In this work, virtual machines were used to set up the required connectivity. The configuration of each virtual machine (VM) was as follows:

1) Operating System: CentOS 7 64-bit.
2) Processor: Intel(R) Xeon(R) Gold 6154 CPU @ 3.00 GHz and 2.99 GHz (2 virtual processors).
3) RAM: 8 GB.
4) Storage: 30 GB.

In addition, the host machine had the following configuration:

1) Operating System: Windows Server 2012 R2 Standard 64-bit.
2) Processor: Intel(R) Xeon(R) Gold 6154 CPU @ 3.00 GHz and 2.99 GHz (2 processors).
3) RAM: 256 GB.
4) Storage: 40 TB.

Figure 2. System architecture of EBAS

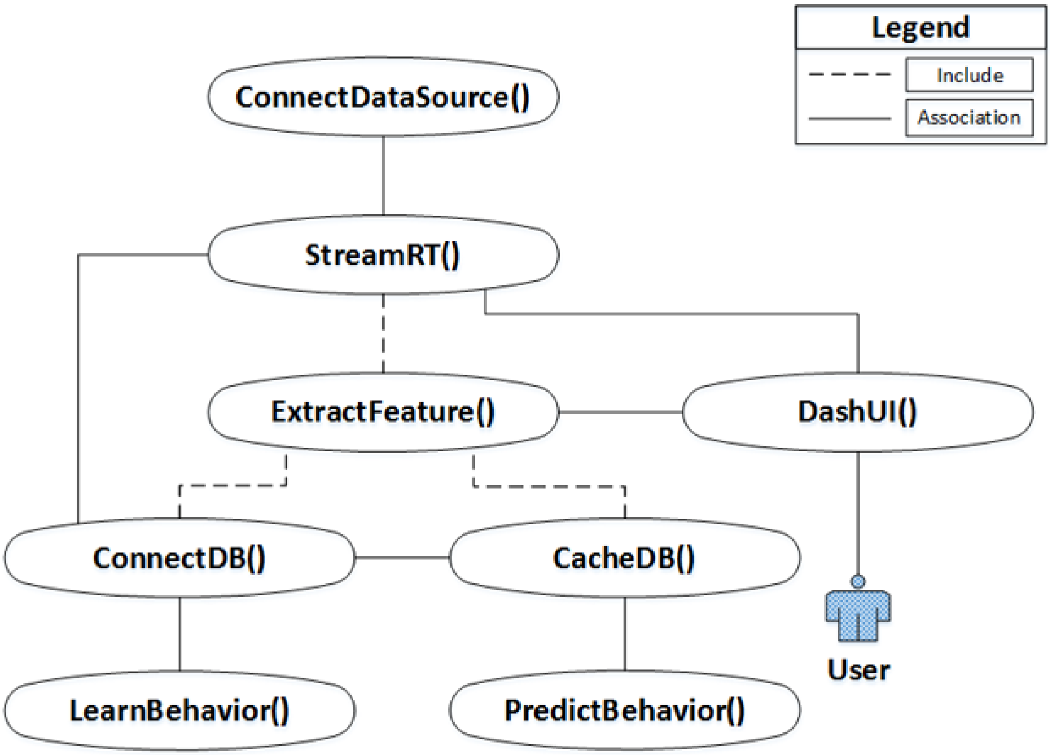


As depicted in this system architecture Figure 2, the end user or the threat hunter will interface with VM 9, which is a front-end interface that takes care of all the requirements for the threat hunter such as displaying the data, showing the extracted features, training, and predicting machine learning, streaming, and database details. VM 1 has a ZooKeeper package (Apache Software Foundation, Apache ZooKeeper, n.d.), which is used for load-balancing the online transaction load of streaming flows (Renato Toasa, Clay Aldas, Pablo Recalde, & Rosario Coral, January 2019). VM 3 is the actual Kafka server (Apache Software Foundation, Apache Kafka - A Distributed Streaming Platform, n.d.) connected through the VM 2 Kafka producer (Apache Software Foundation, n.d.) and VM 4 Kafka consumer (Apache Software Foundation, Apache Kafka - A Distributed Streaming Platform, n.d.). The Kafka producer is connected to the endpoint sources, which in this case are the offline PCAP files stored on the host machine through the Python Scapy package (Scapy Community, n.d.). It is important to note that integrating live endpoint sources through the reconfiguration of this Scapy module is very quick. VM 5 is the relational database SQLite (SQLite Consortium, n.d.) machine that handles all queries, insertions, and reading loads for both learning and prediction. This may be replaced with TimescaleDB. However, for prediction, an additional cache module is used to ensure real-time prediction. This VM uses a memcached module (Dormando, May 2016) (Julien Danjou, n.d.) for high-speed cache write/read operations and to update the VM 5 database with any delta changes in the database. The VM 5 database is required not only for learning the behavior patterns, but also for displaying the filtered data for the features. For example, a feature represented by Figure 1 can have 30 packets, and the threat hunter must look at individual packets represented by the feature after deciding to look at what actual packet transactions happened for that particular feature.

10. SOFTWARE ARCHITECTURE

The proposed cognitive Endpoint Behavior Analytic System (EBAS) system has a software architecture composed of five abstract layers, as shown in Figure 3. The first layer docks the endpoint data source of network traffic. The second layer is a real-time stream processing architecture that coordinates many Kafka workers using a streaming scheme to process raw packets. The third layer includes the database of extracted features. The fourth layer learns the individual and aggregated behavior of endpoints. There is also a cache database for temporarily keeping the raw data to extract features and compare them with the learned model. The fifth layer is the UI layer that interacts with the user for configuration and parameters. Following describes individual APIs developed in this work.

Figure 3. A high-level abstract software development architecture



StreamRT() – This Application Programming Interface (API) object was developed using Kafka and ZooKeeper packages. For ZooKeeper, a cluster of one server was developed in this work, but scaling it with multiple VMs is not difficult. ZooKeeper is used for distributed load-balancing and its fault-tolerant capabilities. The communication architecture of ZooKeeper (Flavio Junqueira & Benjamin Reed, December 2013) uses a handshake mechanism based on a proposal transaction and on Acknowledgement (ACK) and Commit messages. In ZooKeeper, a first-in, first-out (FIFO) message queuing mechanism is deployed that orders the message delivery during the communication. All transactions are tracked using ZooKeeper transaction ID maintained through a 64-bit ID. This ID can be divided into the most significant 32 bits for the epoch timestamp and the least significant 32 bits for the counter. In the ZooKeeper terminology, the master and slave are called the leader and follower, respectively. As it is a distributed computing architecture, there is no fixed leader and all nodes can compete to become a leader (All Programming Tutorials, May 2018). ZooKeeper guarantees the arrival of messages in the order that they are received (Flavio P. Junqueira & Benjamin C. Reed, August 2009). ZooKeeper takes on average less than 200 ms (ZooKeeper, 2019) to recover from failure and to elect a new leader. This provides a dynamic and distributed synchronization mechanism for messaging and transaction for real-time streaming. The experiment used Zookeeper version 3.5.5 with Kafka 2.3.0.

In conjunction with ZooKeeper, Kafka streaming architecture was used to stream data in real time. Kafka architecture is composed of Kafka producer and Kafka consumer connected through a centralized Kafka server. Kafka producer publishes messages to Kafka topic, which is a data stream with metadata such as name. Prior to sending the messages, Kafka server is configured to synch with the producer and the consumer, and uses a queuing mechanism to stream messages between the two. A client node (such as an endpoint) needs to use Kafka Consumer API. For more information on Kafka streaming, refer to (Ableegoldman, n.d.).

ConnectDB() – This API connects the data source and machine learning APIs to the database. It stores both the raw data and the details of the extracted features with unique UUIDs and relevant timestamps.

CacheDB() – This is the memcached connection API to connect the machine learning prediction model to the cache to extract the features from the raw data online. It is a short-term memory, and updates ConnectDB() after sending the data for prediction.

ExtractFeature() – This API extracts the feature attributes from the raw data.

LearnBehavior() – This API is a user-configured module that learns the SARIMA model for the input data for the defined time interval. It can be stopped and restarted through user-defined parameters. It outputs a learned model that is required by the PredictBehavior() API.

PredictBehavior() – Based on the learning model and the user-defined time window, this API correlates the actual live-stream data with the predicted data and outputs a root-mean-square value (RMSE), which is a measure of similarity between the observed and the predicted. A user can configure this value to alert the threat hunter to dissimilar time windows which presents the raw data and feature for that time window on the dashboard.

DashUI() – This API connects to StreamRT() to input the raw data for packet tab and to ExtractFeature() to input feature data.

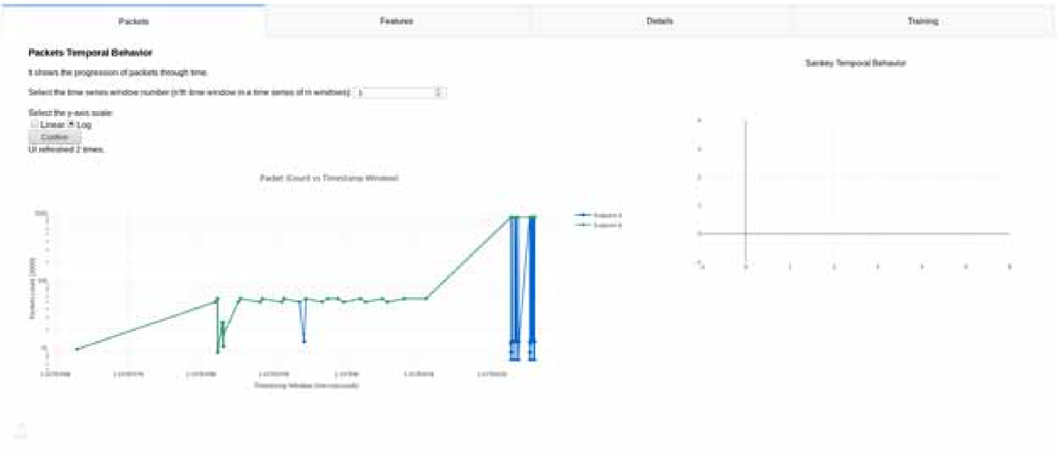
11. NAVIGATING THROUGH THE PROPOSED THREAT HUNTING FRAMEWORK

This section navigates the readers through the process of the proposed threat-hunting model, and shows how visualization relieves the hunter of manual analysis. Figure 4 shows the first tab of the threat hunting interface (dashboard). There are four tabs in total with the first tab showing a time series of raw packet counts for each endpoint. In this navigation, two different endpoints are simulated using two different PCAP files. Each PCAP is associated with a simulated endpoint ID and is assigned a consistent color that remains the same across the platform generated by an internal program. In this walkthrough, endpoint A has the color blue and endpoint B has the color green. Researchers have noted that, after eight endpoints and eight colors, it becomes more difficult to explore data; hence, a management service to divide the endpoints into subgroups is required. The color red is reserved for highlighting anomalous behavior patterns and therefore should not be used to color an endpoint. On the right side of this tab, there is a blank diagram for the Sankey diagram. On this tab, a threat hunter can select the time window number in a time series. For example, if the data have one thousand time windows, the hunter can select to see the behavior at window number 200.

The first tab shows the dynamic packet progression over time, and has two functions. The first function is to display the packet count against the timestamp window. It takes the time series window number given by the input selector and the packets. This User Interface (UI) shows the start date for comparison with the current data. As shown in Figure 5, the user can hover the mouse pointer over an individual point on this raw packet time series to display the information of the particular packet flow that this point belongs to. The reason for offering this feature is to be able to set marker boundaries for the selected area to get a quick overview of the start and end of the window raw packet information, similar to the functionality offered by video editing

As shown in Figure 6, the second function is the Sankey diagram. It changes dynamically when the threat hunter wants to highlight windows of interest and observe the behavior of the communication between the endpoints from a high-level overview. It displays six dimensions representing the endpoint ID, source IP address, source port, destination IP address, destination port, and total number of counts. The width of each flow (for example, there are two flows for the blue endpoint) represents the count weight of each flow for the particular time period (selected window). It allows the threat hunter to compare packets and observe the direction and interaction of the flow. The user can change the y-axis normalization scale to a linear or non-linear scale. Compared to the manual analysis of the packet flows, this scheme provides more prompt and simpler cues to the threat hunter to look for anomalies. For example, it is obvious that the green endpoint has one flow with relatively more counts than any other flow. Sankey also provides a correlation of the flow features between endpoints and within the endpoint itself.

Figure 4. First tab – Raw packets



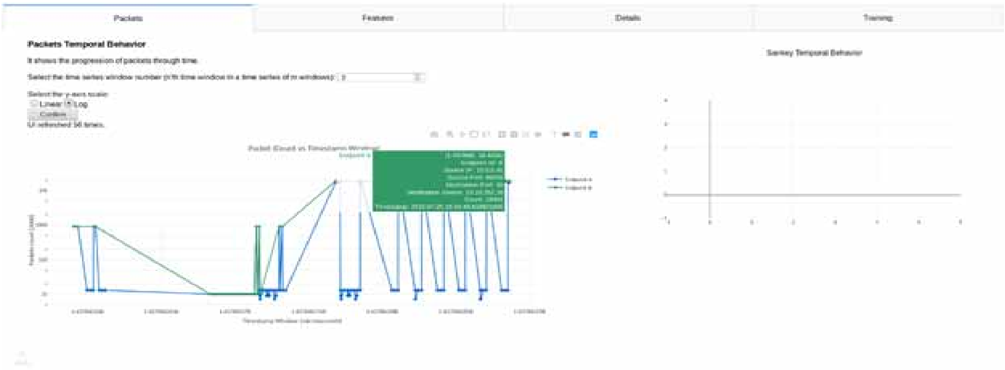
Packets Temporal Behavior

It shows the progression of packets through time.

Select the time series window number (n'th time window in a time series of m windows): 1

Select the y-axis scale:
○ Linear ● Log
Confirm
UI refreshed 2 times.

Figure 5. Information view of a particular packet flow



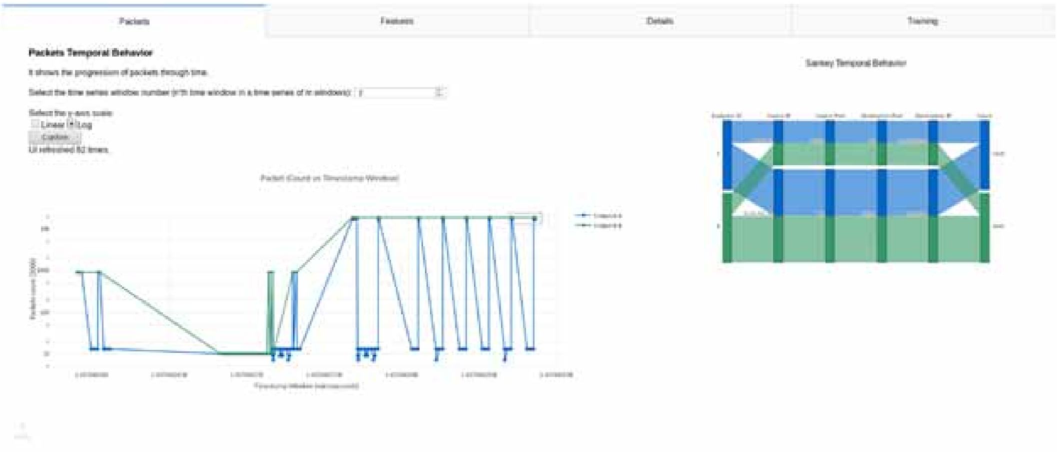
Packets Temporal Behavior

It shows the progression of packets through time.

Select the time series window number (n'th time window in a time series of m windows): 3

Select the y-axis scale:
○ Linear ● Log
Confirm
UI refreshed 36 times.

Figure 6. Sankey diagram of selected packets flows



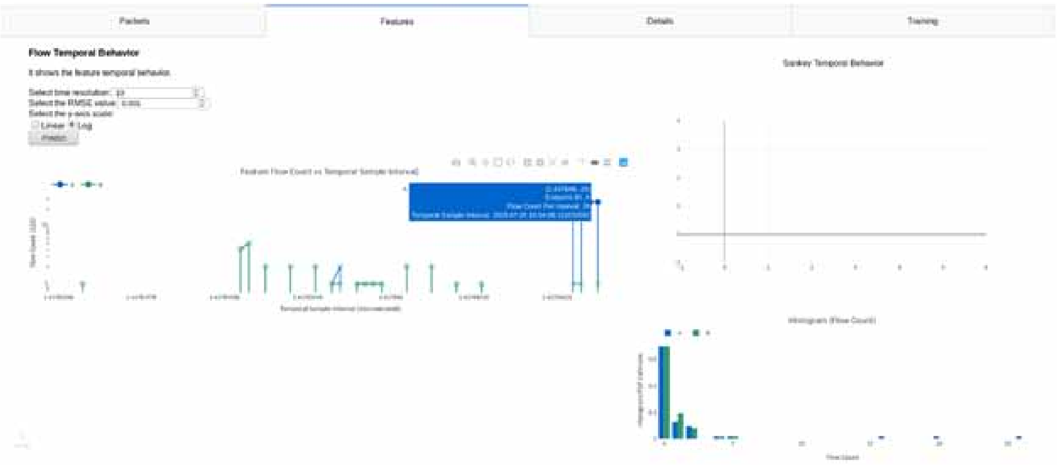
Packets Temporal Behavior

It shows the progression of packets through time.

Select the time series window number (n'th time window in a time series of m windows): 2

Select the y-axis scale:
☐ Linear ● Log
Confirm
UI refreshed 82 times.

Figure 7. Online Feature(s) Extraction Time Series



Flow Temporal Behavior

It shows the feature temporal behavior.

Select time resolution: 10
Select the RMSE value: 0.001
Select the y-axis scale:
☐ Linear ● Log
Predict

As shown in Figure 7, the second tab named "Features" gives the hunter access to the correlation analysis of unique packet flows. It has three diagrams: (1) time series of the flow feature count, (2) Sankey diagram for the selected points on the feature time series, and (3) the probability distribution estimation of the feature data points. In this tab, the time series can be configured for different sampling intervals to either zoom in or zoom out for analysis. For example, given a time series of 12 hours with a sampling resolution of 30 minutes duration, the feature will give 24 time sample windows, the first of which would be the first 30 minutes. The Sankey diagram represents the three dimensions of interval, endpoint ID, and flow count per sample. The endpoint ID is the endpoint that the EBAS is connected to. The flow count per sample is the number of packets counted in a particular flow. Figure 8 shows a Sankey diagram of the selected points on a feature time series. Again, this represents a prompt analysis of various endpoints together with the weight of counts as the width of each flow band. Finally, the probability distribution analysis is shown with the Freedman-Diaconis rule (Shenghua Liu, Bryan Hooi, & Christos Faloutsos, November 2017) to estimate the histogram bin width for wide-sense stationary data. This analysis will be helpful to the threat hunter in investigating the outlier feature counts and determining whether these outliers are noise, a result of some legitimate traffic, or real anomalies.

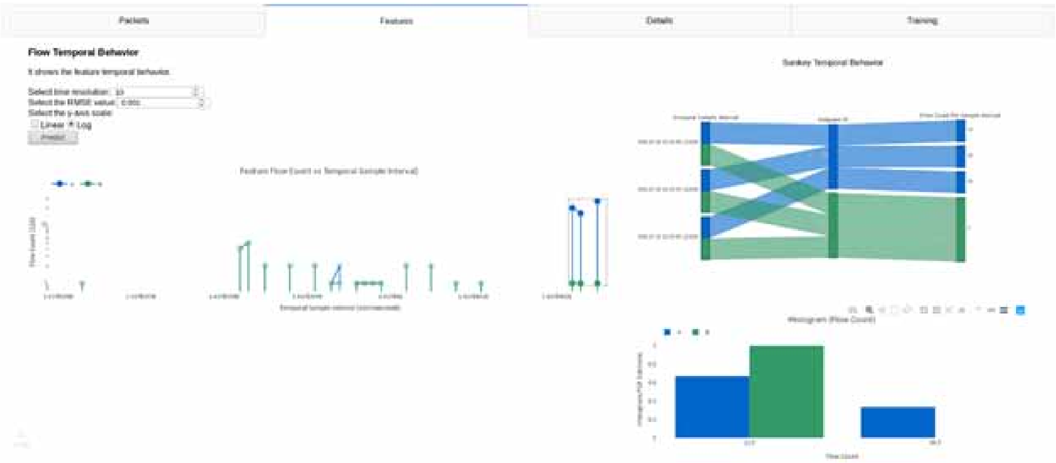
The “Training” tab is shown in Figure 9, where a threat hunter can select learning feature flow behavior using the SARIMA time series model for either an individual endpoint or all the combined endpoints for an aggregated correlation analysis. Once the model learns the behavior and extracts a correlation, it builds a trained model file which is subsequently used by SARIMA for prediction. To run a live prediction on each incoming packet from the streaming pipeline, the hunter should switch back to the Features tab.

Figure 10 shows a prediction of the anomalies using red-colored points on the time series. This prediction is performed using the user-configured RMSE value, and hence it provides the threat hunter the required power to adjust the behavior similarity that the predicted and the actual traffic should tolerate before the prediction model calls a feature point an anomaly.

As each individual flow is an extracted summary of the packet flow counts in either direction, a “Details” tab is provided as shown in Figure 11. The tab displays the details of individual raw packets for the selected point in the Features tab. For example, 15 raw packets are shown for the selected 6 feature samples in Figure 11. The last column of Flow ID represents the unique UUID for the respective sample in the database.

To recap, we propose a new framework that allows the correlation analysis of unique fundamental features that give the threat hunter the capability to perform four main activities: (1) navigate the network traffic, (2) engineer features analysis and extraction, (3) explore machine learning capabilities to train, predict, and correlate time series events in near real-time, and (4) deep-dive into anomalous behavior analytics to extract details and add more intelligence to the cognitive threat hunting model.

Figure 8. Online Feature(s) Extraction Time Series Sankey diagram

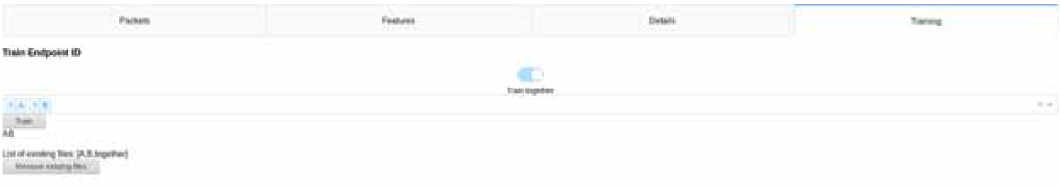


Flow Temporal Behavior

It shows the feature temporal behavior.

Select time resolution: 10
Select the RMSE value: 0.001
Select the y-axis scale:
□ Linear ● Log

Figure 9. Training tab – learning the behaviour feature flow

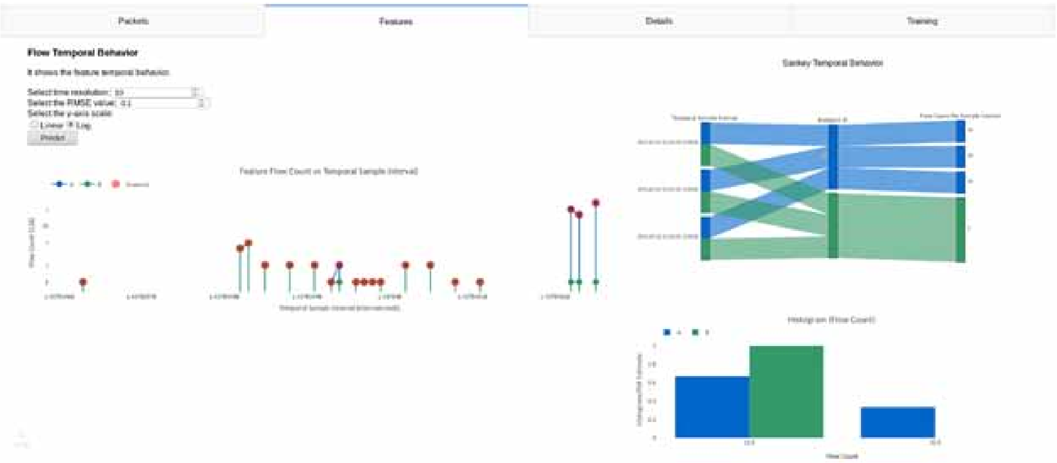


Train Endpoint ID

AB

List of existing files: [A,B,together]

Figure 10. Features tab – prediction of anomalies

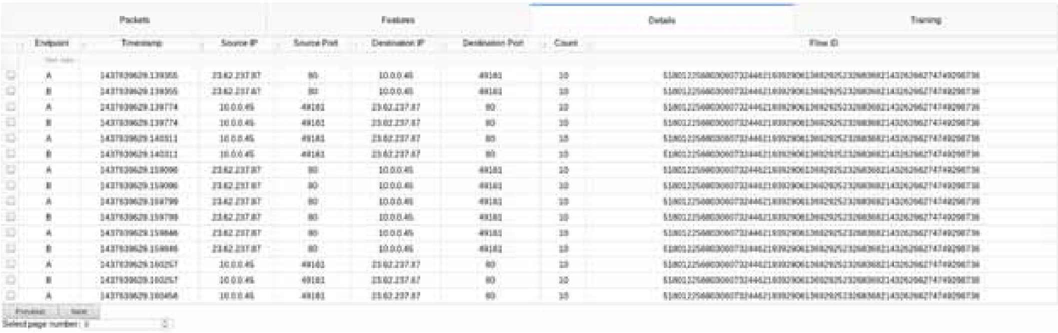


Flow Temporal Behavior

It shows the feature temporal behavior

Select time resolution: 10
Select the RMSE value: 0.1
Select the y-axis scale
○ Linear ● Log

Figure 11. Details tab – packet details of selected packet flow feature



	Endpoint	Timestamp	Source IP	Source Port	Destination IP	Destination Port	Count	Flow ID
☐	A	1437939629.139355	23.62.237.87	80	10.0.0.45	49181	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	B	1437939629.139355	23.62.237.87	80	10.0.0.45	49181	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	A	1437939629.139774	10.0.0.45	49181	23.62.237.87	80	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	B	1437939629.139774	10.0.0.45	49181	23.62.237.87	80	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	A	1437939629.140311	10.0.0.45	49181	23.62.237.87	80	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	B	1437939629.140311	10.0.0.45	49181	23.62.237.87	80	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	A	1437939629.159096	23.62.237.87	80	10.0.0.45	49181	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	B	1437939629.159096	23.62.237.87	80	10.0.0.45	49181	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	A	1437939629.159799	23.62.237.87	80	10.0.0.45	49181	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	B	1437939629.159799	23.62.237.87	80	10.0.0.45	49181	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	A	1437939629.159846	23.62.237.87	80	10.0.0.45	49181	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	B	1437939629.159846	23.62.237.87	80	10.0.0.45	49181	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	A	1437939629.160257	10.0.0.45	49181	23.62.237.87	80	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	B	1437939629.160257	10.0.0.45	49181	23.62.237.87	80	10	5180122568030607324462183929061369292523268368214326296274749298736
☐	A	1437939629.160458	10.0.0.45	49181	23.62.237.87	80	10	5180122568030607324462183929061369292523268368214326296274749298736

Select page number: 0

12. CONCLUSION AND FUTURE DIRECTIONS

This paper presents a cognitive cyber threat-hunting methodology using SARIMA-based time series modeling, which applies a correlation simulating the functions of a human mind during cyber threat hunting. In this paper, four concepts are presented to reinforce the concept of threat hunting using machine intelligence to relieve the threat hunter of cognitive overload: (1) a proof-of-value for proactive threat hunting not based on offline labeled training and using flow feature engineering, (2) real-time streaming, (3) a prediction-based system (SARIMA) rather than a heuristic-based training system, and (4) a new visualization framework that illustrates the proactive strategy using correlation time series to train and predict anomalous behavior in a near real-time temporal window. This paper presents a new idea for aiding threat-hunting tasks by using cognitive machine intelligence. Furthermore, in this work, the time series uses an equal time interval sampling window with valid statistical stationarity concepts that can be relaxed for more real-world data using adaptive and overlapping time windows (Muhammad Salman Khan, Ken Ferens, & Witold Kinsner, July 2015). This work is at preliminary proof of value stage and requires evolution into a more autonomous threat-hunting framework. Future directions include, but are not limited to, adding an autonomous feature-extraction mechanism using machine learning.

REFERENCES

Aafer, Y., Zhang, N., Zhang, Z., Zhang, X., Chen, K., Wang, X. F., & Grace, M. et al. (2015). Hare Hunting in the Wild Android: A Study on the Threat of Hanging Attribute References. In *Proc. of the ACM 22nd SIGSAC Conference on Computer and Communications Security* (pp. 1248-1259). ACM. doi:10.1145/2810103.2813648

Ableegoldman. (n.d.). *Kafka Streams Examples*. Retrieved June 10, 2019, from https://github.com/confluentinc/kafka-streams-examples

Ahamed, R., Habeeb, A., Nasaruddin, F., Gani, A., Abaker, I., Hashem, T., & Imran, M. et al. (2019, April). Real-time big data processing for anomaly detection: A Survey. *International Journal of Information Management*, *45*, 289–307. doi:10.1016/j.ijinfomgt.2018.08.006

Amalina, Hashem, Azizul, Fong, Firdaus, Imran, & Anuar. (2019). Blending Big Data Analytics: Review on Challenges and a Recent Study. *IEEE Access*. 10.1109/ACCESS.2019.2923270

Anton, S. D., Ahrens, L., Fraunholz, D., & Schotten, H. D. (2018). Time is of the Essence: Machine Learning-Based Intrusion Detection in Industrial Time Series Data. In *Proc. of IEEE 2018 International Conference on Data Mining Workshops*. Singapore: IEEE. doi:10.1109/ICDMW.2018.00008

Apache Software Foundation. (n.d.a). *Apache Kafka - A Distributed Streaming Platform*. Retrieved June 17, 2019, from https://kafka.apache.org/

Apache Software Foundation. (n.d.b). *Apache ZooKeeper*. Retrieved June 17, 2019, from https://zookeeper.apache.org/

Arlt, J., & Trcka, P. (2019, June). Automatic SARIMA modeling and forecast accuracy. *Communications in Statistics. Simulation and Computation*, 1–22. Advance online publication. doi:10.1080/03610918.2019.1618471

Bandos, T. (2019). *The Five Steps of Incident Response*. Retrieved July 19, 2019, from Digital Guardian: https://digitalguardian.com/blog/five-steps-incident-response

Boehm, B., & Howard, B. (2017, October 3). *Protocol Numbers. Internet Assigned Numbers Authority (IANA)*. Retrieved July 19, 2019, from https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

Bookshire, N. (n.d.). *Bubble Dock "Virus" Removal (What is Bubble Dock?)*. Retrieved July 23, 2019, from HowToRemove.Guide: https://howtoremove.guide/bubble-dock-virus-removal-what-is-bubble-dock

Chebbi. (2018). *Mastering Machine Learning for Penetration Testing*. Packt.

Chen, Z., Xu, G., Mahalingam, V., Ge, L., Nguyen, J., Yu, W., & Lu, C. (2016, April). A Cloud Computing Based Network Monitoring and Threat Detection System for Critical Infrastructures. *Big Data Research*, *3*(C), 10–23. doi:10.1016/j.bdr.2015.11.002

Chen, Z., Zhang, H., Hatcher, W. G., Nguyen, J., & Yu, W. (2016). A streaming-based network monitoring and threat detection system. In *Proc. of IEEE 2016 14th International Conference on Software Engineering Research, Management and Applications.* Towson, MD: IEEE. doi:10.1109/SERA.2016.7516125

Cherenson, A., & Singhal, A. (2019, May 15). IEEE 802 Numbers. In *IANA Protocol Registries*. Piscataway, NJ, USA: Internet Assigned Numbers Authority (IANA). Retrieved July 19, 2019, from https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml

Cole, E. (2015). Automating the Hunt for Hidden Threats. *SANS Intrusion Detection*. Retrieved July 8, 2019, from https://www.sans.org/reading-room/whitepapers/detection/paper/36282

Cole, E. (2019). Threat Hunting: Open Season on the Adversary. *Best Practices*. Retrieved July 19, 2019, from https://www.sans.org/reading-room/whitepapers/bestprac/paper/36882

Community, S. (n.d.). *Scapy - Packet crafting for Python2 and Python3*. Retrieved May 1, 2019, from https://scapy.net/

Czech Technical University. (2015, July 28). CTU-Mixed-Capture-1. *Mixed Capture Datasets - Malware Capture Facility Project*. Retrieved July 22, 2019, from https://mcfp.felk.cvut.cz/publicDatasets/CTU-Mixed-Capture-1/

Danjou, J. (n.d.). *Python + Memcached: Efficient Caching in Distributed Applications*. Retrieved June 21, 2019, from https://realpython.com/python-memcache-efficient-caching/

David, O. E., & Netanyahu, N. S. (2015). DeepSign: Deep learning for automatic malware signature generation and classification. In *Proc. of 2015 IEEE International Joint Conference on Neural Networks*. Killarney, Ireland: IEEE. doi:10.1109/IJCNN.2015.7280815

Demertzis, K., & Iliadis, L. (2015). Evolving Smart URL Filter in a Zone-Based Policy Firewall for Detecting Algorithmically Generated Malicious Domains. In Statistical Learning and Data Sciences (pp. 223-233). Egham, UK: Springer International Publishing. doi:10.1007/978-3-319-17091-6_17

Dormando. (2016). *memcached*. Retrieved June 19, 2019, from https://github.com/memcached/memcached/wiki/Install

Farwell & Rohozinski. (2012). The New Reality of Cyber War. *Survival - Global Politics and Strategy, 54*(4), 107-120. 10.1080/00396338.2012.709391

Franklin, L., Pirrung, M., Blaha, L., Dowling, M., & Feng, M. (2017). Toward a Visualization-Supported Workflow for Cyber Alert Management Using Threat Models and Human-Centered Design. In *Proc. of IEEE 2017 Symposium on Visualization for Cyber Security*. Phoenix, AZ: IEEE. doi:10.1109/VIZSEC.2017.8062200

Garcia, S. (2014). *Identifying, Modeling and Detecting Botnet Behaviors in the Network* (Ph.D. Dissertation). Prague: Czech Technical University.

Garcia, S., Grill, M., Stiborek, J., & Zunino, A. (2014). An empirical comparison of botnet detection methods. *Computer Security Journal*, *45*, 100–123. doi:10.1016/j.cose.2014.05.011

Hatmaker, T. (2017). *Four years later, Yahoo still doesn't know how 3 billion accounts were hacked*. Retrieved July 24, 2019, from https://techcrunch.com/2017/11/08/yahoo-senate-commerce-hearing-russia-3-billion-hack/

Junqueira & Reed. (2013). *ZooKeeper: Distributed Process Coordination.* O'Reilly Media.

Junqueira, F. P., & Reed, B. C. (2009). The life and times of a zookeeper. In *Proc. of the twenty-first annual symposium on Parallelism in algorithms and architectures*. ACM. doi:10.1145/1583991.1584007

Kempe, S. (2013). *Enterprise Threats: Big Data and Cyber Security*. Retrieved July 24, 2019, from https://www.dataversity.net/enterprise-threats-big-data-and-cyber-security/

Khan, M. S. (2018). *Malvidence - A Cognitive Malware Characterization Framework* (Ph.D. Dissertation). Winnipeg, Manitoba, Canada: Faculty of Graduate Studies, University of Manitoba, Canada.

Khan, M. S. (2019). *Machine Learning and Cognitive Science Applications in Cyber Security* (M. S. Khan, Ed.). IGI Global. doi:10.4018/978-1-5225-8100-0

Khan, M. S., Ferens, K., & Kinsner, W. (2015). Multifractal Singularity Spectrum for Cognitive Cyber Defence in Internet Time Series. *International Journal of Software Science and Computational Intelligence*, *7*(3), 17–45. doi:10.4018/IJSSCI.2015070102

Khan, M. S., Ferens, K., & Kinsner, W. (2015). A cognitive multifractal approach to characterize complexity of non-stationary and malicious DNS data traffic using adaptive sliding window. In *Proc. of IEEE 14th Intl. Conf. Cognitive Informatics & Cognitive Computing.* Beijing, China: IEEE. doi:10.1109/ICCI-CC.2015.7259368

Khan, M. S., Ferens, K., & Kinsner, W. (2015). A Polyscale Autonomous Sliding Window for Cognitive Machine Classification of Malicious Internet Traffic. In *Proc. of the International Conference on Security and Management*. Las Vegas, NV: The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Khan, M. S., Siddiqui, S., & Ferens, K. (2017). Cognitive modeling of polymorphic malware using fractal based semantic characterization. In *Proc. of IEEE 2017 International Symposium on Technologies for Homeland Security*. Waltham, MA: IEEE. doi:10.1109/THS.2017.7943487

Kumar, M. (2018, Jun. 11). *A New Paradigm For Cyber Threat Hunting*. Retrieved Aug. 8, 2018, from https://thehackernews.com/2018/06/cyber-threat-hunting.html

Kumar, M. (2018). *A New Paradigm For Cyber Threat Hunting*. Retrieved July 22, 2019, from https://thehackernews.com/2018/06/cyber-threat-hunting.html

Lee, R. M., & Bianco, D. (2019). *Generating Hypotheses for Successful Threat Hunting*. Retrieved from Threats/Vulnerabilities: https://www.sans.org/reading-room/whitepapers/threats/paper/37172

Liliengren, T., & Lowenadler, P. (2018). *Threat hunting, definition and framework* (Bachelor Thesis). Halmstad, Sweden: School of Information Technology, Halmstad University, Sweden. Retrieved from http://www.diva-portal.org/smash/get/diva2:1205812/FULLTEXT02.pdf

Liu, S., Hooi, B., & Faloutsos, C. (2017). HoloScope: Topology-and-Spike Aware Fraud Detection. In *Proc. of the ACM 2017 Conference on Information and Knowledge Management* (pp. 1539-1548). Singapore: ACM. doi:10.1145/3132847.3133018

Naik, N., Jenkins, P., Savage, N., & Yang, L. (2019). *Cyberthreat Hunting - Part 1: Triaging Ransomware using Fuzzy Hashing, Import Hashing and YARA Rules*. Retrieved July 26, 2019, from Northumbria Research Link: http://nrl.northumbria.ac.uk/id/eprint/38877

O'Flaherty, K. (2018). *Breaking Down Five 2018 Breaches -- And What They Mean For Security In 2019*. Retrieved 07 19, 2019, from Forbes: https://www.forbes.com/sites/kateoflahertyuk/2018/12/19/breaking-down-five-2018-breaches-and-what-they-mean-for-security-in-2019/#50e4911541c4

Obremski, T. (2016). *Is Full Packet Capture Worth the Investment?* Retrieved July 19, 2019, from https://securityintelligence.com/is-full-packet-capture-worth-the-investment/

Pegna. (2015). *Cybersecurity and machine learning: The right features can lead to success - The key-aspects of building successful cybersecurity machine-learning models*. Computer World - The Voice of Business Technology.

Peters, K. M. (2019). *21st Century Crime: How Malicious Artificial Intelligence will Impact Homeland Security.* Naval Postgraduate School Monterey California. Retrieved July 18, 2019, from https://apps.dtic.mil/dtic/tr/fulltext/u2/1073657.pdf

Pilici, S. (2014, January 29). *Remove Bubble Dock pop-up ads (Virus Removal Guide)*. Retrieved July 23, 2019, from https://malwaretips.com/blogs/bubble-dock-virus-removal/

Posta, C. (n.d.). What is Apache Kafka? Why is it so popular? Should you use it? *TechBeacon*. Retrieved July 24, 2019, from https://techbeacon.com/app-dev-testing/what-apache-kafka-why-it-so-popular-should-you-use-it

Schmitt, S. (2018). *Advanced threat hunting over software-defined networks in smart cities* (Masters Thesis). Chattanooga, TN: University of Tennessee at Chattanooga. Retrieved from https://scholar.utc.edu/theses/576

Siddiqui, S. (2017). *Cognitive Artificial Intelligence – A Complexity Based Machine Learning Approach for Advanced Cyber Threats* (M.Sc. Thesis). Winnipeg, Manitoba, Canada: Faculty of Graduate Studies, University of Manitoba, Canada.

Siddiqui, S., Khan, M. S., Ferens, K., & Kinsner, W. (2016). Detecting Advanced Persistent Threats using Fractal Dimension based Machine Learning Classification. In *Proc. of the 2016 ACM on International Workshop on Security And Privacy Analytics* (pp. 64-69). New Orleans, LA: ACM. doi:10.1145/2875475.2875484

Siddiqui, S., Khan, M. S., Ferens, K., & Kinsner, W. (2017). Fractal based cognitive neural network to detect obfuscated and indistinguishable internet threats. In *Proc. of 2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing.* Oxford, UK: IEEE. doi:10.1109/ICCI-CC.2017.8109765

Siddiqui, S., Khan, M. S., Ferens, K., & Kinsner, W. (2017). Fractal Based Cognitive Neural Network to Detect Obfuscated and Indistinguishable Internet Threats. In *Proc. of IEEE 2017 16th International Conference on Cognitive Informatics & Cognitive Computing.* Oxford, UK: IEEE. doi:10.1109/ICCI-CC.2017.8109765

SolarWinds. (n.d.). *What is Netflow?* SolarWinds Worldwide, LLC. Retrieved July 24, 2019, from https://www.solarwinds.com/netflow-traffic-analyzer/use-cases/what-is-netflow

SQLite Consortium. (n.d.). *What Is SQLite?* Retrieved June 21, 2019, from https://www.sqlite.org/index.html

Srinivas, B. V. (n.d.). *Security Operations Centre (SOC) in a Utility Organization*. Retrieved June 12, 2018, from https://www.sans.org/reading-room/whitepapers/ICS/security-operations-centre-soc-utility-organization-35502

Strasak, F. (2017). *Detection of HTTPS Malware Traffic* (Bachelor's Thesis). Faculty of Electrical Engineering, Czech Technical University.

Terzi, D. S., Terzi, R., & Sagiroglu, S. (2017). Big data analytics for network anomaly detection from netflow data. In *Proc. of IEEE 2017 International Conference on Computer Science and Engineering*. Antalya, Turkey: IEEE. doi:10.1109/UBMK.2017.8093473

Toasa, R., Aldas, C., Recalde, P., & Coral, R. (2019). Performance Evaluation of Apache Zookeeper Services in Distributed Systems. In *Information Technology and Systems - International Conference on Information Technology & Systems - Advances in Intelligent Systems and Computing* (Vol. 918, pp. 356-364). Springer Nature Switzerland AG. doi:10.1007/978-3-030-11890-7_35

Tutorials, A. P. (2018). *Implementing Leader Election using ZooKeeper*. Retrieved July 25, 2019, from https://www.allprogrammingtutorials.com/tutorials/leader-election-using-apache-zookeeper.php

Vidaurre, Smith, & Wool. (2017). Brain network dynamics are hierarchically organized in time. *Proc. of National Academy of Sciences of the United States of America, 114.* 10.1073/pnas.1705120114

Vijayan, J. (2016). *'Threat Hunting' On The Rise*. Retrieved July 19, 2019, from Dark Reading: https://w1.darkreading.com/endpoint/threat-hunting-on-the-rise/d/d-id/1325144

Wang, C., Rayan, I. A., & Schwan, K. (2012). Kafka: a Distributed Messaging System for Log Processing. In *Proc. of 2012 Middleware Conference - Posters and Demo Track.* Montreal, Quebec, Canada: ACM. doi:10.1145/2405153.2405157

Xiao, C., Zhang, S., Zeng, Q., & Cao, X. (2018). Real-Time and Distributed Anomalies Detection Architecture and Implementation with Structured Streaming. In *Recent Developments in Intelligent Computing, Communication and Devices - Advances in Intelligent Systems and Computing* (Vol. 752, pp. 973–980). Springer. doi:10.1007/978-981-10-8944-2_112

Zhao, S., Chandrashekar, M., Lee, Y., & Medhi, D. (2015). Real-time network anomaly detection system using machine learning. In *Proc. of IEEE 2015 11th International Conference on the Design of Reliable Communication Networks.* Kansas City, MO: IEEE. doi:10.1109/DRCN.2015.7149025

Zimmerman, C. (2014). *Ten Strategies of a World-Class Cybersecurity Operations Center*. The MITRE Corporation.

ZooKeeper. (2019, May 20). *ZooKeeper*. Retrieved July 15, 2019, from https://zookeeper.apache.org/doc/r3.5.5/zookeeperOver.html

Muhammad Salman Khan (PhD) is a Senior Cyber Scientist at the National Innovation Center for Cyber Security (NICCS) of the Canadian Nuclear Laboratories (CNL). Dr. Khan has leading expertise in the application of new machine learning algorithms in autonomous cyber threat hunting tools. Dr. Khan has 12+ years' experience in research, design, and development of cloud and cyber products. His doctoral dissertation resulted in a new cognitive threat-hunting framework using autonomous machine intelligence tools, and was supported by Mitacs Canada and Canadian Tire Corporation. Dr. Khan was a Fulbright Scholar for his Master's in Electrical Engineering from Rutgers University, USA. He also contributed firmware development of a new real-time nuclear medicine device at University of Manitoba, Winnipeg, Canada, which involved collaboration with and leading contributions from Lawrence Berkeley National Laboratory, USA. Dr. Khan has contributed 30 research publications and served as Principal Investigator of various national and international R&D funds in the cloud and cyber security domains. Dr. Khan has published a book about cyber security as Editor-in-Chief. He is a member of various technical program committees and paper reviewing committees, and is a contributing and advising member of the CyberNB-led Critical Infrastructure Security Operation Center (CI-SOC) initiative (a Canadian government led consortium of public and private cyber eco system) in Fredericton, New Brunswick, which aims to build a next- generation Canadian cyber security operation center for mission- critical infrastructure. Dr. Khan is a member of IEEE and has served as the secretary of the IEEE section at the University of Manitoba, Winnipeg, Canada.

René Richard is a Senior Programmer/Analyst with the Scientific Data Mining group at the Digital Technologies Research Center (DTRC) of the National Research Council (NRC) of Canada. René has worked in various application domains such as Cyber Security, Bioinformatics, Serious Gaming and Health Informatics. He has 20+ years of experience in software development and is adept at developing software architecture for big data and cloud products. In his past professional experiences, René has been a lead developer for a major Canadian retail chain. René has contributed to 5 research publications, a successful patent application, and DARPA-funded projects, and is currently pursuing his Master of Science in Engineering at the University of New Brunswick where he is specializing in data science and streaming analytics.

Heather Molyneaux (PhD) is a Research Council Officer/Analyst with the cyber security team at the Digital Technologies Research Center of the National Research Council (NRC) of Canada. Prior to joining the cyber security team, Heather has more than a decade of experience working with the Human Computer Interaction (HCI) team at the NRC. She has experience in dozens of NRC lead projects and collaborations, and is the author of more than 50 publications as well as numerous internal reports. Her research interests are in the field of human factors and human computer interaction in the creation of usable security, behavioral information security, user's security and privacy perceptions, authentication and access management, and ethics applied to education, healthcare, and critical infrastructure.

Danick Cote-Martel is a Cyber Security Specialist co-op student at the National Innovation Center for Cyber Security (NICCS) of the Canadian Nuclear Laboratories (CNL) situated in Fredericton, New Brunswick. Danick will receive his Bachelor's Degree in Computer Engineering in December 2019 and will pursue a Ph.D. in Computational Intelligence applications in cyber security. Prior to working at CNL, he has contributed to the safety and defense of Canadian critical infrastructure, and collaborated with partners of the global Forum of Incident Response and Security Teams (FIRST) as a Computer Security Incident Handler (CSIH) for the Canadian Cyber Incident Response Center (CCIRC - officially the Canadian Center for Cyber Security (CCCS) since October 2018). In addition, he delivered penetration testing audits for Bentley Systems and the Government of Canada. Currently, he is honing his skills in threat intelligence, finance, and cognitive linguistics. Danick is a zealous Python programmer and regularly applies his coding skills to developing new cyber hunting tools.

Henry Jackson Kamalanathan Elango is a Cyber Security co-op student at the Digital Technologies Research Center of the National Research Council (NRC) of Canada. He is currently pursuing a Master's in Computer Science with a specific interest toward in cyber security at the University of New Brunswick. Having a keen interest in network-related threats, he developed an intrusion detection system to detect slowloris attacks, and has expertise with various penetration testing tools. As future work, he plans to research advanced threat intelligence using an interdisciplinary approach forensic cyber- psychology as principal theme.

Steve Livingstone (PhD) is the Manager of the Applied Physics Branch at Canadian Nuclear Laboratories (CNL), and has extensive experience in Science and Technology in the nuclear sector. He leads the Applied Physics Branch in supporting a variety of engineering and science activities, mainly in support of CNL's Safety and Security Program, and in particular includes CNL's National Innovation Centre for Cyber Security (NICCS) in Fredericton. Over the past few years Dr. Livingstone has been supporting and driving the growth and expansion of the NICCS team and capabilities in Fredericton.

Manon Gaudet is Team Lead of Cyber Security Research for the NRC-DTRC. She is also responsible for the DTRC Montreal Collaboration Center, where she will lead collaborative Cyber Security projects with academia and superclusters such as Scale.ai. She is a member of the Secretariat Study Group on Cyber Security for the International Civil Aviation Industry (ICAO), and also works on strategic dossiers with Transport Canada, (Innovation, Science and Economic Development Canada (ISDE), and other agencies. For NRC-IRAP, Manon chaired the Go-To-eXpert cyber security team and was part of the Information and Communications Technologies (ICT) Sector Team for many years. Manon is a trusted advisor to long-term NRC clients and other government agencies. She combines extensive experience in research and development in ITC and security, on complex project settings in web technologies, IoT, ICS, and Global Positioning System (GPS) among others. Manon is a founding member of In-Sec-M, an industrial and academic Canadian innovation cyber security cluster. She leads a Cyber Security for Intelligent Transportation workshop, and she is working on aviation and intelligent infrastructures (smart cities and grids). Manon has worked for more than 12 years in R&D cyber security, and holds a Bachelor's of Science in Computer Science from the University of Montreal and a Cyber Investigation Certificate from Ecole Polytechnique. She holds professional certifications (CISSP, SANS, and GIAC-GCED) and is an Information Systems Audit and Control Association (ISACA) member.

Dave Trask, after obtaining his bachelor's degree in electrical engineering, joined the Canadian Nuclear Laboratories (CNL) as a Control Systems Designer. With more than 30 years of experience with process control systems for mission-critical applications, Dave is now the Principal Engineer for cyber security at CNL. He supports senior leadership strategy decisions related to R&D activities, and in this role founded CNL's National Innovation Center for Cyber Security (NICCS) that focuses on protecting industrial systems for the nuclear industry. Dave leads research on cyber security in the supply chain, asset qualification, cyber security operations centers, SCADA systems for operating remote Small Modular Reactors (SMRs), cyber security best practices, and the development of an anomaly detection system for ICS environments. Research outcomes inform government, regulators, standards, and utilities.