electronics

MDPI

Article

InSight2: A Modular Visual Analysis Platform for Network Situational Awareness in Large-Scale Networks

Hansaka Angel Dias Edirisinghe Kodituwakku [1,*], Alex Keller [2] and Jens Gregor [1]

[1] Department of Electrical Engineering and Computer Science, The University of Tennessee, 1520 Middle Dr, Knoxville, TN 37996, USA; jgregor@utk.edu

[2] School of Engineering, Stanford University, 450 Serra Mall, Stanford, CA 94305, USA; axkeller@stanford.edu

* Correspondence: angelk@utk.edu

Received: 16 September 2020; Accepted: 13 October 2020; Published: 21 October 2020

Abstract: The complexity and throughput of computer networks are rapidly increasing as a result of the proliferation of interconnected devices, data-driven applications, and remote working. Providing situational awareness for computer networks requires monitoring and analysis of network data to understand normal activity and identify abnormal activity. A scalable platform to process and visualize data in real time for large-scale networks enables security analysts and researchers to not only monitor and study network flow data but also experiment and develop novel analytics. In this paper, we introduce InSight2, an open-source platform for manipulating both streaming and archived network flow data in real time that aims to address the issues of existing solutions such as scalability, extendability, and flexibility. Case-studies are provided that demonstrate applications in monitoring network activity, identifying network attacks and compromised hosts and anomaly detection.

Keywords: visual analytics; cybersecurity awareness; incident response; anomaly detection

1. Introduction

One of the prominent issues security analysts and researchers face when analyzing network data, whether archived or real-time streaming flow data, is finding tools that can extract, enrich, index, filter, process, and visualize the large-scale network data. For exploratory visual analysis in threat hunting and forensic study, a tool that allows processing of network flows filtered by a complex pipeline is important to find threats and events for proper incident response and decision-making. Flow data enriched with Open Source Intelligence (OSINT) as well as proprietary information provide valuable information for the analysis. Intuitive visualizations can help the human analysts not only understand the typical behaviors but also detect anomalies and further investigate them.

Furthermore, when generating datasets to develop novel analytics researchers also have to implement the frameworks to manipulate flow data and generate visualizations since general purpose data analysis tools are not designed to connect to network sockets to read different formats of streaming flow data and process flows at gigabit speeds. Existing works focus on providing visualizations and analytics for specific networks and applications and may not be ideal for real-time visual flow data analysis for large-scale networks. By learning past behavioral patterns, future states of the network can be predicted such as bandwidth utilization patterns to detect anomalies. These functional requirements for situational awareness are critical for identifying incidents and threats, investigating anomalies and making decisions [1–4]. A flexible platform that can provide a framework for researchers to read and manipulate flow data and augment them with contextual information such as geolocation and known threat labels can improve and streamline analytics development.

1.1. Motivation

InSight2 is conceived as a platform specifically designed for flow data analysis that creates a synergy between researchers who develop analytics and analysts that make use of them. This led to the development of InSight2 with the goals to monitor networks in real time for cybersecurity awareness, visually analyze archived flow data for incident response, augment and extract a subset of flow data for the development of novel analytics such as anomaly detection and deploy said analytics back in the system for continued growth of its capabilities. These goals require a common efficient flow processing back-end and a software architecture that overcome issues with existing solutions.

Network flow data are used for the analysis in this work as most networks collect and store some form of flow data. A network flow summarizes the attributes of a communication between two nodes in a computer network, such as size of the data transferred during a given session, source and destination identifiers, protocol used, etc. Capturing and processing network flows instead of network packets has many advantages. Since flow records do not contain any payload data, they are dramatically smaller in size and offer better user privacy as well as efficient storage and faster processing. In most cases, network packet processing techniques, such as deep packet inspection, are becoming less effective due to the widespread use of end-to-end encryption and privacy concerns. Enriching flow data with attributes such as geographic location, Domain Name Service (DNS) hostname, domain name, known malicious status, Autonomous System (AS) number, etc. provides vital information for real-time monitoring and analysis. Enriched flow data can be filtered, sorted and aggregated to generate real-time interactive visualizations to provide comprehensive network visibility. Visualizations based on port, protocol, geographic location, and custom attributes allow for detection of anomalies more effectively [5,6]. It is essential for modern networks to have an operational tool to visualize this data in real time to obtain situational awareness.

While most cybersecurity incidents are caused by intentional malicious activities, some firewall misconfigurations, software bugs, human errors, etc. can degrade the security posture of a network and leave it vulnerable for attackers. Comprehensive data about the network usage down to each device including attributes, such as traffic characteristics, communication with known threats, and use of restricted ports and protocols, can uncover risk factors proactively. Researchers and network operators can gain a vast amount of knowledge through exploratory visual analytics on datasets such as penetration testing data and Capture the Flag (CTF) competition data.

Development of flow analytics requires access to comprehensive datasets. Publicly available benchmark datasets such as KDD-99 may not be suitable in certain cases [7]. Due to the lack of an open platform to collect, enrich, organize, and manipulate flow data to develop flow analytics, each research group has to develop their own code to perform these functions. An open platform that provides such capabilities while facilitating the implementation of the developed analytics as plugin modules and see their output using a unified visualization framework, has the potential to lower the barrier to entry as well streamline the development for researchers.

1.2. Related Work

In this section, we examine current state-of-the-art of network situational awareness in large-scale networks and identify their shortcomings and challenges when adapting them to analyze flow datasets, streaming flow data, and developing analytics. To reduce the amount of data to be processed for large-scale networks, preprocessing steps such as filtering and sampling have been suggested in the literature [8]. These lossy preprocessing steps may, however, eliminate important data. In fact, it has been shown that anomaly detection algorithms can degrade when data are sampled [9,10]. We have analyzed 14.6TB of network traces from the Global Ring Network for Advanced Applications Development (GLORIAD) project [11,12]. Operational from 2012 to 2015, GLORIAD was one of the largest Research and Education (R&E) networks of its time, connecting researchers and scientists at a global scale. We found the GLORIAD traffic to consist of many small flows less than 100 kB in size with a relatively small number of larger 'elephant flows' accounting for the majority of data transmitted.

Only by amalgamating the many small flows can a complete picture of the traffic be obtained. Even though accuracy with sampling may be adequate for operational measurements, detection of malicious activity such as slow port scan traffic may go undetected. We developed a novel multi-threaded flow processing back-end to process all flow data that scales well on modern CPU architectures to overcome this problem.

A number of purpose-built flow data tools have been developed for deployment on specific networks. The GLORIAD team based at the University of Tennessee had previously developed a tool called InSight for their internal operational measurement purposes. InSight sampled Argus flow data [13] and filtered out flows smaller than 100 kB. InSight2 was inspired by this work, and archived flow data from the GLORIAD R&E network was used for testing prior to deployment on live university networks. Other than the name and lineage, InSight2 does not have any relation with InSight as the code bases and capabilities are completely different. Details are provided in Section 2.

Network monitoring using In-band Network Telemetry (INT) is one of the recent developments in network monitoring [14]. However, it requires the maximum number of hops in the INT reports to be six or less and an INTCollector to be placed on every sink switch which takes the switch resources away from its core functions and may be infeasible in certain situations. Another network operations related tool, NetSage [15,16], focuses on visualizing the R&E network infrastructure of the National Science Foundation. NetSage filters out flows smaller than 100 MB. The current implementation consists of a web portal for visualizations of the traffic measurements. Internet2 provides the Deepfield Analytics Service (DAS) [17] for visualization and analysis of cloud and network data and Network Diagnostic Tool [18] for network diagnostics. DAS is only available to Internet2 members. ESNet that is part of the U.S. Department of Energy has a visualization tool for displaying network bandwidth utilization [19]. SiLK [20] is a collection of Unix command-line tools for querying and analyzing converted NetFlow records. It requires the conversion of flow data to SiLK data-structure and additional software are needed for their visualization such as Analysis Pipieline [21]. NVisionIP [22], VisFlowConnect [23], and NfSen [24] provide visualization capabilities for NetFlow data generated by Cisco routers, but none of the projects appear to be active. Tstat visualizes traffic patterns at the network and transport levels [25] and is used by NetSage. However, Tstat also appears to be inactive. Commercial products are considered outside of the scope of our work.

1.3. Outline

The paper is organized as follows. Section 2 describes the software architecture from the modular scalable processing back-end and the visualization front-end to the deployment mechanism. We also explore the design decisions that address the issues with existing systems for network situational awareness. Section 3 presents case-studies relating to real-time situational awareness, incident response and anomaly detection. Section 4 provides the conclusions.

2. System Architecture

The core of InSight2 is a novel and flexible multi-threaded system architecture. Multi-threaded software can increase the throughput by delegating tasks to separate processor cores making the software more scalable. It also employs an optimized data flow control mechanism, uses a redundancy based indexed schemaless distributed database and a search engine, and visualizations to present the information using a web-based front-end. Compared to relational databases, schemaless indexed databases can significantly reduce the time needed to process queries (to the point of supporting real-time). They also allow custom attributes to be added without having to recreate the data tables. These functionalities are required for the generation of in-depth visualizations and running analysis tasks on complex filter pipelines.

InSight2 incorporates up-to-date security features to protect from unauthorized access such as server-side authentication and encryption using Transport Layer Security (TLS). Furthermore, its platform nature allows researchers to collect, enrich, organize, and manipulate flow data from

real and virtual environments, such as Software Defined Networks (SDN), to aid flow analysis research. The modular architecture allows such analytics to be implemented within InSight2 as modules extending its functionality. This modular nature also allows the sharing of these modules for extendability.

We studied the proof-of-concept InSight tool in-depth developed by GLORIAD at the University of Tennessee. It was designed to be an internal tool for the visualization of archived Argus flow data. The code base consisted of Perl 5 scripts that were invoked as Unix cronjobs at set intervals. Each script would carry out a specific task such as reading archived Argus data, extracting elephant flows, adding geolocation information, host attributes, etc. A ZeroMQ publisher-subscriber queue was used to share data between scripts. MySQL and SQLite databases were used for intermediate data storage of host attributes. During this study, we discovered issues with scaling, due to the use of single-threaded enrichment back-end and the use of SQL databases to hold contextual information and temporary flow records. It had issues with usability due to the lack of a proper installation mechanism and a run-time data-flow structure. It also had timing alignment issues due to being a collection of independent scripts that passed data between each other asynchronously. It was not extensible due to the lack of a core enrichment engine nor a mechanism to add modules to further process the enriched data to bring out more insights from the data.

InSight2 was inspired by InSight but has been built from the ground-up addressing the above issues. InSight2 consists of a core enrichment engine written in Python 3 along with peripheral modules that allow further management, processing, and analysis of the flow data. It uses an indexed schemaless database structure and does not depend on intermediate databases to hold temporary results. These design decisions enable better maintainability of the code-base as well as minimize the bottlenecks present in InSight architecture. The system runs in containerized environments, which simplifies the installation, streamlines updating, and eliminates dependency conflicts with the host operating system while allowing it to run on a wide range of hardware and virtualized environments, including standard commodity computers, Virtual Machines (VM) and SDNs. InSight2 uses the open-source Elasticsearch database, Kibana visualizations [26], and Docker containers but does not rely on any commercial software.

2.1. Overview

The flow data are processed through InSight2 as shown in Figure 1. Network packets are converted to network flows and are enriched using different information sources such as OSINT or proprietary information. They are stored in the database which has a search-engine functionality capable of on-the-fly data manipulation without storing an intermediate copy of the results. They are used to build the real-time visualizations and are further processed by the flow analysis plugins. Final results of the flow analysis plugins are added back to the database so that they can also be visualized in the same manner. Each component of the system function as modules as described below, providing data ingestion, database maintenance, and further analysis.

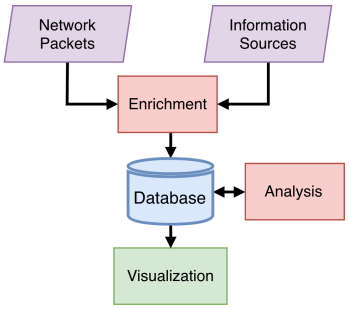

Figure 1. Data-flow overview.

Furthermore, the web-based Graphical User Interface (GUI) hosts the dashboards that allow for visual filtering of the displayed data in real-time. They react in unison according to the filters specified for the particular dashboard. Visual filtering includes limiting the scope of the displayed data based on the parameters such as time-range, IP/port range, geographic location, etc. Once the filters are defined the filtered data can be exported in CSV format to facilitate the use of third-party data analysis tools for further study. This facilitates extraction of critical enriched data as well as the outputs of the analysis modules. Below, we discuss the system modules in detail with reference to Figure 2, the system architecture.

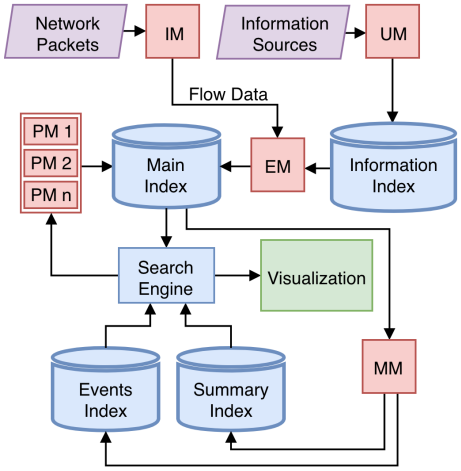

Figure 2. System architecture.

2.2. Input Module

The Input Module (IM) carries out flow data generation, storage, and presentation to the other modules. IM uses the Argus client 'rabins' to process six widely used flow standards in real time: NetFlow, sFlow, JFlow, IPFIX, Flow-tools, and Argus itself. IM also supports ingesting data from raw network streams such as network mirrors, which also are converted into Argus format. Furthermore, generated flow data can be archived for future enrichment and archived flow data can be re-enriched with new information such as updated threat-lists and host attributes at any time for forensic study.

Network packets are converted into bi-directional network flows. They are subjected to time-series binning which allows real-time stream block processing to synchronize the enrichment process. In this method, a network flow record is not only created when a FIN or RST flag is seen in the packet header but also at the end of each time bin. This method is lossless and retains all the information in every network flow. The flow data created at this step are enriched in-memory for efficiency and then are sent to the database for indexing. It facilitates the generation of real-time visualizations when a flow does not terminate at or before the time boundary. Even though this fragments a flow record at the boundary, they are re-aligned based on the timestamp for accurate flow reconstruction for analysis.

2.3. Enrichment Module

The Enrichment Module (EM) augments the raw flow data with contextual information and is crucial in visualizing relationships between end-points in the network. End-points consist of user-devices in the network and critical infrastructure such as network equipment and servers. These connections, and statistics about them, can be filtered to obtain detailed information, as explored in detail in Section 3, providing network administrators and security analysts the capability to evaluate the data from various perspectives for situational awareness. The contextual information can be intrinsic, such as the hostname, the institution associated with the end-point, physical location, type and the owner of the end-point, or extrinsic, such as whether the IP address has been involved with botnet activity, compromised hosts, distribution of malware, as well as the geographical location if it is a

public IP address. Intrinsic information is maintained by the network administrators while extrinsic information such as OSINT is kept up-to-date by different organizations that specialize in a particular task such as tracking botnet activity on the Internet.

EM is multi-threaded, so that it scales well with the processor speed as well as the core count to handle any incoming flow rate to deliver real-time visualizations. EM synchronizes with the bin duration provided by IM to capture its output at the time boundaries and the flow records are passed in memory. Once the records are extracted, it delegates the data into each core of the CPU by invoking process threads. Each thread augments the flow records with the contextual information and queues them for indexing into the database. When indexed, they are made available in the Main Index, which is used for generating visualizations as well as providing ground level data for further analysis by analytics modules.

Each database is divided into multiple shards and stored across multiple distributed servers. Shards are duplicated into backup shards providing data redundancy and faster access.

2.4. Updater Module

The Updater Module (UM) keeps the contextual information up-to-date by tracking changes to contextual data, both intrinsic and extrinsic, so that flow data enrichment is always carried out with the most up-to-date information. This allows on-demand enrichment of past flow data with new contextual information for forensic study.

2.5. Maintenance Module

Storage of enriched flow data is limited by the storage capacity of the host system. The Maintenance Module (MM) routinely prunes the Main Index by deleting older indexes when the system begins to run out of space. As a part of the pruning process, MM extracts and stores security incidents and flagged events in a separate index named Events Index. Enriched data are furthermore summarized to retain high level information about the performance of the network in the Summary Index. In the long term, MM contributes to retaining high level summarized information while retaining highly granular information in the short term.

2.6. Plug-in Modules

Plug-in Modules (PM) extend the core functionality of InSight2 provided by the aforementioned core modules. The Main Index, Summary Index, and the Events Index are all made accessible to the plug-in modules to enable flow analysis. Novel analytics are implemented and incorporated as modules, extending the capability of the system. Dashboards specific to the output of each plug-in visualize the results. When a plug-in module is created, a dashboard is created that reads data from the relevant index.

2.7. Front-End Functionality and Security

Figure 3 illustrates web-based GUI and security features of InSight2. The GUI consists of various tabs that are created per use case and each tab has one or more dashboards. A Dashboard consists of a set of modular visualizations and can be configured to show a particular aspect of the data. For example it can be, a bar chart of the traffic of top ten countries sorted in descending order along with a line chart showing total traffic variation over time. Information shown in the dashboard can be filtered visually from the GUI limiting the displayed data to a given scope—for example, filtering all Secure Shell (SSH) traffic from outside the network to a particular host in the network during the past hour. When this filter is applied to the previous scenario it will result in a bar chart showing the top incoming SSH traffic from outside the network to that particular host during the past hour sorted in descending order by the country as well as in the line plot spanning over the last hour. New dashboards can be created to group frequently used visualizations together or by a specialized purpose. Dashboards can be saved for later use as well as for reporting purposes. Data displayed in

the dashboard after being filtered according to the filter pipeline can be exported and saved to the disk in CSV format to create new datasets. Furthermore, when accessing data in time ranges past current retention, an option to re-enrich the data is provided.

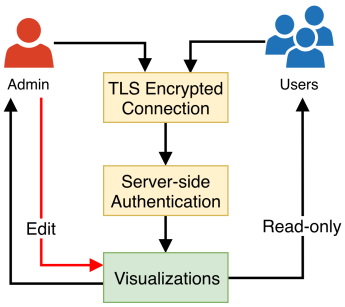

Figure 3. Web-interface security.

The dashboards are capable of detecting the user agent of the web-browser, either mobile or desktop, and serving the appropriate format of the dashboard that fits the available screen real-estate. The visualizations are created on-demand within the web-browser. The web-interface is secured with modern authentication and encryption standards to restrict access only to authorized users. It uses TLS (HTTPS) encryption to thwart man-in-the-middle (MITM) attacks. Based on the user type, relevant dashboards are loaded when each user logs in, controlling access to information. This also allows dashboards to be modified only by authorized users, making it possible to delegate monitoring and generate reports to less privileged users. It also facilitates collaborating with remote analyst teams for network forensics.

2.8. Deployment Mechanism

Virtualization allows InSight2 to be installed on a wide range of operating systems, hardware, and virtual environments such as SDN [27,28]. Using InSight2 with SDNs is not tested as of writing of this paper, but the Ingestion Module is designed to connect to either a supported network flow source such as Argus with the default port 561 over the network or convert network packets from a network mirror port to Argus format. Modular structure of the system provides better maintainability as well as deployability of the system. They can be installed in either hardware computers or VMs. InSight2 modules use application level abstraction that packages code and dependencies together, and shares the operating system (OS) kernel. InSight2 packages each of its modules as a separate Docker image and connects them using the Docker network. This is an abstract network that enables each image to be installed on either a single host or across multiple hosts as they communicate using network sockets. Figure 4 illustrates the connectivity between the containers.

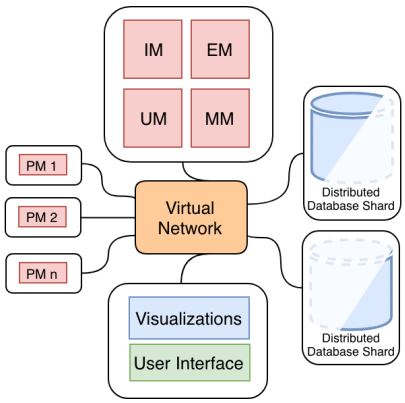

Figure 4. Deployment mechanism.

The database is distributed using shards which can be installed in separate containers. Duplicate shards aid in mitigating data loss in case of container failure or reachability issues. Furthermore, duplicate shards are searched in parallel during a search query to produce faster results.

3. Case Studies

In this section, we illustrate how we used InSight2 for situational awareness in three different scenarios: real-time situational awareness, incident response, and anomaly detection. Network packets are converted into Argus flow data which contain information extracted from the packet headers as well as measurements computed for the packets associated with each flow including number of bytes transmitted, start and end times, etc. The flow data are enriched using information available from other sources—for example, the mapping of IP addresses to geolocations using MaxMind GeoIP databases [29]. The enriched flow data are stored in a searchable indexed database. Plug-in modules add data analysis capabilities. Data are visualized using intuitive dashboards that combine different metrics by common needs such as measurement and security. InSight2 is currently deployed at Stanford University, Queen's University, Canada, and the University of Tennessee, Knoxville with plans underway for further deployments at additional universities and research institutes. The data for the case studies discussed here were obtained from the deployment at Stanford University. There, InSight2 is connected to a 10 Gbps mirror port in one of the School of Engineering (SoE) primary datacenters connecting over 200 physical servers and approximately 100 virtual machines, handling both administrative applications and research computation. Up-stream connectivity to the campus backbone is provided via 2×10 Gbps (Link Aggregation Control Protocol) LACP connection.

3.1. Real-Time Situational Awareness

Flow data are enriched in real time and the resulting visualizations are used for network situational awareness. Dashboards group a set of web-based visualizations that show information about the status of the network for current or past time periods. Each dashboard represents a particular aspect of the network traffic. Figure 5 illustrates the first page seen when the web-based GUI is loaded. Sensitive information such as IP addresses and country names are pixelated to preserve privacy. This dashboard gives a high-level view of the network traffic using gauges that display average network bandwidth utilization, packet loss, packet retransmissions, and producer-consumer ratio (PCR). Number of bytes transmitted as a function of time is shown along with four histograms displaying the top fifteen IP addresses and protocols for flows that originate from both source and destination. Tag clouds show the top ten countries sorted by the amount of traffic generated, while the segmented pie-chart provides organization name, project, and department information. Geographic heat-maps provide a geo-spatial breakdown of flows. Figure 5 shows a second dashboard that gives a more detailed view of the network traffic including connectivity between organizations, traffic heat-map, and composition by country, average number of hops, TCP handshake times, average packet sizes, etc. The connectivity map shows the top three organizations with the highest amount of traffic utilization and what other top organizations they communicated with. This is useful to identify the "top talkers" in the network. The sent and received traffic per country is plotted in a matrix on a time axis which allows for understanding traffic breakdown per country in the given time-frame. PCR is plotted in more detail on the time axis that visually shows the ingress and egress traffic ratio. TCP connection times, packet drops, and packet retransmissions indicate potential network congestion. Other visualizations include packet size and number of packets sent by country. Data enrichment capabilities and the ability to manipulate the data in real time from within the web interface enable interactive visualizations that do not require writing manual queries to the database.

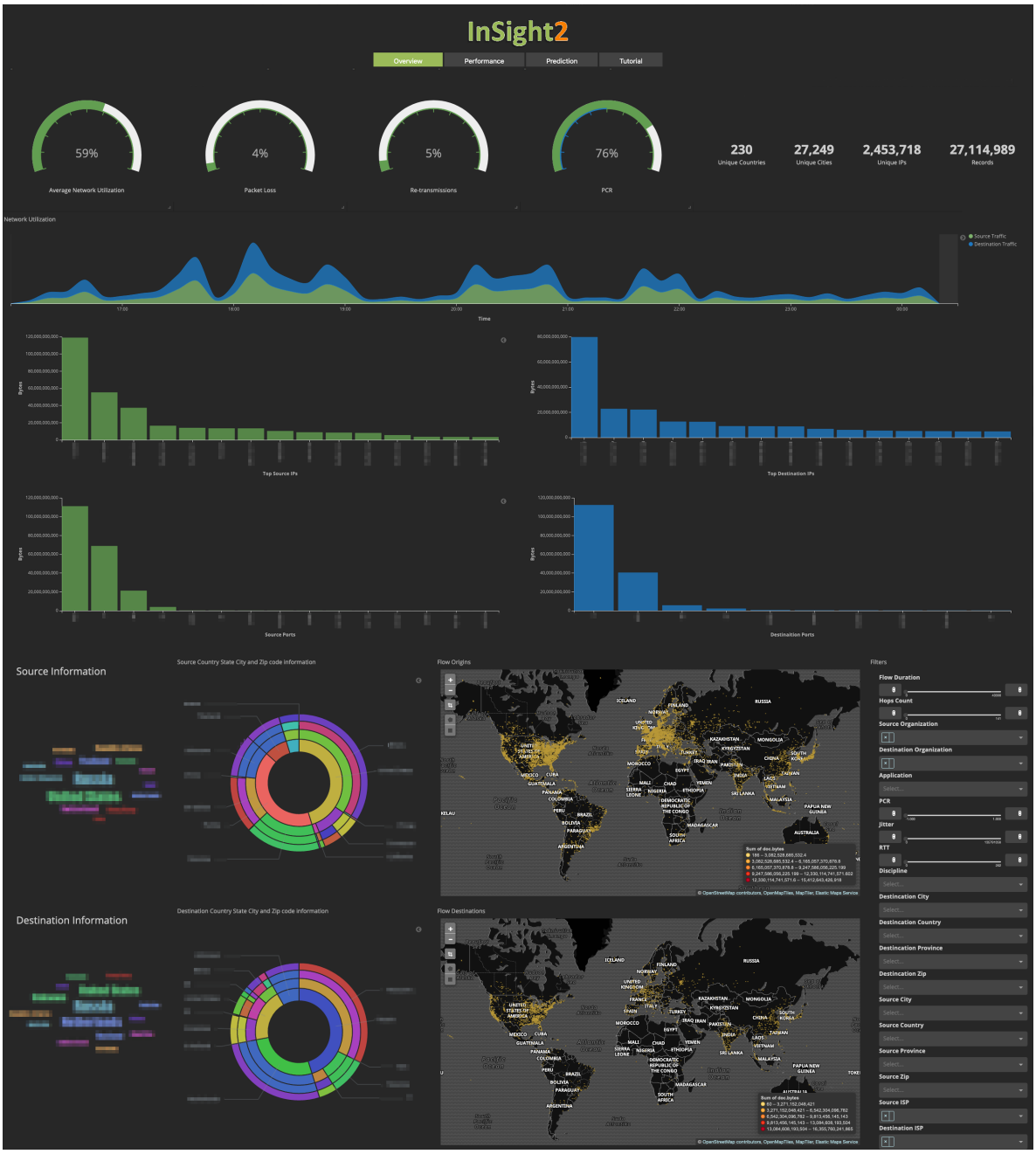

Figure 5. Network traffic overview dashboard.

Visualizations discussed in Figures 5 and 6 were built to understand the various aspects of the particular traffic under inspection. Filters can be applied to limit the scope within that data. For example, by applying outgoing port to port 80, the visualizations will adjust on-the-fly to show only the flows with HTTP requests that are exiting the network. This allows for understanding how the network handles web traffic as the flows are bi-directional. Furthermore, existing visualizations can be modified. For example, the country can be replaced with city for more granular visualizations. Finally, new dashboards can be created and saved to be re-used at any time.

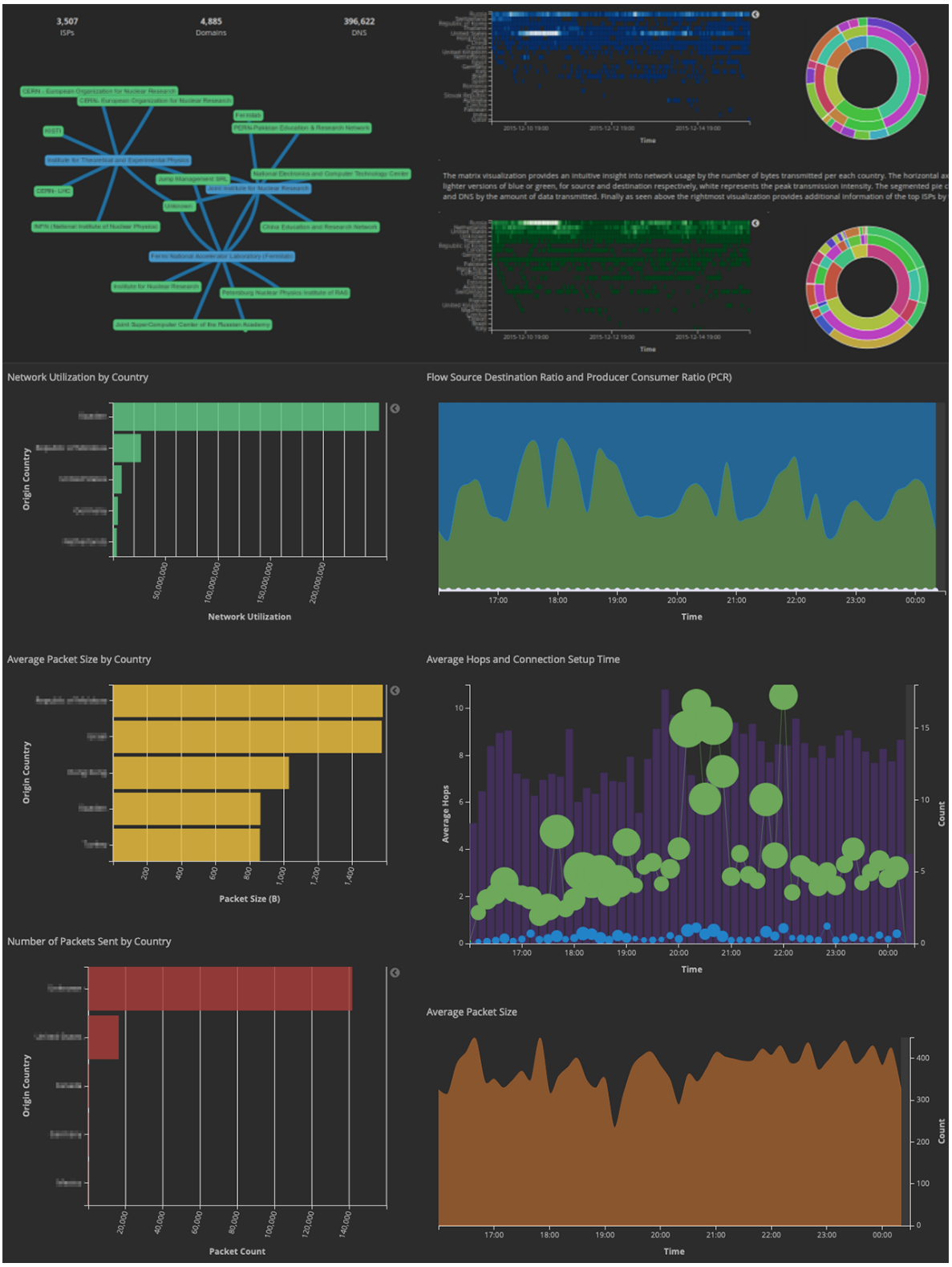

Figure 6. Network performance dashboard.

An important factor when exploring flow data is having the ability to filter and sort data according to one or more criteria, such as isolating all traffic to certain source or destination hosts or countries, ports or protocols, flows of given sizes, Internet Service Providers (ISPs) with the most packet losses, hosts that communicate with suspicious IP addresses, etc. This has been a core objective in the development of InSight2. For example, all visualizations within a dashboard can be focused on traffic

originating from the host currently generating the most traffic simply by selecting it via the dashboard. More sophisticated and granular filtering is possible via the advanced controls in the dashboard. This method allows for selecting traffic based on any one or more features associated with each flow including augmented features added during enrichment. For example, the network operator can create a filter that isolates all traffic to SSH traffic on port 22. In such cases, visualizations made for displaying source and destination hosts are sorted by the number of flows in descending order. Additionally, other sections of the dashboard will show where they are coming from geographically and topologically along with other attributes such as whether they have been previously tagged as malicious. A workflow such as this allows the network administrator to find potential threats that try to infiltrate critical systems in the network, and learn more information about them.

Furthermore, in conjunction with numerical and categorical filters, geographic areas can be easily isolated as well, either by clicking on countries in the tag cloud or by drawing a region of interest on the global map. To filter information based on an organization, project, or department of interest, the user simply clicks on the desired section on the segmented pie chart. The dashboards can be customized by modifying, adding, removing, relocating, and resizing visualizations to meet the needs of each deployment site.

3.2. Incident Response

We used flow data from Western Regional Collegiate Cyber Defense Competition 2019 (WRCCDC) [30] for visual analysis of the attacks and to find compromised hosts. Here, we discuss the process of knowledge inference that can be used in incident response and decision-making from the point of view of the participating Stanford University team. A simple forensic study workflow is followed in order to find compromised IPs in the network as shown in Figure 7.

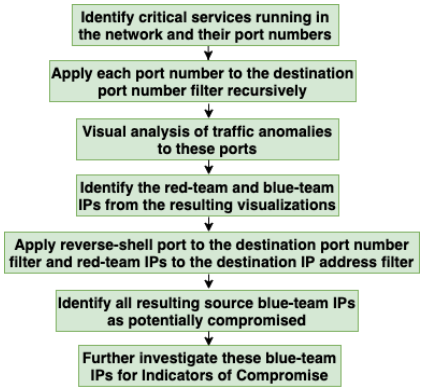

Figure 7. A simple forensic study workflow.

The WRCCDC competition consisted of eight blue teams whose objective was to defend network services such as web and email servers. The blue teams operated on subnets 10.47.x.0/24, where x denotes team numbers 1 through 8. A red team operating on a wide network range of 10.128.0.0/9 attacked the blue team servers using various techniques. A service check engine predominantly operating at 10.0.0.111 (and various other IPs) periodically checked the status of the blue team services to see if attacks had successfully disabled their critical network applications. Archived network traces are analyzed in this subsection using InSight2 to explore the attack sequence to demonstrate the network forensics capability of the platform.

Figure 8 shows the initial dashboard created for the analysis which contains the top IP addresses of both attackers and defenders before applying any filters. This visualization provides a starting point for the analysis when determining which services were attacked. We can discern some details of the attacks based on the destination port numbers and protocols. For example, port 3389 is used for RDP (Remote Desktop Protocol) connections in Windows systems, port 5900 is used by VNC (Virtual Network Computing), port 22 for SSH, port 80/443 for web services, etc. Analyzing the source and

destination IP addresses associated with these ports, we observed ports 137 (NetBIOS) and 445 (SMB) were under sustained attack by the red team. After a successful attack we see a connection initiated from the victim to port 4444 back to the attacker. Being the default port for the Metasploit [31] reverse shell, we can reasonably conclude the attacker has taken control of the target system and is engaged in post-exploitation techniques. This is shown in the Figure 9, where the reverse shell traffic is filtered to show the timeline, statistics of associated IP addresses and port numbers, as well as blue and red team IPs.

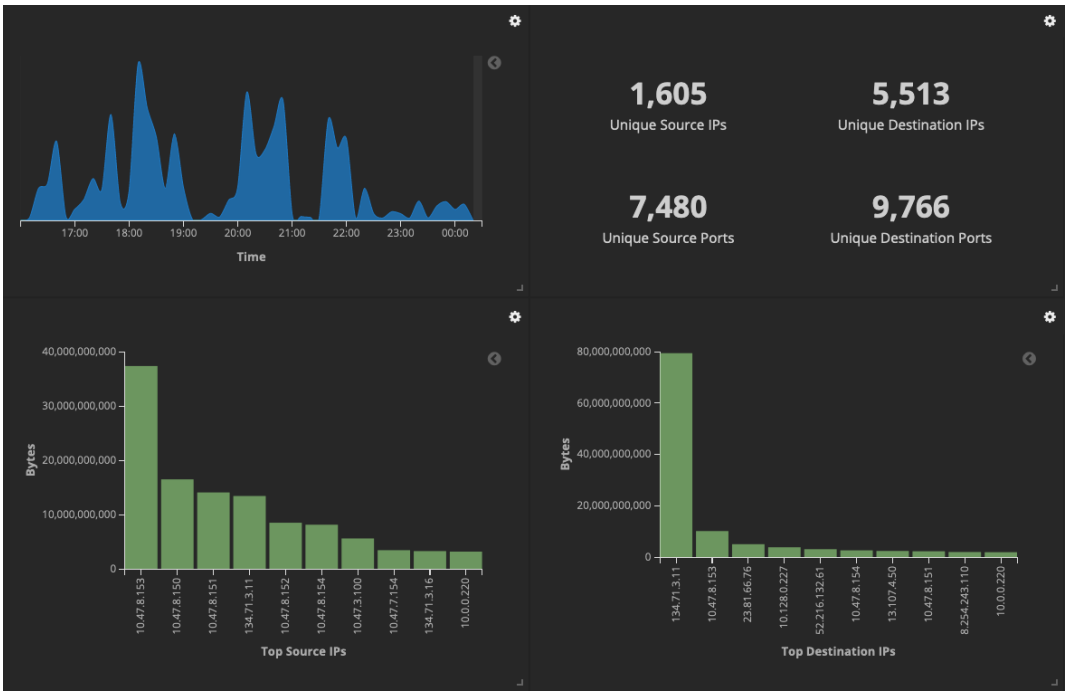

Figure 8. Attack statistics before filtering.

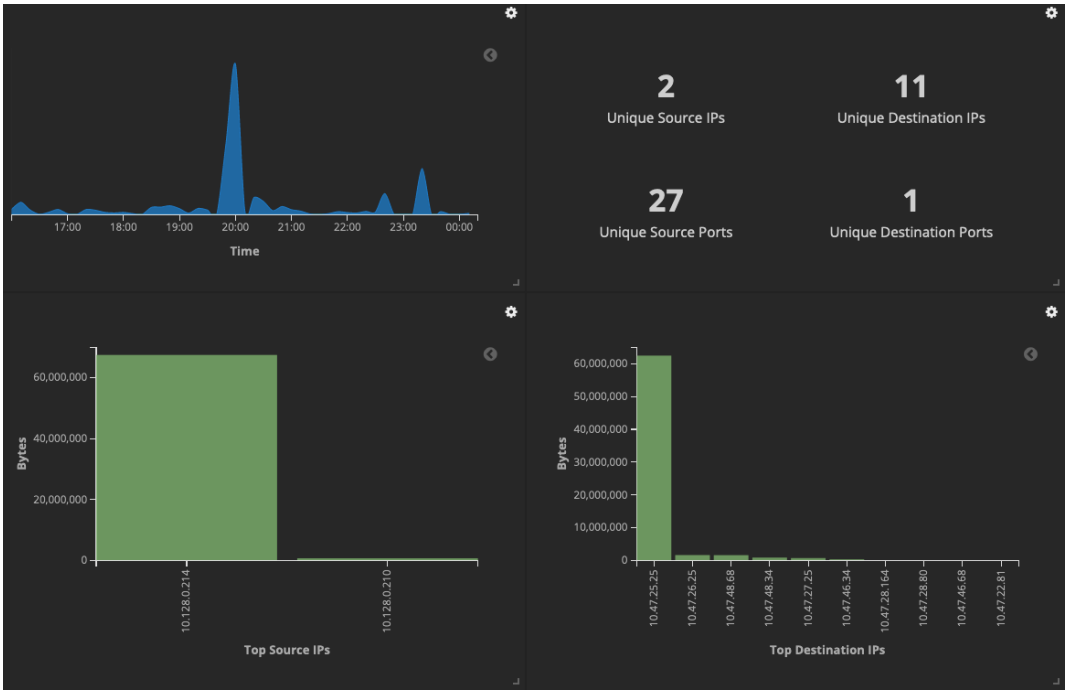

Figure 9. Attack statistics after filtering.

This case study illustrates how security analysts and incident response teams can find the end-points associated with network attacks, gauge what attacks are being launched, and isolate the command and control traffic to and from the compromised host(s).

3.3. Anomaly Detection

Plug-in modules that extend the functionality of the system reuse the data ingestion, enrichment, indexing and visualizing capabilities of InSight2 to do further analysis. With reference to Figure 2, a plug-in module reads data from the Main Index, the Events Index, and/or the Summary Index and writes its results to its own index. Dashboards tied with the analytics module visualize the output of that index.

Flow analysis is critical to identify anomalies automatically [32,33]. To illustrate the modular analytics capability of InSight2, we describe Markov chain prediction of network bandwidth utilization to detect anomalies. A Markov chain is a probabilistic finite-state machine for which future transitions depend only on the present state and not the states visited before reaching it. States here are defined by a simple three-level discretization of the network bandwidth utilization: low, medium, and high. The state transition matrix which makes up the Markov chain is inferred from observed transition frequencies when going from each state to every other state. The expected number of transitions needed before going from one state to another and the standard deviation associated therewith can be calculated from the transition matrix. Analyzing the bandwidth utilization data of the Stanford University for one week shows the daily mean time before high bandwidth usage is replaced by low bandwidth usage is about six hours with a standard deviation of about 1.3 h. Once the module was deployed on InSight2, we were able to detect a statistical anomaly where almost 16 h of high bandwidth usage was reported as shown in Figure 10. This simple module demonstrates an example of automating the analysis to detect anomalies.

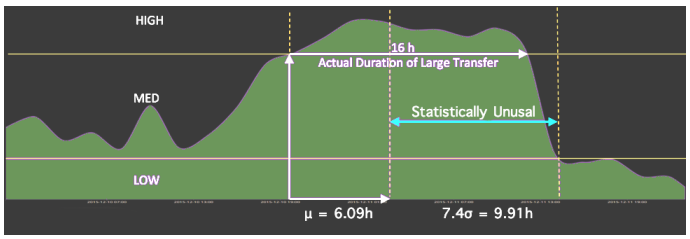

Figure 10. Network bandwidth utilization prediction.

Plug-in modules such as this can be developed and shared by the community of InSight2 users. They are language agnostic since they query, process, and insert data to the ES database without requiring any message passing between the core modules.

4. Conclusions

InSight2 is open-source software [34] that aims to fulfill the need for a platform for real-time situational awareness, incident response, and development and deployment of flow analytics in large-scale networks. We discuss the novel software architecture that addresses the issues with current solutions such as scalability, extendability, and flexibility. We discuss three case studies from one of its deployments at Stanford University on real-time situational awareness, anomaly detection and incident response using WRCCDC dataset. In these situations, we use real-time graphs to understand its behavior under normal conditions, infer knowledge from visual analysis for incident response that can be used to make decisions to improve the security, and detect abnormal behavior using automated anomaly detection.

The intuitive web-based front-end allows analysts and researchers to explore the data quickly, reducing the effort and time needed to monitor network traffic compared to querying databases followed by manual visualization of each result. A novel system architecture is developed for efficient

flow data enrichment leveraging modern multi-core CPUs. The modular architecture allows users to develop their own extensions to the platform. Finally, a container-based installation mechanism allows for ease of deployment and seamless maintenance of the software.

Author Contributions: Conceptualization, H.A.D.E.K.; Methodology, H.A.D.E.K., J.G.; Project administration, J.G.; Software architecture, H.A.D.E.K.; Supervision, J.G.; Validation, J.G., A.K.; Software implementation, H.A.D.E.K., Software deployment, H.A.D.E.K., A.K.; Writing—original draft, H.A.D.E.K.; Writing—review and editing, J.G., A.K.; All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work supported by the National Science Foundation under Grant No. IRNC-1450959.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Onwubiko, C. Functional requirements of situational awareness in computer network security. In Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI 2009), Richardson, TX, USA, 8–12 June 2009; pp. 209–213.
2. Husák, M.; Jirsík, T.; Yang, S.J. SoK: contemporary issues and challenges to enable cyber situational awareness for network security. In Proceedings of the 15th International Conference on Availability, Reliability and Security (ARES 2020), Dublin, Ireland, 25–28 August 2020; pp. 1–10.
3. Li, J.; Yi, X.; Wei, S. A Study of Network Security Situational Awareness in Internet of Things. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Byblos, Lebanon, 3–29 June 2020; pp. 1624–1629.
4. Gutzwiller, R.; Dykstra, J.; Payne, B. Gaps and Opportunities in Situational Awareness for Cybersecurity. In *Digital Threats: Research and Practice*; ACM: New York, NY, USA, 2020; pp. 1–6.
5. Zhang, T.; Liao, Q.; Shi, L. Bridging the gap of network management and anomaly detection through interactive visualization. In Proceedings of the 2014 IEEE Pacific Visualization Symposium, Yokohama, Japan, 4–7 March 2014; pp. 253–257.
6. Franke, U.; Brynielsson, J. Cyber situational awareness–a systematic review of the literature. *Comput. Secur.* **2014**, *46*, 18–31. [CrossRef]
7. Divekar, A.; Parekh, M.; Savla, V.; Mishra, R.; Shirole, M. Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives. In Proceedings of the 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), Kathmandu, Nepal, 25–28 October 2018; pp. 1–8.
8. Silva, J.M.; Carvalho, P.; Lima, S.R. A Modular Traffic Sampling Architecture: Bringing Versatility and Efficiency to Massive Traffic Analysis. *J. Netw. Syst. Manag.* **2017**, *25*, 643–648. Available online: https://www.overleaf.com/project/5f8fbb2378dfed00019bbb5b (accessed on 15 September 2020). [CrossRef]
9. Mai, J.; Chuah, C.N.; Sridharan, A.; Ye, T.; Zang, H. Is sampled data sufficient for anomaly detection? In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, Rio de Janeriro, Brazil, 25–27 October 2006; pp. 165–176.
10. Carela-Español, V.; Barlet-Ros, P.; Cabellos-Aparicio, A.; Solé-Pareta, J. Analysis of the impact of sampling on NetFlow traffic classification. *J. Comput. Netw.* **2011**, *55*, 1083–1099. [CrossRef]
11. Cole, G.; Bulashova, N. GLORIAD: A ring around the Northern Hemisphere for science and education connecting North America, Russia, China, Korea and Netherlands with advanced network services. In Proceedings of the TERANA Networking Conference, Pznan, Poland, 19–21 May 2005.
12. Cole, G.; Jun, L.; Sobieski, J.; Kim, D.; Riley, D. NSF Award: GLORIAD, Award number: IRNC-0963058. 2010. Available online: https://www.nsf.gov/awardsearch/showAward?AWD_ID=0963058 (accessed on 15 September 2020).
13. QoSient Argus. Available online: https://qosient.com/argus/ (accessed on 15 September 2020).
14. Hyun, J.; Van Tu, N.; Yoo, J.H.; Hong, J.W. Real-time and fine-grained network monitoring using in-band network telemetry. *Int. J. Netw. Manag.* **2019**, *29*, e2080. [CrossRef]
15. Gonzalez, A.; Leigh, J.; Peisert, S.; Tierney, B.; Lee, A.; Schopf, J.M. NetSage: Open Privacy-Aware Network Measurement, Analysis, In addition, Visualization Service. 2016. Available online: https://escholarship.org/uc/item/5rz6t3q4 (accessed on 15 September 2020).

16. Gonzalez, A.; Leigh, J.; Peisert, S.; Tierney, B.; Balas, E.; Radulovic, P.; Schopf, J.M. Big Data and Analysis of Data Transfers for International Research Networks Using NetSage. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Boston, MA, USA, 11–14 December 2017; pp. 344–351.
17. Deep Field Analytics. Available online: https://www.internet2.edu/media/medialibrary/2014/07/01/IS-deepfield-analytics.pdf (accessed on 15 September 2020).
18. Network Diagnostic Tool. Available online: http://software.internet2.edu/ndt/ndt-cookbook.pdf (accessed on 15 September 2020).
19. US Department of Energy Energy Sciences Network. Available online: https://my.es.net/Network (accessed on 15 September 2020).
20. System for Internet Level Knowledge (SiLK). Available online: https://tools.netsa.cert.org/silk/ (accessed on 15 September 2020).
21. Analysis Pipeline. Available online: https://tools.netsa.cert.org/analysis-pipeline5/ (accessed on 15 September 2020).
22. Lakkaraju, K.; Yurcik, W.; Lee, A.J. NVisionIP: Netflow visualizations of system state for security situational awareness. In Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, Washington, DC, USA, 29 October 2004; pp. 65–72.
23. Yin, X.; Yurcik, W.; Treaster, M.; Li, Y.; Lakkaraju, K. VisFlowConnect: Netflow visualizations of link relationships for security situational awareness. In Proceedings of the 2004 ACM workshop on Visualization and Data Mining for Computer Security, Washington, DC, USA, 29 October 2004; pp. 26–34.
24. Haag, P. Watch your Flows with NfSen and NFDUMP. In Proceedings of the 50th RIPE Meeting, Stockholm, Sweden, 2–6 May 2005.
25. Mellia, M.; Carpani, A.; Cigno, R.L. Measuring IP and TCP behavior on Edge Nodes. In Proceedings of the Global Telecommunications Conference, Taipei, Taiwan, 17–21 November 2002; pp. 2533–2537.
26. Elastic Stack and Product Documentation. Available online: https://www.elastic.co/guide/index.html (accessed on 15 September 2020).
27. Sun, G.; Xu, Z.; Yu, H.; Chen, X.; Chang, V.; Vasilakos, A.V. Low-latency and resource-efficient service function chaining orchestration in network function virtualization. *IEEE Internet Things J.* **2020**, *7*, 5760–5772. [CrossRef]
28. Sun, G.; Zhou, R.; Sun, J.; Yu, H.; Vasilakos, A.V. Energy-Efficient Provisioning for Service Function Chains to Support Delay-Sensitive Applications in Network Function Virtualization. *IEEE Internet Things J.* **2020**, *7*, 6116–6131. [CrossRef]
29. MaxMind, LLC. GeoIP Database. Available online: http://www.maxmind.com (accessed on 15 September 2020).
30. Western Regional Collegiate Cyber Defence Competition. Available online: http://archive.wrccdc.org/pcaps/2019/regionals/ (accessed on 15 September 2020).
31. Holik, F.; Horalek, J.; Marik, O.; Neradova, S.; Zitta, S. Effective penetration testing with Metasploit framework and methodologies. In Proceedings of the 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 19–21 November 2014; pp. 237–242. [CrossRef]
32. Zhang, J.; Chen, C.; Xiang, Y.; Zhou, W.; Vasilakos, A.V. An effective network traffic classification method with unknown flow detection. *IEEE Trans. Netw. Service Manag.* **2013**, *10*, 133–147. [CrossRef]
33. Fadlullah, Z.M.; Taleb, T.; Vasilakos, A.V.; Guizani, M.; Kato, N. DTRAB: Combating against attacks on encrypted protocols through traffic-feature analysis. *IEEE ACM Trans. Netw.* **2010**, *18*, 1234–1247. [CrossRef]
34. InSight2 Installation and Configuration Repository. Available online: https://github.com/angelkdev/InSight2 (accessed on 15 September 2020).