# PULSAR: Deploying Network Monitoring and Intrusion Detection for the Science DMZ

Shivam Trivedi
Purdue University
trivedi0@purdue.edu

Lauren Featherstun
Purdue University
lfeathe@purdue.edu

Nathan DeMien
Purdue University
ndemien@purdue.edu

Callum Gunlach
William Henry Harrison High School
gundlachcallum@gmail.com

Sagar Narayan
Purdue University
narayan8@purdue.edu

Jacob Sharp
William Henry Harrison High School
sharpjacob657@gmail.com

Brian Werts
Purdue University
bwerts@purdue.edu

Lipu Wu
Purdue University
wu857@purdue.edu

Carolyn Ellis
Purdue University
carolynellis@purdue.edu

Lev Gorenstein
Purdue University
lev@purdue.edu

Erik Gough
Purdue University
goughes@purdue.edu

Alex Younts
Purdue University
ay@purdue.edu

Xiao Zhu
Purdue University
zhu472@purdue.edu

## ABSTRACT

The Purdue Live Security Analyzer (PULSAR) is a state-of-the-art, high speed network monitoring and intrusion detection system designed to enhance the security of Purdue University's research cyberinfrastructure. PULSAR project goals include *empowering domain scientists* to conduct research at Purdue with heightened cybersecurity requirements and *engaging undergraduate students* through the design, deployment and operation of advanced cyberinfrastructure. Deployment strategies and design decisions are discussed, ultimately providing a recipe book for other institutions to use as a guide for effective implementation of a large scale intrusion detection system for Science DMZs.

## CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**; *Network security*; • **Networks** → *Network monitoring*.

## KEYWORDS

cyberinfrastructure, cybersecurity, science DMZ, research computing, intrusion detection systems

**ACM Reference Format:**
Shivam Trivedi, Lauren Featherstun, Nathan DeMien, Callum Gunlach, Sagar Narayan, Jacob Sharp, Brian Werts, Lipu Wu, Carolyn Ellis, Lev Gorenstein, Erik Gough, Alex Younts, and Xiao Zhu. 2019. PULSAR: Deploying Network Monitoring and Intrusion Detection for the Science DMZ. In *Practice and Experience in Advanced Research Computing (PEARC '19), July 28-August 1, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, **??** pages. https://doi.org/10.1145/3332186.3333058

## 1 INTRODUCTION

The Research Computing department at Purdue University supports the research needs of faculty at the university by providing state-of-the-art digital services, such as high performance computing and data storage resources. These resources are connected via a high speed network designed and optimized to support large scale data transfers and data intensive scientific application workflows.

The PULSAR (Purdue Live Security Analyzer) project puts a team of undergraduate and high school students as leads for the deployment and implementation of a Zeek (formerly Bro) [**?** ] network monitoring and intrusion detection system for Purdue's research network. PULSAR leverages a strong partnership among Purdue Research Computing, IT Security and Policy, the Department of Computer Science and the Purdue Polytechnic Institute. The project impacts a variety of research groups on campus and the growth of corporate partnerships by providing a more robust, reliable and secure research computing cyberinfrastructure.

## 2 MOTIVATION

The PULSAR project is driven by two main goals: empowering domain scientists by securing campus cyberinfrastructure and empowering network security education and training by mentoring undergraduates in the design and deployment of advanced cyberinfrastructure.

## 2.1 Empowering Domain Scientists

Purdue University has an advanced, large scale research computing infrastructure that provides high performance computing (HPC) and storage resources to over 200 research groups across campus. The core of Purdue's research cyberinfrastructure is the Community Cluster program [? ], which provides cost effective access to HPC resources and data storage. Purdue research cyberinfrastructure is designed for open science and follows the Science DMZ model [? ] for separating scientific computing resources from commodity and enterprise computing systems. The Science DMZ model requires security policies be tailored not to impact the performance of scientific computing environments. Typically, these policies do not meet the standards required by companies working in the areas of life sciences, medicine, aerospace and defense. The PULSAR project improves the unified security framework of Purdue's cyberinfrastucture by addressing the gaps that prevent the university from taking full advantage of potential partnerships with these government and industry partners.

## 2.2 Engaging Undergraduate Students

Development and education of the next generation of computational scientists and HPC professionals is an important part of Purdue Research Computing's mission. A variety of instructional methods have been utilized, from formal classroom instruction to practical experiential education and on the job training [? ]. This project is a next step to increase student involvement in developing advanced cyberinfrastructure.

In conjunction with their classroom experience, students are participating in real-world implementations with a high speed network and computing systems that are simply not available in a classroom setting. The team is paired with researchers and staff who provide guidance, additional informational resources, and university specific assistance such as procurement. This project flips the script of the typical "student employee" by allowing the student team to lead the implementation while the staff help to make sure it does not drive off a cliff. Such an opportunity to apply their conceptual knowledge to practical situations will help them build transferable skills and better prepare them as future workforce in the areas of networking, systems administration and cybersecurity.

## 3 STUDENT TEAM

Given the complexity and challenges of advanced research cyberinfrastructure, successful implementation and deployment of a network monitoring and intrusion detection system requires a team with a very broad skillset. The team as a whole needs to excel in multiple areas of networking, network security, and systems administration, as well as in large-scale data analysis and visualization. A decision was made to recruit a diverse set of talent that would *(i)* posess collective strength in all necessary areas, and *(ii)* have an opportunity to learn from each other and grow beyond their specific specialization, leading to more well-rounded and broader-skilled professionals at the end. A diverse student team was selected in January 2018. Along six Purdue undergraduates, the team included two juniors from William Henry Harrison High School (West Lafayette, IN) who were recruited from the school's 2017 Student Cluster Competition team. At the time of this writing, both high school students are accepted into Purdue's incoming 2019 freshman class.

Another important decision made by project leadership from the very beginning was to put most of the decision-making into students' hands. The mentors intentionally refrained from making technical decisions or handing out detailed project plans for the students to carry out. Instead, by making themselves available for general discussion and consultations, the mentors guided students to deep understanding of underlying technical problems. This allowed the students to come up with appropriate solutions on their own, which lead to better decision-making and sense of involvement. Sharing and discussion of alternative opinions, methods and ideas was consistently encouraged, and the environment of creative and stimulating partnership was intentionally maintained throughout the entire project, which was noted by both students and mentors to be very fulfilling.

Based on their specialization and interests, team members fell into two major subgroups, which could be broadly characterized as "systems engineers" and "security and policy analysts". Each subgroup held regular weekly operational meetings, as well as participated in a more strategic bi-weekly joint forum. Current and future project tasks were discussed, developed and defined during joint meetings, and consecutively assigned to students based on their workload and interests. Staff mentors actively encouraged interaction and cross-pollination between the student groups when assigning tasks so students could learn additional concepts outside their specific area of interest. Joint meetings also served as a venue for students to present upon completion of major milestones, as well as for lectures by project mentors and invited guest experts. To stimulate interaction between students and project staff outside of formal meetings, all students were given office space in close proximity to the mentors, and a mailing list and Slack workspace were also created and actively used.

## 4 ARCHITECTURE AND IMPLEMENTATION

This section provides details of the components that make up the PULSAR system. An architecture diagram with short descriptions of each system component can be found in Figure ??. At the time of writing, all components are deployed in a production quality system with the exception of the blocking component, which is in a testing and validation phase. Software and hardware components (e.g., Arista, xCAT) used in the system are well documented [? ? ], especially those related to Zeek [? ]. Inspiration and guidelines for traffic distribution and flow shunting techniques were found in the Berkeley 100G intrusion detection paper [? ]. However, in many cases, the exact implementation details are left to the individual to determine on their own. Sharing the knowledge our team gained throughout this process with the broader community of networking, systems and security professionals is a priority of the project. A link to an online guide for our deployment can be found in the Documentation section at the end of this paper, providing detailed, step-by-step instructions for deploying each component.

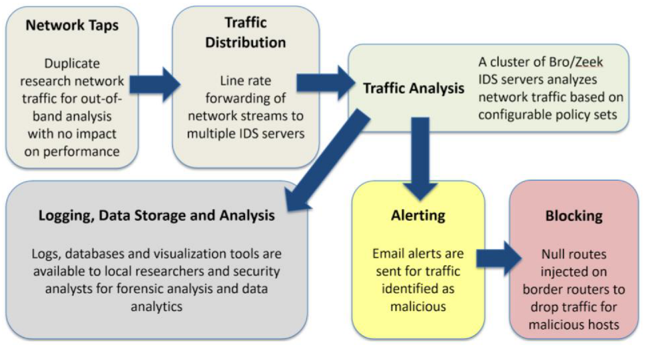

Network Taps
Duplicate research network traffic for out-of-band analysis with no impact on performance
Traffic Distribution
Line rate forwarding of network streams to multiple IDS servers
Traffic Analysis
A cluster of Bro/Zeek IDS servers analyzes network traffic based on configurable policy sets
Logging, Data Storage and Analysis
Logs, databases and visualization tools are available to local researchers and security analysts for forensic analysis and data analytics
Alerting
Email alerts are sent for traffic identified as malicious
Blocking
Null routes injected on border routers to drop traffic for malicious hosts

**Figure 1: PULSAR Architecture**

## 4.1 Purdue Research Network

The Purdue research network is aligned with the Science DMZ model for providing high bandwidth and low packet loss on a "friction-free" network path. Figure ?? provides an overview of the Purdue network, showing both the commodity/enterprise campus connections alongside the research connections. Most notable is the separation of research and commodity traffic at the WAN layer. Research traffic traverses the WAN on a 100G connection to the Indiana GigaPOP [? ] while commodity traffic takes a 20G path that includes an in-line intrusion prevention system (IPS). Purdue WAN, campus core and research core networks are connected in a ladder of two 400G meshes. PULSAR monitoring is performed on the four 100G network connections between the research core and the campus core. The enterprise datacenter and campus networks, including residential networks and the Purdue wireless networks are connected to the campus core via multiple 40G links. Research computing resources, high performance computing systems and large scale data storage systems are connected to the research core via multiple 100G and 40G connections. Average utilization across links between the research and campus core is between 10 and 20 Gbps, with frequent sustained network bursts over 60 Gbps. A three month usage pattern from Q1 2019 can be seen in Figure ??. The resource with the most network utilization is our CMS Tier-2 center, whose primary purpose is the storage and processing of data from the Large Hadron Collider at CERN.

## 4.2 Server Deployment Strategies

The PULSAR cluster makes use of multiple deployment strategies, as the cluster is made up of many different components. Given the complexity of the project, there is no single out-of-the-box solution that can perform all the functions of the PULSAR system.

*4.2.1 Zeek Network Monitoring Cluster.* The core of the PULSAR project is Zeek, a highly parallel network security monitor. While Zeek can be used to perform the traditional IDS functionality, it is extremely versatile and flexible, and can be used for much more than simple detection. A typical Zeek cluster consists of the following parts:

Worker: These are the processes which perform the core functionality of Zeek, namely analyzing and logging the traffic. Each worker process is a single thread, therefore multiple worker processes can be run on a single physical machine. This helps with the parallel nature of the Zeek infrastructure.

Manager: The manager process receives all the logs (unless there is a logger process), and de-duplicates notices received from different workers across the cluster.

Logger: This process receives all the logs from the Zeek workers if configured. In absence of a logger, the logs are automatically sent to the manager.

Proxy: A proxy process manages the synchronized state of all Zeek workers. This removed the need for individual Zeek workers from communicating directly to each other.

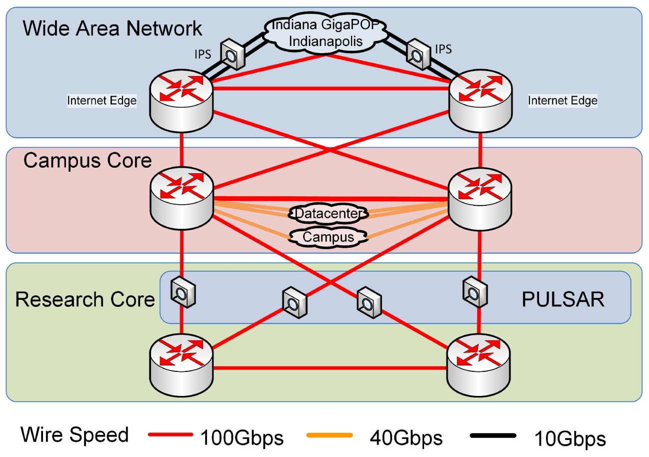
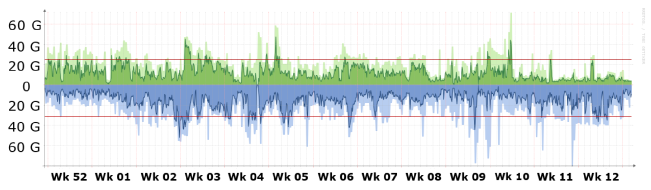


**Figure 2: Purdue Research Network Overview**



```
Bits/s   Last      Avg     Max      95th
■ In     4.03G   11.91G  71.91G   25.18G
■ Out    8.97G   15.04G  80.41G   31.46G
 Total   27.08P   (In 11.97P    Out 15.11P)
```

**Figure 3: Purdue Research Network - 3 Month Usage Pattern**

Given this architecture, the core cluster created by the team was as follows:

- `bro-a[000-007]`: A cluster of 8 machines which all run a number of Zeek workers. The primary purpose of these machines is to analyze network traffic.
- `pulsar-master`: A machine hosted as a virtual machine (VM) on a Kernel-based Virtual Machine (KVM) server, which runs the proxy, logger, and manager processes. A decision was made to host this system as a VM because the functions of this machine were not resource-intensive and using KVM allowed the team to flexibly assign more resources when required.

To increase the efficiency of the Zeek cluster, some third-party packages were integrated into the cluster. PF_RING was one of the modules that was used as it provided the team a way to direct individual traffic flows to Zeek worker processes at the server level.

*4.2.2 Server Deployment.* PULSAR server hardware consists of 17 machines, which include servers dedicated for Zeek workers, a KVM server for virtualization, Kafka brokers, and servers running the ELK (Elasticsearch, Logstash, Kibana) stack [**?** ]. In this project, the team made use of a variety of tools to cater to the different requirements of different components.

Puppet [**? ?** ] was used to manage the state of all the machines in the PULSAR environment. Puppet is an open source tool which automates the provisioning and configuring of nodes in a network. There are many alternatives to Puppet including Chef, Ansible, and Terraform that provide the same functionality, but Puppet was chosen as all of Purdue's research infrastructure was already maintained using Puppet [**?** ]. This meant that the team has access to an extensive internal codebase for reference, and access to Puppet experts within the organization. Puppet was used by the team to maintain the state of all PULSAR machines, with different modules for each type of machine. Puppet was used for many different purposes in the PULSAR project:

- to place configuration files in the correct locations
- to ensure that all the package dependencies were met
- to deploy keys to the system
- to manage the Zeek policy scripts
- to ensure that the correct permissions were set for all files

A traditional Puppet implementation usually consists of two components: a Puppet master and Puppet agents (which run on nodes). Here, the agent on the nodes periodically query the master, receive catalogs from the master, and then configure themselves based on the catalog. This system also allows the master machine to receive information from all the nodes. While model works well, the machine hosting the Puppet master can end up becoming the choke-point of the entire system and introduce scaling challenges. Over the last few years, due to the size of Purdue's research infrastructure, the research computing department moved towards a masterless Puppet model [**?** ], which was replicated in the PULSAR project. In this architecture, all the Puppet files are hosted on Github, and the PULSAR repository is used by all the machines in the PULSAR cluster. The machine periodically "pull" the changes from Github to the local copy, and the Puppet agent checks against the local repository to make the changes and perform audits of the system. This architecture is highly scalable because the Puppet load is distributed across multiple systems and pulling changes from Github requires little resources. In this scenario, there is no single machine that acts as a choke point, and even if Github is down, Puppet runs do not fail as they only refer to local copies of the repository.

The use of Github is instrumental to the PULSAR project as it is the central repository of all project codebase. The cluster was designed in a way that any change made to the PULSAR system had to be made by committing it to Github. This ensured accountability as every commit was attached to a user, and one can audit any line in the codebase to see when it was added and by who. This also helped in tracking changes and versions and allowed the team to roll back any changes that caused issues. Another advantage to using git was the use of branches, as different branches could be used for different purposes. For example, the initial changes could be added to the "pulsartest" branch, which automatically deployed the changes to the test cluster. If all the tests passed on the test cluster, the specific commit could be cherry-picked and transferred to the "deployed" branch which ran the production environment. This also allowed multiple team members to set up their own development environments for features that took a longer time to test and implement.

Apart from Puppet and Github, PULSAR makes use of xCAT, the Extreme Cloud Administration Toolkit [**?** ] to generate and distribute boot images for the Zeek worker machines. Maintaining consistency across the Zeek worker cluster is extremely important from the data integrity standpoint. At the same time, there is no persistent data that is stored on Zeek worker machines (each worker immediately ships its data to the Zeek manager process). These two considerations provide for using image-based stateless worker servers. With xCAT, images of the machines are generated on the the xCAT master, and with the use of DHCP, these images are served to Zeek worker hosts via PXE on boot. In this scenario, the operating system with all the changes is directly loaded into RAM instead of installing on disk. This allows Zeek workers to follow a serverless architecture where machines can be added and removed from the cluster at will. This way, all major upgrades can be performed by simply rebooting machines, and a reboot can also fix a machine with problems by returning it to the correct state.

## 4.3 Tap Aggregation and Traffic Distribution

Commodity or enterprise networks typically utilize inline intrusion detection and prevention systems at the network border to prevent malicious traffic from traversing wide area network links. Since these solutions perform inspection of each packet before forwarding, they cannot operate at high speeds and add considerable latency to each network connection, going against the Science DMZ model. The most feasible way to perform analysis on high speed network links (without adding latency) is through out-of-band analysis. Duplicating network traffic for out-of-band analysis is typically performed with either Switch Port ANalyzer (SPAN) ports or with Test Access Points (TAPs). The drawbacks of SPAN ports are well known, with the primary issue being that they easily become oversubscribed and drop packets, as they duplicate full-duplex (TX/RX) connections into a single TX signal.

For PULSAR, network visibility was achieved by installing passive Test Access Points (TAPs) on fiber optic connections between the research core routers and the campus core routers. The location of the TAPs provides visibility of network connections between research and campus hosts as well as research hosts and the Internet. Referring to Figure **??**, it can be seen that traffic destined for external networks has the potential to pass through both PULSAR and the commodity IPS depending on the destination. Traffic from research hosts at Purdue to research hosts at other universities on ESNet peerings only is analyzed by PULSAR, while traffic from Purdue research hosts to the commodity Internet will be analyzed by both PULSAR and the commodity IPS.

Our requirement was to be able to tap and aggregate the transmit and receive signals from the four full-duplex 100G connections between the campus core and research core routers. Network links

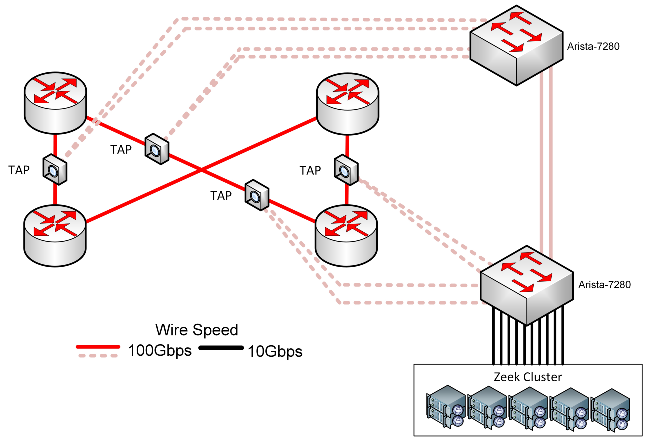


**Figure 4: Tap Aggregation and Traffic Distribution**

were tapped using Gigamon G-TAP G Series passive optical fiber TAPs. The tap aggregation solution needed to handle eight 100G TX signals and distribute them to a cluster of eight servers running multiple Zeek worker instances. This was achieved using two Arista 7280SRA-48C6-F devices for TAP aggregation. Each Arista 7280 has six 100G QSFP ports and 48 1/10G SFP+ ports. The Arista based solution offered significant cost savings over alternatives such as the Gigamon GigaVUE line of devices.

One Arista is used to aggregate four transmit signals from two of the TAPs into a 200G link aggregation group (LAG) connected to the second Arista device. Analysis of traffic volumes suggests that the 200G LAG will not face oversubscription issues at current traffic rates. The second Arista device then aggregates the four transmit signals from the remaining two TAPs and distributes traffic out ten 10G SFP+ ports to the Zeek cluster for analysis. The TAP aggregation and distribution design can be seen in Figure **??**.

## 4.4 Traffic Analysis, Logging, and Usage

The Zeek component of PULSAR automatically logs connection details of all traffic crossing the network links between Purdue's research core and campus core. This gives staff knowledge of what is traversing the network and provides the ability to audit traffic should any incident occur. Local policy determines any actions taken by PULSAR beyond simply logging traffic. Local policy defines which security incidents should be separately logged. For example, by implementing an SSH bruteforcing script in local policy with ACTION_LOG invoked, one can dictate that all SSH bruteforcing attempts should be detected and logged separately so that they are easy to identify. Local policy also dictates what patterns should result in staff being notified. By defining specific subnets to use in these policies, one can also specify that a notification should only be sent if the SSH Bruteforce attempt is coming from an host internal to the network. This ability to customize the response to various incidents and traffic patterns is what makes Zeek particularly powerful. This enables one to craft very specific security policies to fit the organization's needs. PULSAR generates a lot of data, some of which should be acted upon. Ultimately, a team will need to be established to respond to the security threats that PULSAR identifies, or this process will need to be automated.

PULSAR incorporates data from external intelligence sources to identify known bad actors on the network. PULSAR pulls data from these collective intelligence sources and compares it against traffic that is traversing the network. If there is a match, then this traffic flagged as malicious and separately logged so that it can easily be identified. This incorporates data points related to known bad actors such as IP addresses, hostnames, file hashes, more. If known malicious traffic is identified, the traffic is logged in notice.log. Ultimately, some solution will likely be implemented to automatically block the malicious traffic which is identified from accessing the network. Currently, PULSAR incorporates data from AlienVault OTX to identify bad actors. The advantage of AlienVault OTX is that it is free and incorporates a variety of community intelligence sources. AlienVault OTX was also chosen since it has a prewritten module that easily allows it to integrate into PULSAR's underlying software (Zeek). The team is also currently investigating other paid options to obtain community intelligence that may be more reliable.

There are several ways that PULSAR's data has been enriched. Incorporating intelligence data from outside sources provides value by helping those using PULSAR to understand the network data that they are seeing. The addition of labels, such as adding a "local" label for certain subnets, also helps those reviewing traffic to quickly understand what is happening. PULSAR's capabilities have also been extended by incorporating additional scripts. For example, a script to detect and flag traffic related to cryptocurrency mining has been implemented. A script to perform SSL Fingerprinting has also been implemented, giving insight on what is occurring inside encrypted traffic.

## 4.5 Zeek Policy Management

Configuring PULSAR's logging and alerting policies has come with some major challenges that anyone implementing a similar system should be aware of. A deployment of Zeek with little additional configuration will provide logging of any traffic that passes through it and the ability to separately log certain types of traffic and notify upon certain security events. This is sufficient for some auditing purposes or for some organizations that simply wish to have a basic understanding of what traffic is on their network. But, the real power of Zeek is in its ability to be customized and organizations or universities will likely want to harness this power.

Zeek policies are written using a Zeek Scripts domain-specific language. This is an event driven language and mastering it comes with a learning curve. While there are many scripts that apply to various security issues freely available online, these often must be tweaked to the specific site before they can be fully implemented. An organization that wishes to truly harness the power of Zeek must have someone who is proficient enough in the Zeek scripting language to codify desired polices. Past initial deployment, the organization must also retain security team members tasked with effectively management of script policies, as new threats emerge and new uses for the deployment are discovered. Organizations should also have a procedure in place for the deployment and modification of Zeek's polices.

Integration of PULSAR with outside systems may also present a challenge. For example, if a different solution is chosen for intelligence feeds, integrating its data into Zeek's Intelligence Framework may require writing additional software. If a new type of security threat is identified, the organization may need to write a custom script to monitor for the threat. Organizations should be prepared to dedicate staff time and resources for managing similar systems.

## 4.6 Log Storage and Analysis

The Zeek component of PULSAR produces between 20-30 GB of logs daily. Logs produced by Zeek are sent to a long term GPFS storage and to a Kafka [? ] cluster that acts as a message broker for an ELK (Elasticsearch, Logstash, Kibana [? ]) stack. The ELK stack gives IT Security and Policy and PULSAR team members the ability to quickly search and visualize Zeek data, while the GPFS storage acts as a secondary backup location for the raw log data. In the future, these log storage locations will allow data scientists to utilize the data for machine learning and other computational research projects.

Following Figure **??**, the log ingestion and storage process begins with Filebeat ingesting data from Zeek log files and outputting that data to the Kafka cluster. This data is stored in a single Kafka topic. Each topic is split up into multiple Kafka partitions across multiple nodes in the cluster to provide read/write parallelism for higher throughput and redundancy. When data is written to a Kafka partition it is given an offset number, these offsets are used to keep track of where data consumer nodes leave off in the data stream. In the event an Elasticsearch consumer goes down for maintenance or becomes unresponsive due to a technical issue, new data that's published during the downtime will not be lost as the consumer's place will have been kept and will automatically continue in place upon returning to service.

Once data is written to a Kafka topic, Logstash is then used to send data to Elasticsearch. Elasticsearch stores incoming data in databases called indexes, these indexes write to multiple shards across the cluster for read/write pluralism, similar to the Kafka partitions mentioned above. Shards are given a replication factor of "one" to provide data redundancy in case of a node failure. Shard replicas also play a big role in overall index search speeds as they serve read requests along side the primary shards.

Kibana is given access to the data stored on Elasticsearch so that powerful interactive dashboards can be created that gives us the ability to dynamically select time windows and narrow down specific data subsets to give us further insight into what is happening on our network at any given time. Users can create custom dashboards that can be used for monitoring anomalies or correlating the significance between anomalies detected on the network.

## 4.7 Flow Shunting

Flow shunting is a technique used to decrease the amount of processing required by an IDS by dynamically filtering large volume data transfers. Data transfer nodes can easily transfer files at multiple gigabits per second using protocols such as SSH, HTTPS, GridFTP and XRootD. Analyzing these "bulk" or "elephant" flows provides no real benefit from a security perspective as transfers are authenticated, data channels are encrypted, and transfers are logged by the server that is providing the data access service.

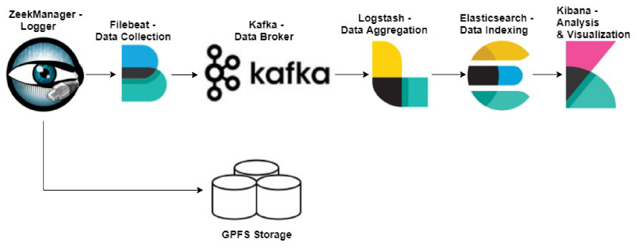

ZeekManager - Logger
Filebeat - Data Collection
Kafka - Data Broker
kafka
Logstash - Data Aggregation
Elasticsearch - Data Indexing
Kibana - Analysis & Visualization
GPFS Storage

**Figure 5: Data Distribution**

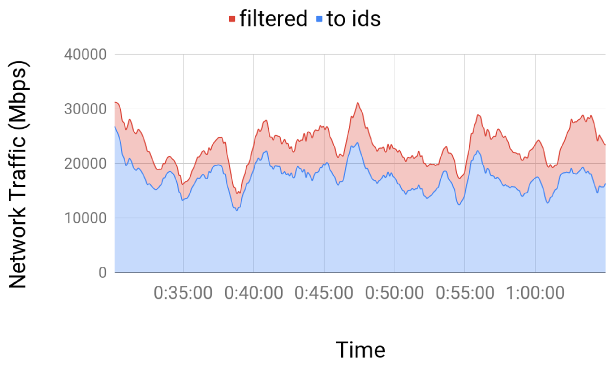


**Figure 6: Effectiveness of Flow Shunting. Total traffic volume is shown with filtered (red) traffic and traffic analyzed by Zeek (blue).**

Flow shunting is implemented using the Zeek reaction framework [? ], a python script called dumbno [? ] and the API of the Arista traffic distribution devices. The reaction framework is responsible for identifying bulk network flows. Bulk flows are identified after the connection reaches a certain configurable threshold (128 MB by default). Once a bulk flow is identified, 4-tuple (src/dst IP address and src/dst port) connection information is sent to dumbno, which sends a command to the Arista API to update an access control list on the Arista device to drop further packets from the bulk connection. The ACL permits TCP control packets (SYN/FIN/RST), allowing Zeek to keep track of shunted connections and log connection size and duration even though no data packets are analyzed. Figure **??** shows the impact of flow shunting on the amount of traffic sent to the Zeek IDS during a 30 minute period. In our experience, flow shunting provided a 10-30% decrease in the amount of traffic analyzed by Zeek.

While flow shunting provides an obvious benefit, we found that shunting alone was not effective enough to prevent the network links to our Zeek workers from being overloaded. It is common for the Purdue research network to experience bi-directional bursts of traffic that exceed 100 Gbps (combined TX/RX). Currently, traffic is analyzed by 8 servers, each with a single 10G network card,

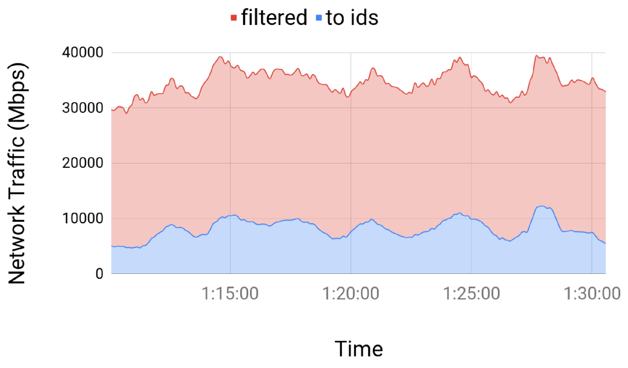


**Figure 7: Effectiveness of Flow Shunting and Whitelisting. Total traffic volume is shown with filtered (red) traffic and traffic analyzed by Zeek (blue).**

meaning that the amount of traffic analyzed by Zeek cannot cross the 80 Gbps threshold without packet drops.

The reaction framework will shunt a connection when it crosses a configurable threshold. Currently, the reaction framework is configured to shunt a connection after 128MB of traffic is detected. The reaction framework can also shunt a connection when Zeek identifies it as a GridFTP data channel. Typically, these can be identified after only a few megabytes of traffic. This feature has the potential to significantly decrease the amount of data processed for all GridFTP connections. Zeek uses a heuristic to identify a GridFTP data channel that assumes a null cipher will be used for data transfer. We found the majority of GridFTP connections that occur on the Purdue research network come from the CMS Tier-2 Center, which encrypts all GridFTP data channels. The CMS Tier-2 also uses a file access protocol called XRootD for remote dataset reads. GridFTP and XRootD traffic to/from the CMS Tier-2 is typically 75% of all traffic seen on the Purdue research network. Since both protocols are authenticated, encrypted and logged at the server level, we considered the ports and IP addresses associated with these services to be "trusted" and configured the Arista ACLs to drop all packets from these connections (while still permitting TCP control packets), significantly decreasing the amount of traffic analyzed by the Zeek workers. Figure **??** shows the impact of using flow shunting and traffic whitelisting of known services running on known host/port combinations. Combining the flow shunting and traffic whitelisting techniques provides a 70-80% decrease of the amount of traffic analyzed by Zeek and further mitigates issues with oversubscribing the 80G aggregate bandwidth to servers running Zeek worker processes.

## 5 RESULTS

At the time of writing, PULSAR has been in production for approximately six months, providing detailed logs of network traffic and notices for events of interest on the network.

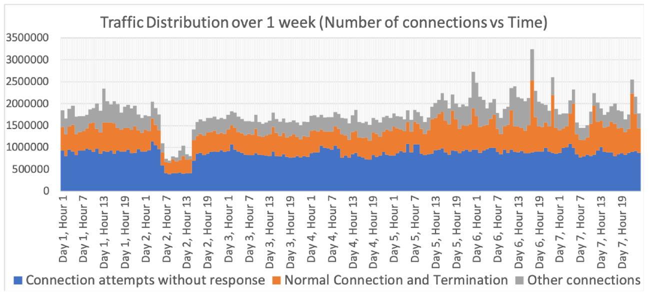


**Figure 8: Traffic Statistics reported by Zeek**

### 5.1 IDS statistics

Currently, PULSAR monitors all network traffic passing between the research core and campus core. Figure **??** shows the number of connections monitored by the Zeek cluster over a span of 1 week. The traffic in the figure is split into 3 categories:

- Connections that are attempted but do not get a response. These also include all the network scans that occur on the research infrastructure.
- Connections that are connected and terminated normally.
- All other connections, including any partial traffic that may be seen on the network

The Zeek cluster monitors on average between 1.5 and 2 million connections every hour. A dip in traffic for a few hours on day 2 represents planned maintenance work on a research network router hosting a tapped network link.

### 5.2 Operational Experiences

On the first day of analyzing production network traffic, PULSAR logged an extensive port scanning effort by an internal (privately addressed) host. Analysis of the Zeek connection logs showed the internal host making repeated HTTP requests to an external host. After several requests, the host would start scanning either the entire A, B or C private subnet space on a specific port. After finishing the port scans, communication was logged from the internal host to hosts associated with the Tor network. The malicious host was discovered to be a virtual machine in an OpenStack instance used for teaching and instruction. The virtual machine was shut down and preserved for forensic analysis. It is not known how long this machine had been scanning the internal address space before it was flagged by PULSAR. This type of attack could not have been detected by the security measures in place before PULSAR was deployed and is a prime example of why network based intrusion detection systems are necessary for securing research cyberinfrastructure.

In a case where thousands of files were accidentally deleted from a research lab's Samba based network folder, PULSAR provided an additional level of auditing of all file delete operations between the client and the Samba server. An analysis of the Samba server logs and the logs generated by the Zeek SMB protocol analyzer showed and confirmed the same event from two distinct locations, at the server level and the network level.

PULSAR is configured to send email alerts on certain events only if they come from internal hosts. For example, port scan notifications from external hosts are logged, while port scans from internal hosts send an email to the PULSAR team. These events are then forwarded to IT Security and Policy systems for incident response.

## 6 FUTURE WORK

While the implementation of the majority of PULSAR components is complete, there are still many potential areas of investigation to fine tune and expand its capabilities.

While our testing shows that PULSAR is accurately identifying and alerting on malicious traffic, a cross validation of IDS statistics for traffic that traverses both the commodity IPS and PULSAR could provide additional assurance that our scripts and policies work as intended. Discussions have also begun about enhancements that could allow PULSAR to respond to security events in real time and act as an IPS, instead of just an IDS. After six months of alerting on port scans and SSH brute force attacks, the team has established a set of criteria that can be reliably used for blocking traffic at campus border. A solution currently under development involves PULSAR acting on notifications of external threats and sending null routes to campus border routers to block malicious hosts from accessing Purdue's network.

It is our hope that PULSAR can now act as a springboard for other cybersecurity and networking related research projects as well as programs aimed at undergraduate teaching and learning. Students working with Purdue's IT Security and Policy department can assist security analysts by using the Kibana interface to mine PULSAR log data and develop new alerting and dashboards. Students and staff from Research Computing can investigate alternative Zeek server deployment strategies (Docker/Kubernetes) as well as traffic distribution and filtering techniques (SDN/OpenFlow) using a PULSAR testbed that sees a portion of production traffic. Progress has also been made in creating an anonymization and distribution framework for the logs produced by PULSAR. This data could be used by researchers who wish to understand more about network traffic and network security on large Science DMZs.

The development of PULSAR will continue to be an ongoing processes. As security staff use PULSAR, they will likely come across different events they want to log, or different ways they wish data to be presented. With this will come tweaks to the scripts that PULSAR uses. More value will be derived from PULSAR over time as it is continuously modified to fit the site's needs.

## 7 CONCLUSIONS

Implementation of a network monitoring and intrusion detection system at a large scale is a challenging and rewarding project. The project goals of empowering domain scientists and engaging undergraduate students were successfully met. PULSAR has increased the security of Purdue's research infrastructure by increasing visibility of network traffic. The system effectively filters and analyzes large amounts of traffic without impacting the speed of the network. PULSAR aligns Purdue's computing environment with heightened security requirements of industry partners and enables new research opportunities for scientists.

This project provided an opportunity for students to learn new skills and become well versed in a real world deployment of advanced cyberinfrastructure. Students used teamwork and their complementary skills in networking, network security and systems administration to make decisions, lead the deployment and significantly contribute to successful implementation of a complex, meaningful and valuable project.

## 8 ACKNOWLEDGMENTS

This project is supported by the NSF CICI Award (#1738981).

## 9 DOCUMENTATION

Deployment documentation, digital artifacts and anonymized data sets will be available on https://github.com/PULSAR-security