FileTSAR Final Summary Overview

Problem Statement

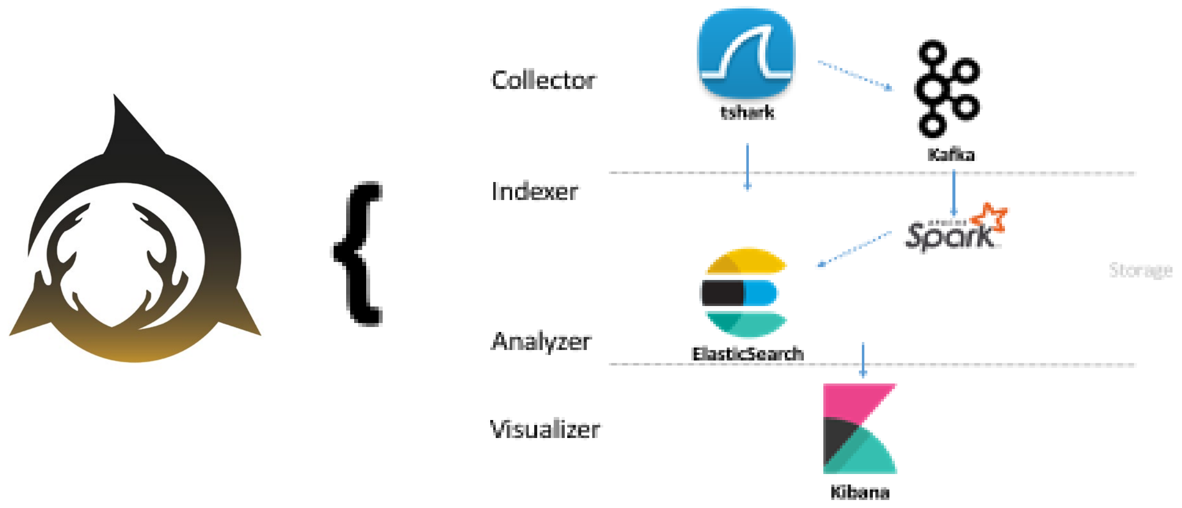
The forensic investigation of files and computers already poses major challenge, and file recovery and file carving are well known processes on these systems to reconstruct deleted file data. However, attempting to observe, capture, and reconstruct files of any type from only network data is significantly more complicated. Unlike data stored on computers, mobile phones, or tablets, networks forward this data in a transient nature and do not store the data. Additional tools are necessary to capture the file data as it transits the network. Network packet and flow capturing tools must be installed and enabled on precise, controlled, and coordinated devices within the enterprise-scale network in order to make any file and data analyses possible in a forensically sound manner. Thus, the objective of this project was to create a unifying toolkit for law enforcement known as FileTSAR (Toolkit for Selective Acquisition and Reconstruction of Files).

FileTSAR

In our implementation, the acquisition and analysis of data from the network are divided into several processes: 1) packet capture (i.e., recording the packet traffic on a network), 2) protocol parsing (i.e., parsing out the different network protocols and fields), 3) search and analysis, and 4) visualization. Based on the goals of this project, the architecture was divided according to functionality with a storage repository included (see Figure 1).

The Collector module captures network traffic and saves the traffic to the Storage implementation. The only interconnections to the large-scale network required for FileTSAR are via the Collector. The engine is built using open source tools and custom code wrappers. This makes it compatible with existing incident response systems as well as provides a standardized interface into other modules in FileTSAR. While the tools focus on conducting live captures during active investigations, an additional set of functionalities designed into the toolkit can ingest previously captured data (PCAPs) from other systems already present in the enterprise network. This allows for the ingestion of data previously captured from the intrusion detection/prevention system, firewall, incident response system, and logs.

Figure 1. The architecture of FileTSAR



The Indexer module takes input from the Collector module and then processes it for file contents. The Parser included in the Indexer architecture has two distinct components: Packet Parser and Flow Parser. These two components handle the different types of data being processed. The Packet Parser takes each packet in the PCAP file. Raw packet data contains an extraordinarily large number of fields. While network traffic itself is sent in a binary format, each packet contains many different fields which can be parsed out into numbers, text, timestamps, and IP address, for example. The Packet Parser parses the packets into their respective Application layer, Transport layer, and Internetwork layer; subsequently, the data is archived into the active case directories within the Storage subsystem to be explored, searched, and visualized later. This output is formatted to JSON, which is well known industry accepted format. The Indexing Engine then constructs an optimized data structure or set of data structures for rapid and intensive retrieval and processing by the Analyzer module. The Flow Parser works by identifying a group of packets sent over the same time period sharing common properties, such as the same source address, destination address, and protocol; that is, it identifies the packets that belong to the same flow. As each flow is unidirectional, if there is a corresponding flow, it is parsed and stored as a separate flow. Each flow is stored in the DFXML format. Additional data integrity is added to the indexing process by including hashing function output as part of the data structures. This provides validation of the data being stored and processed by ensuring that it remains the same throughout the investigation and indicating if any tampering of the data had occurred throughout the digital forensic investigation.

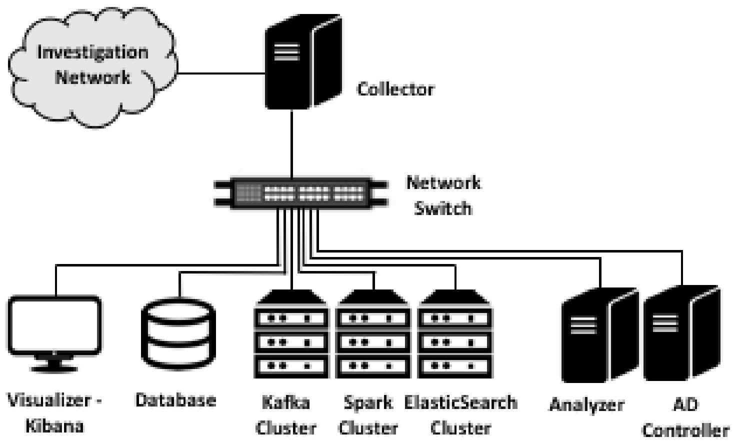
One of the major capabilities of our implementation is the analyzer. Being able to look into every single piece of metadata and payload provides very useful visibility and helps to monitor systems, detect anomalies, and detect attackers. The Analyzer module performs the critical functions of identifying interrelatedness of files, flows, packets, users, timelines, etc. The Selector Engine in Analyzer receives input from the Selector Agent in the Visualizer module to further refine the dataset being analyzed by the forensic investigator. This dataset, or datasets, are then passed back to the Collector for rapid search, selection, identification, and relating of individual data points into a collection of resources for further investigation. The Analyzer feeds this data to the Visualizer for presentation capabilities. An additional set of functionalities in Analyzer module is to selectively reconstruct multiple types of data, including documents, images, email, voice over IP conversations, and messaging. FileTSAR is capable of indexing and parsing the following protocols: HTTP, FTP, SMTP, IMAP, IMF, SIP & RTP.

The Visualizer module provides the interface for the digital forensic investigator to identify trends, patterns, or repetitions. The visualization capabilities are combined in an authenticated, web-based dashboard to provide dynamic environment to select and view reconstructed files and file fragments, for example. Only authenticated users are allowed to access the Visualizer module. This authentication requirement provides an additional level of legitimacy and confidentiality of the investigation through closure. Users authenticated via the visualizer can create a task with the interface. The Visualization module is used to start the collection process via the Collector, as well as to view the analysis of the data provided via the Indexer and Analyzer. The front end of the system that investigators use to interact with FileTSAR is handled by the visualizer.

In order to start the collection process, or to visualize the collected information, the Visualizer links the Indexer and Analyzer modules to the web front end, which allows for examiners to actively look through the data. The web application interface of the toolkit contains a "dashboard" to notify the user how much of the collector, indexer, or analyzer is in progress. The visualizer is built on an open source analytics and visualization platform and is embedded in the interface for the user to analyze network traffic. The reconstructed files from the PCAPS are also displayed on the "dashboard." The dashboard is dynamic so as to allow specific visualization modules to be loaded, unloaded, or added to provide the specific environment needed by the digital forensic examiner. This also provides the utilities to allow investigators to open/close the investigation, identify and list files of probative value, itemize, and detail each step of the investigation for admissibility to the courts.

These roles are implemented using a collection of open-source tools along with custom code wrappers to integrate the data. This custom code that provides the integration functions is written Python 3. The primary Collector functionality is implemented using tshark, which is responsible for packaging the PCAP and sending data to the Indexer. The Indexer functionality is implemented in Apache Kafka, Spark, and ElasticSearch. Data flows from Kafka as JSON into Spark, which itemizes each packet and flow. Once data is populated into ElasticSearch, the Analyzer functionality is implemented with tcpflow and additional ElasticSearch functionality. This allows the Visualizer to present, sort, and select the data to move from probative information to identify evidence. The Visualizer is implemented with Kibana (see Figure 2).

Figure 2. The Visualizer for FileTSAR



Beta Testing

For 30 days, a beta test was conducted to gain detailed feedback and suggestions on FileTSAR's feature and functionality implementations. Digital forensic examiners from the Tippecanoe County High Tech Crime Unit and the National White Collar Crime Center were given remote access to the toolkit along with a user manual that provided details on functionality and usage. These examiners represented the broad range of knowledge in the area of network forensic investigations, from entry-level, or basic training, to advanced training and knowledge in the area. The research team incorporated a ticketing system in order to provide technical support to the examiners during testing.

The examiners were provided with an anonymous link to an Internet-based survey in order to provide feedback on the toolkit as they experienced the features and functionality. The survey was comprised of three sections: bug reports, feature requests, and overall feedback. The examiners were able to provide feature requests and bug reports for the following categories: setup and login, user interface, capture options, status management, dashboard, and final reconstruction. In addition, the anonymous survey included open-ended questions to which the examiners were able to provide additional feedback and/or comments.

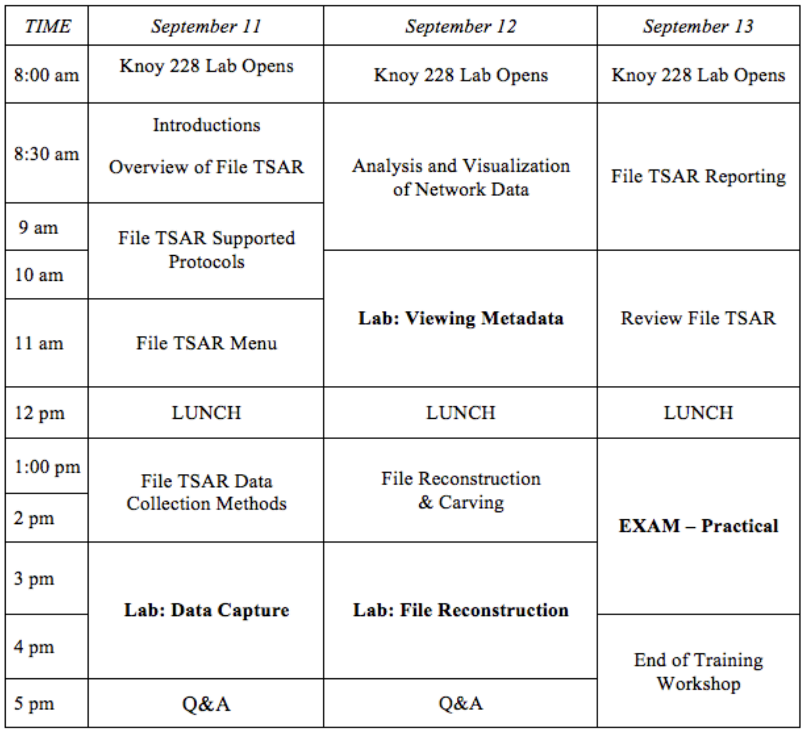
Based on the feedback and suggestions provided, the examiners were satisfied with the core functionalities, such as analyzing network data via the Visualizer and the reconstruction of files, emails, and voice over IP sessions. The survey results also showed support for the manner in which the volume of data was handled as well as the level of granularity that could be attained in the data provided. In addition, the examiners made suggestions on the user interface and user experience components that would ultimately increase user-friendliness for future versions of FileTSAR. After beta testing, we worked to include additional features and improve the flow of FileTSAR's functionalities and clarify the nomenclature so as to make case management more intuitive for the examiners. While we finalized the development of FileTSAR, we planned a 3-day law enforcement training workshop for digital forensic examiners across the United States.

Dissemination

TechnoSecurity Conference. The PI and lead architect for FileTSAR attended the TechnoSecurity Conference in Myrtle Beach, SC from June 4-6, 2018. We presented FileTSAR as a part of a workshop organized by NIJ's Martin Novak, *Big Forensics – Investigating Very Large Organizations.*

FileTSAR Law Enforcement Training Workshop. A 3-day law enforcement training workshop was hosted at Purdue University for FileTSAR from September 11-13, 2018 (See Figure 3). We recruited from law enforcement agencies across the United States to fill 20 seats at the training workshop. All tra vel expenses were covered and this 3-day training workshop was free to law enforcement thanks to federal funding from the NIJ. The FileTSAR training workshop was open to sworn law enforcement officers who are permanent residents in the United States. Interested digital forensic examiners applied through an event page, and answered questions regarding their agency size, current job duties, and level of knowledge in network forensics and basic digital forensics.

Figure 3. *FileTSAR Training Workshop Schedule*



TIME	*September 11*	*September 12*	*September 13*
8:00 am	Knoy 228 Lab Opens	Knoy 228 Lab Opens	Knoy 228 Lab Opens
8:30 am	Introductions Overview of File TSAR	Analysis and Visualization of Network Data	File TSAR Reporting
9 am	File TSAR Supported Protocols		
10 am		**Lab: Viewing Metadata**	Review File TSAR
11 am	File TSAR Menu		
12 pm	LUNCH	LUNCH	LUNCH
1:00 pm	File TSAR Data Collection Methods	File Reconstruction & Carving	**EXAM – Practical**
2 pm			
3 pm	**Lab: Data Capture**	**Lab: File Reconstruction**	
4 pm			End of Training Workshop
5 pm	Q&A	Q&A	

A week prior to the training workshop, 20 sworn, certified digital forensic examiners from across the United Sates (Arizona, California, Indiana, Illinois, Pennsylvania, South Carolina, Tennessee, and Texas) confirmed their registration for the FileTSAR training workshop. However, there were 3 last-minute cancellations due to casework and 1 due to Hurricane Florence. 16 digital forensic examiners participated in the FileTSAR training from across the United States (Arizona, California, Indiana, Illinois, Tennessee, and Texas) and from different agencies (local, state, and federal law enforcement).

The training workshop provided a general overview on networks and protocols as well as the functionalities of FileTSAR (data capture, analysis and visualization, file reconstruction, and reporting; see Figure 3). For the labs, the examiners were provided with a PCAP of a fictional criminal case so they could work through the different functionalities of FileTSAR. On the 3rd day, the examiners completed a practical exam testing their use of FileTSAR on a new PCAP of a different fictional criminal case (Practical Exam received IRB approved). For the final exam scenario, the average exam score was 88.8% regarding the items related to using FileTSAR to investigate the test scenario.

The examiners were also able to provide anonymous comments and questions at the end of the exam. Below are some of the anonymous comments for FileTSAR:

1. I really like where the tool is headed and I understand the tool was designed to the NIJ standards to complete large scale investigations but within that lies the problem for most local or state agencies. Most of our investigations are on a much smaller scale and thus would not need the same resources required to complete the large-scale investigations. If there was a way to scale down the system requirements so that our vastly underfunded forensics lab could use the product then I think it could be of value to our department. I would like to see a bit more automation especially in the VOIP dashboard. It was a difficult task identifying the IP address for Alice as you had to drill down considerable to locate it. Not knowing much about VOIP communications makes this a challenging task. If the module could be setup to automatically obtain the correct sender/receiver IP address I believe it would make the process easier to figure out who is involved in the conversations and to actually use the data for evidence purposes. I really appreciate the

opportunity to come to Purdue and learn about File TSAR. I have learned a tremendous amount of general network traffic knowledge and would love to delve deeper into cyber security. Thank you

2. I anticipate setting up FileTSAR on a mobile server cart at my Lab. I would like know the configuration suggestions. For instance, if there is documentation on the install of the software, versions of OS, service packs, DB software, etc.? That would be great.

3. Thank you for the training and opportunity to attend. We are a smaller agency and have very little experience with network capture investigations thus far. We know it's coming. That said, it's nice to have a tool for that time. If the tool was smaller and easier to deploy, we might be able to use it sooner. But without a demand for those types of investigations yet, I'd be hard pressed to secure the funding to build the network needed as was described in class. Also, we do not use an IT unit to maintain our systems. We have a few in the lab who have network and limited server experience, but that's it. So.... an opinion from my smaller departments angle would be a lite version and a collection tool. This gives us a cheaper starting point to work up from. It was hard for me to wrap my mind around the tool at first and how to use it being we haven't done these investigations, but by the end it was much clearer. A list of rerecorded specs and/or equipment would help also being we'd likely start from the ground up for a system. Thank you again.

4. Great program and great instructors. I appreciate the opportunity to learn about File TSAR. The big challenge (inter-dept) will be to have portable equipment to apply.

5. I really enjoyed the introduction as well as learning the current capabilities of the program. I think the functionality and ease of use at this point is good. I like the interface and once completed familiarity would definitely help. As far as deployment not sure where it will go, but as discussed storage and budget limitations as well as backend support are key considerations.

6. Great concept overall. I believe this tool, in its current form, would best serve IT departments that already has equipment that meets the minimum system requirements. Medium to large businesses, as well as larger LE agencies, especially Gov. IT departments. For example: ISP CC & ICAC hve a robust server on site, managed by the state. ISP could utilize FileTSAR for casework (upload pcap option), Indiana IT could use it in a live response capacity for that site. ISP is unique, however, to the extent that they have a larger budget / equipment and the potential support to see a project like this through. The crash cart scenario or the few iterations of, are good ideas, but doesn't fall within the means of the majority of law enforcement agencies... As we talked about in class, if FileTSAR could be scaled down and deployed as a packaged executable... Most agencies have a robust forensic computer that would work, although not as efficient. Great job so far, I hope to see this being utilized in the field soon!

Based on the feedback we received from our training workshop, we made small updates to FileTSAR in October and November then finalized all development in December 2018.

2nd International Workshop on Big Data Analytic for Cybercrime Investigation and Prevention. We published a paper at the 2nd International Workshop on Big Data Analytics for Cybercrime Investigation and Prevention at the 2018 IEEE International Conference on Big Data in Seattle, Washington in December. Dr. Seigfried-Spellar presented this paper at the IEEE Big Data Workshop in Seattle in December 2018 (see Hansen et al., 2018).

FileTSAR Website. We have developed the following website, which provides information about FileTSAR as well as access to training videos for Law Enforcement: https://polytechnic.purdue.edu/facilities/cybersecurity-forensics-lab/tools. Law Enforcement interested in FileTSAR are asked to contact the authors at filetsar@purdue.edu, we will provide them with a username and password to access the secured training videos, installation manual, and user manual for FileTSAR. We consulted with the Tippecanoe High Tech Crime Unit as we created our training videos; we have decided to create 9 short videos instead of a long webinar or tutorial; this way, if they have a specific question about the tool (e.g., capturing data), they can access that specific training video rather than trying to identify which part of a long video they would need to watch. All videos are completed and uploaded to our secured training site.

Office of Technology Commercialization (Purdue). We currently have a trademark for our logo; We have filed a provisional patent (62/841,316). We have been contacted by over 100 LE officers from around the US and world, and once our patent paperwork is finalized, we will begin disseminating licenses for FileTSAR.

Respectfully Submitted by Dr. Kathryn Seigfried-Spellar (PI)

File Toolkit for Selective Analysis & Reconstruction (FileTSAR) for Large-Scale Networks

Raymond A. Hansen*
Department of Computer Science & Networking
Wentworth Institute of Technology
Boston, MA, USA
*hansenr2@wit.edu

Kathryn Seigfried-Spellar, Seunghee Lee,
Siddarth Chowdhury, Niveah Abraham,
John Springer, Baijian Yang, and Marcus Rogers
Department of Computer & Information Technology
Purdue University
West Lafayette, IN, USA
kspellar@purdue.edu

***Abstract*—There are many challenges in digital forensic investigations involving large-scale computer networks; these include large volume of data, the limited scope of tools, the financial burdens of purchasing and licensing those tools, and identifying salient evidence from the vast amounts of network data. We have implemented a collection of open-source tools and code wrappers to provide a tool for network forensic investigators to capture, selectively analyze, and reconstruct files from network traffic. The main functions of this tool (FileTSAR) are capturing data flows and providing a mechanism to selectively reconstruct documents, images, email, and VoIP conversations. To validate the large-scale capabilities of the toolkit, we conducted a "stress test" of the system using approximately 123,500,000 packets from a collection of packet capture files totaling nearly 100GB. Additionally, sixteen (16) digital forensic examiners participated in a 3-day law enforcement training workshop for FileTSAR from across the United States; the examiners expressed substantial support for FileTSAR with large-scale investigations as well as an interest in a scaled-down version for smaller agencies with storage, budget, and back-end support limitations.**

***Index Terms*—network forensics, network forensics investigations, digital forensics tools, file reconstruction, packet capture**

I. Introduction

Many modern criminal investigations involve digital evidence; often from multiple digital devices [1]. As a result, law enforcement relies on specialized digital forensic investigative tools to acquire, evaluate, process, and present the probative data in a forensically sound manner to the criminal justice system. However, the digital forensic tools that are currently available to law enforcement have multiple significant challenges: limited scope, multiple tools required to complete an examination, incompatibilities with other tools, correlation of data between tools, outdated for newer/ever-evolving technologies (e.g., cell phones, gaming consoles, new Operating Systems), large amounts of data, the cost of purchasing, and the costs of certifying and maintaining licensure for investigators using these tools [2] [3] [4]. While there are several well-known and reputable commercial products for capturing and analyzing evidence that originate from computers, there are too few complete tools of the same caliber for network forensics.

The large data volume issue is compounded for networks where 100s or 1000s of computers and other systems are interconnected over various technologies. That is, there is a vast amount of data being created and transmitted on a daily basis by large-scale computer networks [5] [6]. It is estimated that the total amount of data generated will reach 163 zettabytes (163 billion TB) by the year 2025 [7]. This will result in 3.3ZB of traffic over the Internet backbone [8] with up to 100x this amount flowing over private enterprise networks per year. This creates a major challenge for the ability to capture that amount of network traffic as well as an additional challenge to have the ability to perform any useful analysis. According to [9], there are organizations that routinely record some or all of the traffic on their external Internet connections. This defines an explicit need for tools capable of dealing with large volumes of data.

An additional challenge for investigators is the architecture of existing tools for network forensics. The tools currently available are generally single-computer systems and often single threaded, making them ill-equipped to investigate modern networks. While each of these tools performs tasks that can be functionally categorized, their efficacy, efficiency, robustness, and reliability vary widely [10]. Additionally, many fail to follow any accepted standard for data formats or digital forensic processes, lack sufficient provenance of evidence as a result of their processing, and are suspect in their ability to operate in a forensically sound manner.

A number of these tools are developed and maintained by a commercial entity that provides many of the features of professional applications: technical support, refined interface design, and sometimes predictable patch and upgrade schedules. However, there are significant disadvantages to these proprietary tools as well as cost is almost always a determining factor for use. The products and services provided by Splunk, a company that manages a set of popular security event management platforms capable of correlating data across large volumes of data, can potentially cost $1000s/year (see www.splunk.com). At the less expensive end of the spectrum is NetworkMiner, an open source Network Forensic Analysis

This project was funded by the National Institute of Justice, Office of Justice Programs, U.S. Department of Justice (#2016-MU-MU-K091). The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the U.S. Department of Justice.

Tool, that costs $900 per seat license that is current for three years. Whereas Splunk can generally handles 100s of GBs of data from disparate sources, NetworkMiner only handles PCAP data of a few GBs in size. These tools are designed to handle a variety of use cases but still come with significant price tags for most law enforcement agencies.

Extending this current state of tools, having forensically sound processing and processes, and dealing with the exascale big data problem is necessary for continued advancement of admissible evidence within the courtroom [5]. [5] discusses various instances where the manner in which digital evidence was collected can benefit or hinder the judicial process. In the first case, evidence was acquired from cellphone apps and the communicating cellular tower, resulting in the conviction of the suspect. In the second case, mistakes in the acquisition procedures of digital evidence weakened the prosecutions case in a murder trial. In the third case, a victim with almost no digital fingerprint left nothing of inculpatory value for detectives to determine possible suspects. These highlight the value of sufficient tools and training for law enforcement personnel while also demonstrating the downsides of poor or insufficient training (or adherence to defined processes).

In Section II, we discuss related tools and other prior projects. Section III details the generalized architecture of FileTSAR. Next, Section IV covers the process validation and reconstruction validation of FileTSAR: 1) beta-testing phase involving examiners from the HTCU and NW3C, 2) 3-day law enforcement training workshop with examiners from across the United States, and 3) a test of FileTSAR's reconstruction capabilities using a 100 GB packet capture (PCAP) file. Finally, Section V details the future work and conclusions.

II. Related Work

The forensic investigation of files and computers already poses major challenges, and file recovery and file carving are well known processes on these systems to reconstruct deleted file data [11], [12]. However, attempting to observe, capture, and reconstruct files of any type from only network data is significantly more complicated. Unlike files and data stored on computers, mobile phones, or tablets, the networks forward these data in a transitive nature and do not store the data. Only in cases where the network administrators or network engineers have specifically configured network infrastructure to do so are traces of those file or data transfers present [13]. This approach still cannot provide actual file content of probative value nor provenance of that data to a digital forensic investigator. Additional tools are necessary to capture the file data as it transits the network. Network packet and flow capturing tools must be installed and enabled on precise, controlled, and coordinated devices within the enterprise-scale network in order to make any file and data analyses possible in a forensically sound manner. Thus, not only is the digital forensic investigation of computers, phones, tablets, and other devices necessary, it is also required on the network for forensically sound reconstruction of files and data. [10] provided a general classification of tools used for these types of investigations.

While many of these tools have been maintained, it is also clear that many of them have not. For example, ngrep is currently at version 1.46. Version 1.43 was released in late-2004 or early 2005. There was a nearly eight-year gap between 1.44 and 1.45, and another 3 years between versions 1.45 and 1.46. Additionally, Nessus and Snort, while originally free and open source projects, have been purchased by corporations that have either closed their codebase and required licenses for non-home use or closed the code management and development process to suit the companys needs rather than investigator's needs.

The reliance upon tools that have an indeterminate update cycle, or lifecycle, can present ongoing challenges for the investigator. This could require the use of outdated operating systems, compilers, or system tools just to keep a specific investigative tool running. This could then result in conflicting libraries, tools, or supporting applications that have the potential to interfere with the accurate processing of the evidence. Additionally, there are 100s more tools that were created for specific needs that may be available but unknown to investigators. Browsing GitHub or Sourceforge presents these tools in their current states, regardless of what that state may be (e.g., active, abandoned, alpha, beta).

Due to the vast amount of network data, the varied types of data, and speed at which that data traverses the network, network forensic visualization is a big data problem. Originally dealing with complex Homeland security issues in the United States, [14] coined the term visual analytics as the science of analytical reasoning facilitated by interactive visual interfaces. It combines human factors with machine analysis to aim at solving complex problems and is concerned with analytical process with interactive visual representations. Visual analytics is the interplay of data analysis, visualization, and interaction methods [15].

As seen in Wang, [16], this network data is temporal (e.g., timestamps in logs and packet captures), spatial (e.g., IP address, MAC addresses), and multivariate. Yet, traditional dimension reduction method trims out too many details about the data, leading to possible oversights by forensic investigators. [16] designed and implemented a NetFlowMatrix visual analytic system to provide a meaningful overview to identify potential port scans, data exfiltration, and server redirect events from over 7 million rows of network log data. Therefore, any approach needs to scale for effective and forensically sound processing of large-scale enterprise networks.

Additionally, a reliance on tools to aid the investigation may be disastrous if there is no digital evidence to follow. Overall, this project team is aware of the necessity for our developed toolkit, while operating on a large-scale network, to have a defined adherence to forensically sound models, processes, policies, and procedures in order to preserve the probative value of the data captured, analyzed, and presented as digital evidence in the judicial system.

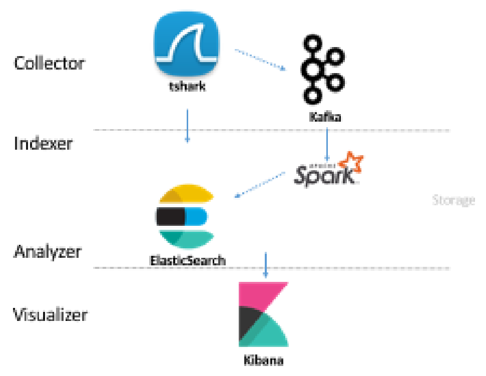


Fig. 1. FileTSAR Logical Architecture

III. FileTSAR Architecture

The objective of this project was to provide a platform for investigators to capture the necessary network traffic, analyze that traffic for probative content, maintain provenance of that content throughout the analysis process(es), and preserve admissibility of the evidence. The collection and preservation of evidence was based the Field Model described by [17]. A toolkit was assembled that provides a) an interface to selective capture data flows and analyze the contents of those flows, and b) a mechanism to reconstruct multiple types of data, such as documents (e.g. doc, docx, pdf), images (e.g. jpg, png, gif), email (based on SMTP, IMAP, and IMF), and VoIP sessions. This system was named the Toolkit for Selective Acquisition and Reconstruction of Files (FileTSAR).

The researchers aimed to create a tool that addressed the challenges as detailed by [5] faced by digital forensic examiners when investigating cases involving large-scale computer networks. This solution was implemented as a crash cart with all the necessary hardware, software, and system interconnections present. For an investigation, the fully contained system would be connected to the appropriate network interface(s) within the network to begin gathering necessary data through forensically sound processes.

A. Architecture and Implementation

In this implementation, the acquisition and analysis of data from the network are divided into several processes: 1) packet capture (i.e., recording the packet traffic on a network), 2) protocol parsing (i.e., parsing out the different network protocols and fields), 3) search and analysis, and 4) visualization (i.e., exploring the data). Based on the goals of the project, the architecture is divided according to functionality.

1) Collector: The Collector module captures network traffic and saves the traffic to the Storage implementation. The only interconnections to the large-scale network required for FileTSAR are via the Collector. The capturing interface is placed into promiscuous mode, allowing the network interface card to receive all packets on the network regardless of the destination of the packet. The engine is built using open source tools and custom code wrappers. This makes it compatible with existing incident response systems as well as provides a standardized interface into other modules in FileTSAR. While the tools focus on conducting live captures during active investigations, an additional set of functionalities designed into the toolkit can ingest previously captured data (PCAPs) from other systems already present in the enterprise network. This allows for the ingestion of data previously captured from the intrusion detection/prevention system, firewall, incident response system, and logs. The Collector can be initiated in one of four distinct modes of operation:

1) Time-based collection (specified start time, stop time, or duration)
2) Data-based collection (specified packet count or data size),
3) Unlimited collection (continue collection until stopped manually), and
4) Ingest existing PCAP file(s).

The Capture Engine operates two distinct components: Packet Capture and Flow Capture. The Packet Capture acquires packets directly on the network and then outputs them in PCAP format. This makes it compatible with existing incident response systems as well as provides a standardized interface into other modules in FileTSAR. The Packet Capture component provides detailed network information such as application or website names, whereas the Flow Capture is intended to collect and report statistics on network flows. Additionally, multiple Capture Engines can be operated in parallel throughout the investigation environment to be able to track flows of data at various specified points

2) Indexer: The Indexer module takes input from the Collector module and then processes it for file contents. The Parser included in the Indexer architecture has two distinct components: Packet Parser and Flow Parser. These two components handle the different types of data being processed. The Packet Parser takes each packet in the PCAP file. Raw packet data contains an extraordinarily large number of fields. While network traffic itself is sent in a binary format, each packet contains many different fields which can be parsed out into numbers, text, timestamps, and IP address, for example. The Packet Parser parses the packets into their respective Application layer, Transport layer, and Internetwork layer; subsequently, the data is archived into the active case directories within the Storage subsystem to be explored, searched, and visualized later. This output is formatted to JSON, which is well known industry accepted format. The Indexing Engine then constructs an optimized data structure or set of data structures for rapid and intensive retrieval and processing by the Analyzer module.

The Flow Parser works by identifying a group of packets sent over the same time period sharing common properties, such as the same source address, destination address, and protocol; that is, it identifies the packets that belong to the same flow. As each flow is unidirectional, if there is a corresponding flow, it is parsed and stored as a separate flow. Each flow is stored in the DFXML format. Additional data integrity is added to the indexing process by including hashing function output as part of the data structures. This provides validation of the data being stored and processed by ensuring that it remains the same throughout the investigation and indicating if any tampering of the data had occurred throughout the digital forensic investigation.

3) Analyzer: One of the major capabilities of our implementation is the analyzer. Being able to look into every piece of metadata and payload provides useful visibility for the investigator and helps to monitor systems, detect anomalies, and detect attackers. The Analyzer module performs the critical functions of identifying interrelatedness of files, flows, packets, users, timelines, etc. The Selector Engine in Analyzer receives input from the Selector Agent in the Visualizer module to further refine the dataset being analyzed by the forensic investigator. This dataset, or datasets, are then passed back to the Collector for rapid search, selection, identification, and relating of individual data points into a collection of resources for further investigation. The Analyzer feeds this data to the Visualizer for presentation capabilities. An additional set of functionality in Analyzer module is to selectively reconstruct multiple types of data, including documents, images, email, voice over IP conversations, and messaging. FileTSAR is capable of indexing and parsing the following protocols: HTTP, FTP, SMTP, IMAP, IMF, SIP and RTP:

1) HTTP: Open-source tools and code wrappers are used to reconstruct data from a packet capture. Open-source file carving tools are then used to carve data into common file types.
2) FTP: Open-source tools and code wrappers are used to reconstruct data from a packet capture. FTP data follows the same process as HTTP with classifications based on file carving.
3) Email (SMTP, IMAP, IMF): As with the previous protocols, reconstruction and file carving tools are used on the PCAP file. Based on the MIME headers, the email content (text and attachments if any) are decoded from base64 encoding.
4) VoIP: The SIP session is parsed from a flow and the voice payload (RTP) is decoded as G.711 or GSM and then converted to an MP3 file for storage.

4) Visualizer: The Visualizer module provides the interface for the digital forensic investigator to identify trends, patterns, or repetitions. The visualization capabilities are combined in an authenticated, web-based dashboard to provide dynamic environment to select and view reconstructed files and file fragments, for example. Only authenticated users are allowed to access the Visualizer module. This authentication requirement provides an additional level of legitimacy and confidentiality of the investigation through closure. Users authenticated via the visualizer can create a task with the interface. The Visualization module is used to start the collection process via the Collector, as well as to view the analysis of the data provided via the Indexer and Analyzer. The front end of the system that investigators use to interact with FileTSAR is handled by the visualizer.

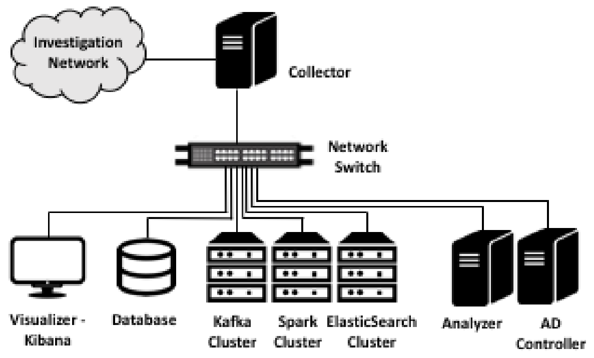
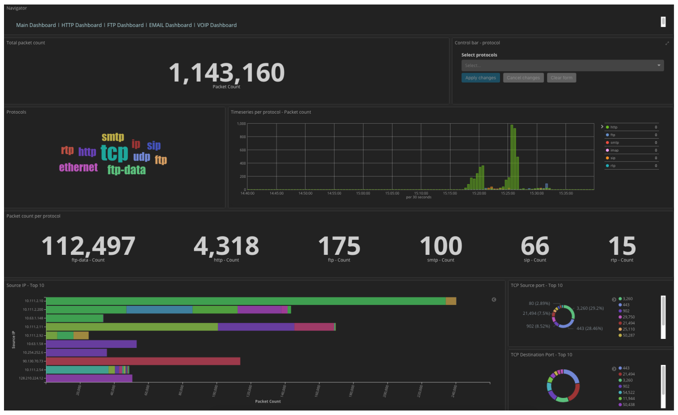


Fig. 2. FileTSAR Architecture

In order to start the collection process, or to visualize the collected information, the Visualizer links the Indexer and Analyzer modules to the web front end, which allows for examiners to actively look through the data. The web application interface of the toolkit contains a dashboard to notify the user of the respective statuses of the collector, indexer, and analyzer. The visualizer is built on an open source analytics and visualization platform and is embedded in the interface for the user to analyze network traffic. The reconstructed files from the PCAPs are also displayed on the dashboard. The dashboard is dynamic so as to allow specific visualization modules to be loaded, unloaded, or added to provide the specific environment needed by the digital forensic examiner. This also provides the utilities to allow investigators to open/close the investigation, identify and list files of probative value, itemize, and detail each step of the investigation for admissibility to the courts.

These capabilities are implemented using a collection of open-source tools along with custom code wrappers written Python 3 to integrate the data. The primary Collector functionality is implemented using tshark, which is responsible for packaging the PCAP and sending data to the Indexer. The Indexer functionality is implemented in Apache Kafka, Spark, and ElasticSearch. Data flows from Kafka as JSON into Spark, which itemizes each packet and flow. Once data is populated into ElasticSearch, the Analyzer functionality is implemented with tcpflow and additional ElasticSearch functionality. This allows the Visualizer to present, sort, and select the data to move from probative information to identify evidence. The Visualizer is implemented with Kibana, and a generalized view of the dashboard is shown in Fig. 3.

5) Storage: This module is the most generic component of the architecture, designed to provide a secure, robust, flexible, and scalable storage environment for all of the data within the File TSAR system. A critical aspect to the security of the Storage module is to provide non-repudiation for all filesystem changes. This was accomplished by only using filesystems capable of journaling, such as ZFS, and enabling this functionality within the kernel. All system data is stored and logged into the Storage module for inclusion in the investigation.



Main Dashboard | HTTP Dashboard | FTP Dashboard | EMAIL Dashboard | VOIP Dashboard
1,143,160
smtp
ip
sip
rtp
http
tcp
udp
ftp
ethernet
ftp-data
112,497
4,318
175
100
66
15

Fig. 3. FileTSAR Investigation Dashboard - Network Traffic Analysis

This data includes the output from the Collector as well as output from the Indexer module, processing information from the Analyzer module, and authentication information from the Visualizer module. Also, any data previously captured from the intrusion detection/prevention systems, firewall, incident response systems, logs, etc. from the local staff can be included and cataloged in the Storage module for evaluation of its probative value.

In order to start the collection process, the Visualizer links the web application front-end to the FileTSAR backend infrastructure. The options selected for collection by the examiner are forwarded as parameters to the Collector to begin the investigation.

IV. Functionality

The overall functionality of the FileTSAR toolkit had several iterations of analysis prior to being released. This involved local law enforcement officers, forensic investigators at national organizations, and others. The analysis and details provided by these officers and investigators led to the detailed improvements that resulting in a specific training session for forensic investigators from across the US. Additionally, the validation of this functionality was proven with multiple investigations containing upwards of 100GB of network traffic, and 1000s of files for reconstruction.

A. Process Validation

For 30 days, a beta test was conducted to gain detailed feedback and suggestions on FileTSARs feature and functionality implementations. Digital forensic examiners from the Tippecanoe County High Tech Crime Unit and the National White Collar Crime Center were given remote access to the toolkit along with a user manual that provided details on functionality and usage. These examiners represented the broad range of knowledge in the area of network forensic investigations, from entry-level, or basic training, to advanced training and knowledge in the area. The research team incorporated a ticketing system in order to provide technical support to the examiners during testing. All tickets for bugs and requests were logged and no changes were made until after the testing phase concluded.

The examiners were provided with an anonymous link to a web-based survey in order to provide feedback on the toolkit as they experienced the features and functionality. The survey was comprised of three sections: bug reports, feature requests, and overall feedback. The examiners were able to provide feature requests and bug reports for the following categories: setup and login, user interface, capture options, status management, dashboard, and final reconstruction. In addition, the anonymous survey included open-ended questions to which the examiners were able to provide additional feedback and/or comments.

Based on the feedback and suggestions provided, the examiners were satisfied with the core functionalities, such as analyzing network data via the Visualizer and the reconstruction of files, emails, and voice over IP sessions. The survey results also showed support for the manner in which the volume of data was handled as well as the level of granularity that could be attained in the data provided. In addition, the examiners made suggestions on the user interface and user experience components that would ultimately increase user-friendliness for future versions of FileTSAR, such as a mouseover feature to obtain more information about the item. The research team revised FileTSAR based on the feedback provided during beta-testing.

A 3-day law enforcement training workshop was hosted at Purdue University for FileTSAR from September 11-13, 2018. We recruited from law enforcement agencies across the United States to fill 20 seats at the training workshop. All travel expenses were covered and this 3-day training workshop was free to law enforcement thanks to federal funding from the NIJ. The FileTSAR training workshop was open to sworn law enforcement officers who are permanent residents in the United States. Interested digital forensic examiners applied through an event page, and answered questions regarding their agency size, current job duties, and level of knowledge in network forensics and basic digital forensics.

A week prior to the training workshop, 20 sworn, certified digital forensic examiners from across the United Sates (Arizona, California, Indiana, Illinois, Pennsylvania, North Carolina, South Carolina, Tennessee, Texas, and West Virginia) confirmed their registration for the FileTSAR training workshop. However, there were 3 last-minute cancellations due to casework and 1 due to Hurricane Florence. Sixteen (16) digital forensic examiners participated in the FileTSAR training from across the United States (Arizona, California, Indiana, Illinois, North Carolina, Tennessee, Texas, and West Virginia) and from different agencies (local, state, and federal law enforcement). For the attendees, their years of experience as a digital forensic examiner ranged from less than a year to over 10 years; the most common response was 3-6 years of experience (44%). More than half (56%) of the participants reported some level of college. Finally, only four of the respondents reported that they did not have any certifications or training in network forensics.

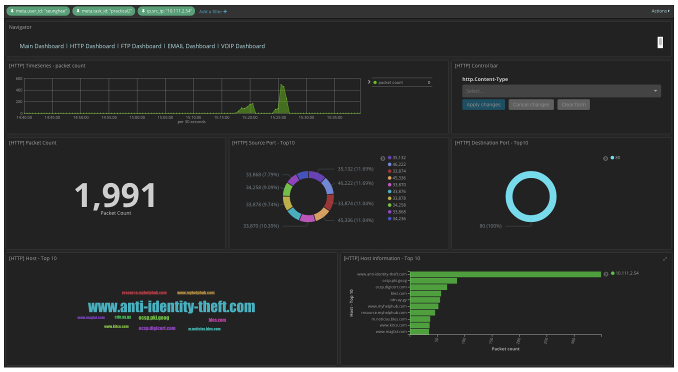


Fig. 4. FileTSAR Dashboard - Follow Individual IP Source

The training workshop provided a general overview on networks and protocols as well as the functionalities of FileTSAR (data capture, analysis and visualization, file reconstruction, and reporting). For the labs, the examiners were provided with a PCAP of a fictional criminal case so they could work through the different functionalities of FileTSAR. On the 3rd day, the examiners completed a practical exam testing their use of FileTSAR on a new PCAP for a different fictional criminal case (IRB Protocol #1809021014). For the final exam scenario, the average exam score was 88.8% regarding the items related to using FileTSAR to investigate the test scenario. Overall, there was wide support for FileTSAR with large-scale investigations as well as an interest in a scaled-down version for smaller agencies with storage, budget, and back-end support limitations.

B. *Reconstruction Validation*

The system was deployed using VMWare ESXi to host the virtual machines that made up the components described in Section III. The underlying hardware that was utilized was three custom-build servers utilizing dual Intel Xeon E5-2620 CPUs (10 core, 2.1GHz), and 256 GB DDR4 RAM per system. A local SAN was constructed using Intel Xeon E-52603 CPUs (6 core, 1.7GHz), 32 GB DDR2 RAM, and eight 900 GB 10K SAS hard disk.

To validate the large-scale capabilities of the toolkit, we conducted a "stress test" of the system using approximately 123,500,000 packets from a collection of PCAP files totaling nearly 100GB. In our test, FileTSAR reconstructed all sessions containing HTTP, FTP, SMTP, and IMF data; note that SIP and RTP data were not present in this test. This reconstruction process parsed and carved approximately 3,000,000 files from the overall dataset. Each reconstruction protocol was executed independently to evaluate its efficacy and effectiveness. During each reconstruction, CPU utilization peaked at 25%, and the average RAM usage was 30.9 GB. A rate of 91.8 files/sec was the average over the course of each test. As an example of the dashboard for specifically the HTTP reconstruction, Fig. Fig. 4 contains the total HTTP packets counts for this particular test – 1,143,160 – along with the top viewed websites, the time the websites were visited, and graphs indicating top source and destination IP addresses within the selected flows.

This reconstruction rate is of note, as this required the assembly of flows across multiple PCAP files to be able to reconstruct all of the files. After each file is reconstructed, it is written to disk in an evidence directory specific to each case. Both of these processes are substantial I/O operations, as opposed to processing operations. This additional I/O overhead to the SAN could indicate an area of improvement for system performance. There are additional processing operations including the MD5-hashing of each carved file for integrity purposes as part of the chain of custody of evidence.

V. Conclusion

As previously discussed, there are many challenges in network forensic investigations. As the volume of data created and subsequently sent over a network continues to grow, these challenges continue to increase as well, i.e., the so-called Big Data problems. Many of the current tools potentially used by investigators have internal challenges that lead to a potential lack of admissibility of evidence should the case go to a court of law. FileTSAR solves many of these issues by creating wrappers around well-known functions and maintained tools, utilizing the Field Triage process, an accepted process flow for forensic evidence collection and preservation, and architecting the system to handle large data sets. Additionally, all the data at each stage of processing is stored in well-known (and forensically viable) formats, and contents are cryptographically hashed to ensure integrity of the data at rest.

There are several avenues of future work to be considered for this project. During the training workshop, a brainstorming session on deployments resulted in a request for a variety of deployment models that required fewer computing resources or are more easily deployed in a capture-only mode. This would allow for a Collector to be deployed to capture and store network information to be analyzed later in the investigation laboratory. Additionally, further optimization on the system could provide for more selective options for reconstruction of files (such as only specific file types, from specific hosts, during a specific timeframe, and delivered via a specific protocol). This then necessitates a more robust reporting mechanism that would allow investigation reports to be created in formats other than PDF format. Another item requested by the digital forensic examiners was the ability to integrate FileTSAR reports into other investigative tools, such as FTK and Griffeye.

VI. Acknowledgements

The authors wish to thank Trent Epperson, Nicolas Vukadinovic, Xiang Liu, and Kelsey Billups for their contribution to this project.

References

[1] R. Clifford, *Cybercrime: The Investigation, Prosecution and Defense of a Computer-related Crime*. Carolina Academic Press, 2011. [Online]. Available: https://books.google.com/books?id=w6ZfzgAACAAJ

[2] V. S. Harichandran, F. Breitinger, I. Baggili, and A. Marrington, "A cyber forensics needs analysis survey: Revisiting the domain's needs a decade later," *Computers & Security*, vol. 57, pp. 1–13, 2016.

[3] A. Bossler, T. J. Holt, and K. C. Seigfried-Spellar, *Cybercrime and digital forensics: An introduction*. Routledge, 2017.

[4] M. K. Rogers and K. C. Seigfried-Spellar, “The future of computer forensics: a needs analysis survey,” *Computers & Security*, vol. 23, no. 1, pp. 12–16, 2004.

[5] S. E. Goodison, R. C. Davis, and B. A. Jackson, “Digital evidence and the us criminal justice system,” *Identifying Technology and Other Needs to More Effectively Acquire and Utilize Digital Evidence. Priority Criminal Justice Needs Initiative. Rand Corporation*, 2015.

[6] M. K. Rogers and K. C. Seigfried-Spellar, “The curse of big data in digital forensics,” *Interdisciplinary Conference on Cybercrime*, 2016.

[7] D. Reinsel, J. Gantz, and J. Rydning, “Data age 2025: The evolution of data to life-critical,” *Dont Focus on Big Data*, 2017.

[8] Cisco. (2017) The zettabyte era: Trends and analysis. Cisco. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html

[9] S. Garfinkel, “Network forensics: Tapping the internet,” *IEEE Internet Computing*, vol. 6, pp. 60–66, 2002.

[10] E. S. Pilli, R. C. Joshi, and R. Niyogi, “Network forensic frameworks: Survey and research challenges,” *digital investigation*, vol. 7, no. 1-2, pp. 14–27, 2010.

[11] C. Beek. (2016) Introduction to file carving. McAfee. [Online]. Available: http://fliphtml5.com/zmxx/prcw/basic

[12] B. Carrier, *File system forensic analysis*. Addison-Wesley Professional, 2005.

[13] B. Claise, B. Trammell, and P. Aitken, “Specification of the ip flow information export (ipfix) protocol for the exchange of flow information,” IETF, Tech. Rep., 2013.

[14] K. A. Cook and J. J. Thomas, “Illuminating the path: The research and development agenda for visual analytics,” Pacific Northwest National Lab.(PNNL), Richland, WA (United States), Tech. Rep., 2005.

[15] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, “Visual analytics: Scope and challenges,” in *Visual data mining*. Springer, 2008, pp. 76–90.

[16] W. Wang, B. Yang, and Y. Chen, “A visual analytics based approach on identifying server redirections and data exfiltration,” in *Proceedings of the 16th Annual Information Security Symposium*. CERIAS-Purdue University, 2015, p. 20.

[17] M. Rogers, J. Goldman, R. Mislan, T. Wedge, and S. Debrota, “Computer forensics field triage process model,” *Journal of Digital Forensics, Security, and Law*, vol. 1(2), pp. 19–38, 2006.

File Toolkit for Selective Analysis & Reconstruction (FileTSAR) for Large Scale Computer Networks

Installation and Deployment Guide
Software Version: 1.0.0

April 2019

Table of Contents

1 Introduction

1.1 Purpose

This toolkit, named the Toolkit for Selective Analysis and Reconstruction of Files (FileTSAR), is designed to follow the Computer Forensic Field Triage Process Model (Rogers et al., 2006) for on-scene acquisition of probative network data (AKA Digital Evidence). The system can be installed and run on any appropriate network interface(s) within a large-scale network to begin gathering necessary data from the network through forensically sound processes. The main functions of this toolkit are to capture data flows and provide a mechanism to selectively reconstruct multiple types of data, including documents, images, email, VoIP conversations, and messaging.

The Installation guide provides a detailed description of each part of the FileTSAR infrastructure and its setup. By following the guidelines outlined in this documents, administrators and investigators can set up their instance of FileTSAR for network investigations. Readers will find both hardware and software requirements required to run the toolkit at optimal functionality. For all needed hardware, the roles, features, packages, requirements, configurations, and default setup are described; software descriptions include prerequisites, installation steps, and configured values.

1.2 Intended audience and reading suggestions

The Installation and Deployment Guide is a technical document assumes a certain pre-requisite knowledge. While it would be possible for anyone to follow this manual and online resources to set it up, someone with fundamental networking knowledge is recommended, to make the setup process as simple and convenient as possible. It is intended that stakeholders and software support personnel can read this document and coordinate their efforts in the installation/deployment of the application.

This document contains a description of each module, a server configuration, and the installation procedure. Also, the specific versions of the packages and libraries that need to be installed for each module are provided. Note that other versions of libraries or packages may be used but this does not guarantee a successful installation. Additionally, you can visit the following websites to read more about the technology used in the FileTSAR manual:

VMware ESXi https://www.vmware.com/products/esxi-and-esx.html

Wireshark https://www.wireshark.org/

Apache Spark https://spark.apache.org/

Apache Kafka https://kafka.apache.org/

Elasticsearch https://www.elastic.co/

1.3 System Architecture

This physical architecture is a representation of the structure of the physical elements of the FileTSAR system. Figure 1 depicts the actual architecture for the FileTSAR system. For actual operation, the FileTSAR system uses two physical hosts and multiple virtual machines (guest operating systems or application environments) that run on the physical machines as shown in Figure 1. The server and host machines described in this guide are hosted virtually for the FileTSAR setup. The entire network it set up over a 10.x.x.0/24 range. All the virtual machines are hosted on 3 ESXi servers. The storage on the server is managed through FreeNAS 9.10.2-u2. The virtual machines are managed through vCenter Server. Each of the individual configurations of the virtual machines used in FileTSAR is described below.

In a virtual environment, the architecture is composed of a Kafka & Zookeeper cluster, an Elasticsearch cluster, a Spark cluster, a collector, a web server, an analyzer, and a database, and installation guides for each module are provided in the following chapters. Note that the number of physical host and the number of machines that consist clusters can be configured differently for better performance or availability.

The FileTSAR system uses the Network File System (NFS), which allows the server to share directories and files over a network as shown in Figure 2. Therefore, NFS must be set up and the resource (file or directory) must be available from each server prior to the server installations (NOTE: This installation guide does not provide a procedure to set up and configure NFS).

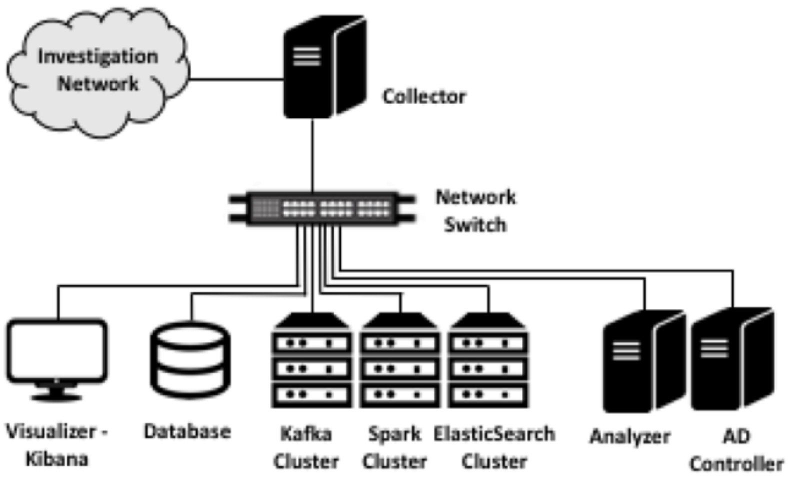


Figure 1. FileTSAR System Physical Architecture

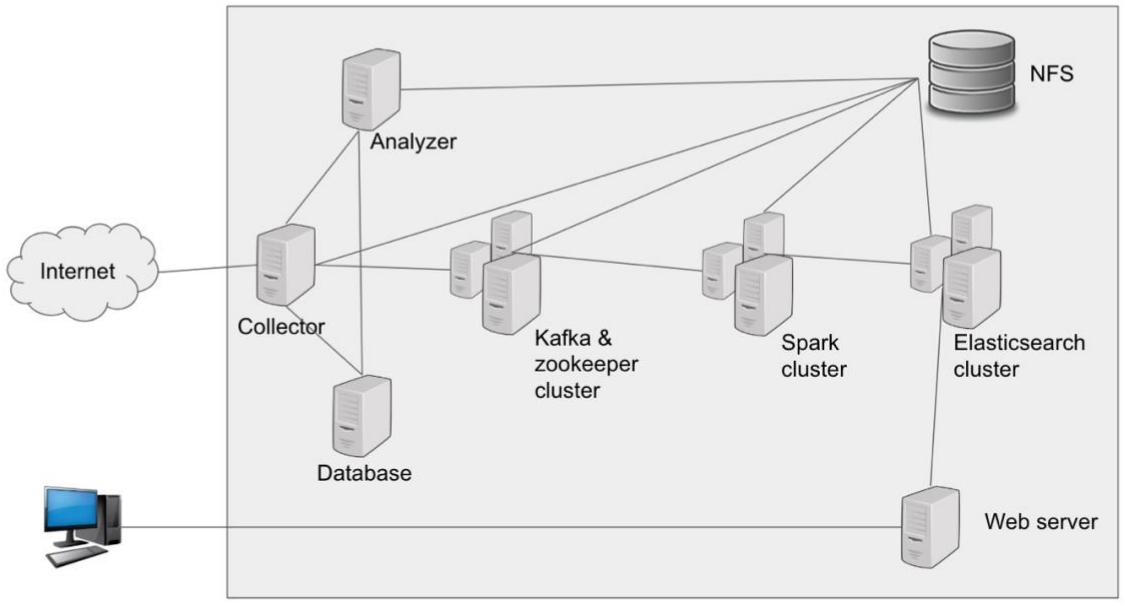


Figure 2. FileTSAR System Network Diagram

Lastly, the FileTSAR system has two types of users: (1) Super user and (2) Regular user as shown in Figure 3. The super user is a special user account used for system administration and they can capture network traffic and have permission to change settings. The Regular user can analyze the existing tasks and generate reports, but they are not given the ability to capture traffic and change settings.

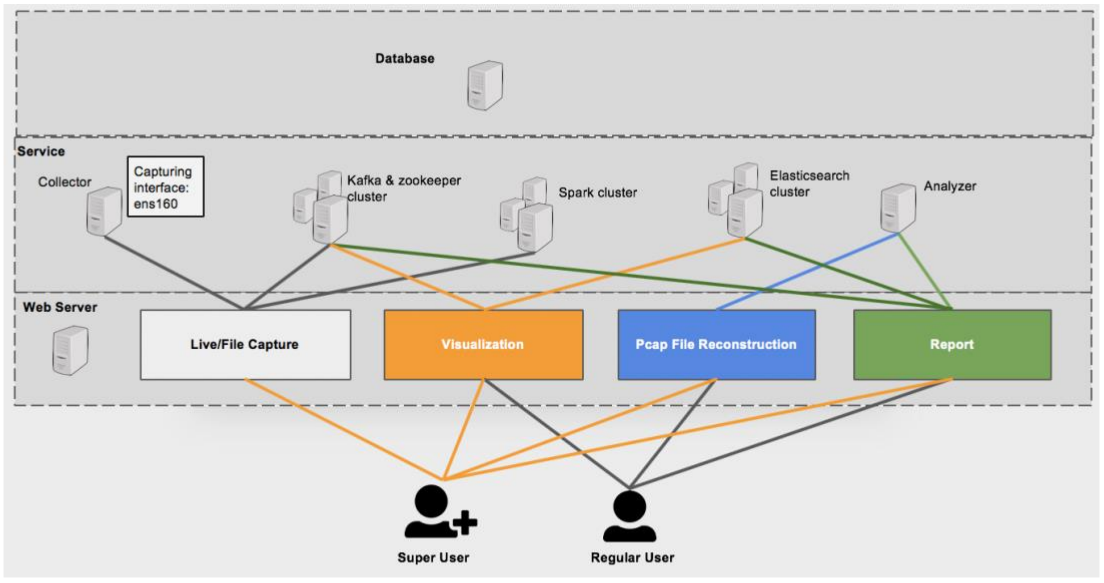


Figure 3. FileTSAR System Logic Architecture

2 Server 1 (Database)

Installation of this system is supported on the following operating system and version:

- Linux Ubuntu 16.04 (64-bit)

2.1.1 Roles, Features, and Packages

Roles

The role of the database server is to manage the user information and the metadata of the created tasks and files.

Packages

The following software package must be installed on the operating system prior to installation of the software:

- MySQL Server (Version 5.7.25)

2.1.2 Configuration

Authentication

Basic authentication should be enabled

2.1.3 Configured Values

Use the table below to make note of the values for your installation environment for future reference.

Information	Value
Server name	database

2.2 Server 2 (Kafka System)

Installation of this system is supported on the following operating system and version:

- Linux Ubuntu 16.04 (64-bit)

2.2.1 Roles, Features, and Packages

Roles

The role of Kafka system is to buffer unprocessed packet data coming from the collector and pipe it to Spark cluster for further processing.

Features

Kafka is a distributed, multi-partition, multi-copy message queue system. For actual operation, Kafka is run as a cluster on multiple servers that can span multiple datacenters. This installation guide provides how to set up a Kafka cluster with three Kafka servers, but more nodes can be added for better performance.

Packages

The following software packages must be installed on the operating system of each instance:

- Zookeeper (Version: 3.4.10)
- Kafka (Version: 2.11-1.1.0)

2.2.2 Hardware Requirements

When it comes to deploying Kafka to production, there are a few recommendations that you should consider. For running Kafka in production mode, please refer to the following URL

- https://docs.confluent.io/current/kafka/deployment.html

2.2.3 Configured Values

Use the table below to make note of the values for your installation environment for future reference.

Information	Value
Kafka Server1 Name	Kafka1
Kafka Server2 Name	Kafka2
Kafka Server3 Name	Kafka3
Zookeeper Server Name	Zookeeper1

2.3 Server 3 (Collector)

Installation of this system is supported on the following operating system and version:

- Linux Ubuntu 16.04 (64-bit)

2.3.1 Roles, Features, and Packages

Roles

The role of the collector server is to capture and save network traffic in pcap format and send the captured packet to Kafka server for further processing and analysis.

Packages

The following software packages must be installed on the operating system prior to installation of the software:

- Wireshark/Tshark (Version: 2.6.6)
- Python 3.6.0
- Libpcap 1.7.4-2

2.3.2 Configured Values

Use the table below to make note of the values for your installation environment for future reference.

Information	Value
Server name	collector

2.4 Server 4 (Elasticsearch)

Installation of this system is supported on the following operating system and version:

- Linux Ubuntu 16.04

2.4.1 Roles, Features, and Packages

Roles

The role of Elasticsearch is to store processed data in JSON format for analysis and visualization.

Features

Elasticsearch is a distributed, RESTful search and analytics engine based on the Lucene library. It provides a distributed multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. For actual operation, Elasticsearch is run as a cluster on multiple servers, consisting of master nodes, data nodes, and coordinating nodes. For more detail, please visit the Elasticsearch official website (https://www.elastic.co/guide/en/elasticsearch/reference/6.2/modules-node.html). This installation guide provides how to set up an Elasticsearch cluster with one master node, three data nodes, and one coordinating node. However, it becomes important to consider more nodes into a cluster for better performance.

Packages

The following software packages must be installed on the operating system prior to installation of the software:

- Elasticsearch (Version: 6.2.4); Lucene version: 7.2.1

2.4.2 Hardware Requirements

For sorting and aggregation operations it is required that Elasticsearch must have enough memory to process. For reliable processing of Elasticsearch **RAM** should be allocated from **16 GB to up to 64 GB**. Elasticsearch uses heap and OS cache for better performance.

- Random Access Memory (RAM): Minimum 16GB, Maximum: 64 GB
- CPU Cores for each Cluster: Minimum 2 Cores, Up to 8 Cores
- Disk Size: It should depend on number of cluster and data size

2.4.3 Configured Values

Use the table below to make note of the values for your installation environment for future reference.

Information	Value
Server name	elasticsearch-master

	elasticsearch-data-1
	elasticsearch-data-2
	elasticsearch-data-3
	elasticsearch-coordinator

2.5 Server 5 (Spark)

Installation of this system is supported on the following operation systems and versions:

- Linux Ubuntu 16.04

2.5.1 Roles, Features, and Packages

Roles

The role of Spark system in FileTSAR is to parse, analyze, and index packet information to Elasticsearch.

Features

The FileTSAR system uses a Spark cluster consisting of several master and slave nodes on standalone mode. However, other cluster managers like Apache Mesos and Hadoop YARN can be used. This installation guide provides information on how to set up a spark cluster with one master node and three slave nodes. However, it is important to consider adding more nodes into a cluster for better performance.

Packages

The following software packages must be installed on the operating system prior to installation of the software:

- Java JDK 8
- Spark (Version: 2.3.0)

2.5.2 Hardware Requirements

When it comes to deploying Spark to production, there are a few recommendations that you should consider. For running Spark in production mode, please refer to the following URL

- https://spark.apache.org/docs/2.3.0/hardware-provisioning.html

2.5.3 Configuration

The spark-submit script can load default Spark configuration values from a properties file and pass them on to your application. By default, it will read options from conf/spark-defaults.conf in the Spark directory. For more detail, see the section on loading default configurations.

2.5.4 Configured Values

Use the table below to make note of the values for your installation environment for future reference.

Information	Value
Server name	Spark-master
	Spark-slave1
	Spark-slave2
	Spark-slave3

2.6 Server 6 (Analyzer)

Installation of this system is supported on the following operating system and version:

- Linux Ubuntu 16.04

2.6.1 Roles, Features, and Packages

Roles

The role of Analyzer is to analyze and reconstruct pcap files captured by the collector or uploaded by users.

Features

- Analysis of pcap files
- Extract and reconstruct files transferred via HTTP, FTP, SMTP, IMAP, SIP and RTP protocols.
- Report of the analyzed tasks

Packages

The following software packages must be installed on the operating system prior to installation of the software:

- wireshark==2.6.4
- python==2.7.12
- pip==18.1
- amqp==2.3.2
- bcrypt==3.1.5
- billiard==3.5.0.5
- blinker==1.4
- celery==4.2.1
- certifi==2018.11.29
- cffi==1.11.5
- chardet==3.0.4
- Click==7.0

- elasticsearch==6.3.1
- Flask==1.0.2
- Flask-Assets==0.12
- Flask-Bcrypt==0.7.1
- Flask-Caching==1.4.0
- Flask-CeleryExt==0.3.1
- Flask-DebugToolbar==0.10.1
- Flask-Login==0.4.1
- Flask-Mail==0.9.1
- flask-paginate==0.5.1
- Flask-Script==2.0.6
- Flask-SQLAlchemy==2.3.2
- Flask-Uploads==0.2.1
- Flask-WTF==0.14.2
- idna==2.8
- itsdangerous==1.1.0
- Jinja2==2.10
- kombu==4.2.2
- MarkupSafe==1.1.0
- Pillow==5.3.0
- pycparser==2.19
- PyMySQL==0.9.3
- pytz==2018.7
- reportlab==3.5.12
- requests==2.21.0
- six==1.12.0
- SQLAlchemy==1.2.15
- urllib3==1.24.1
- vine==1.1.4
- webassets==0.12.1
- Werkzeug==0.14.1
- WTForms==2.2.1

2.6.2 Hardware Requirements

Random access memory (RAM): 32GB

2.6.3 Configured Values

Use the table below to make note of the values for your installation environment for future reference.

Information	Value
Server name	Analyzer

2.7 Server 7 (Kibana)

2.7.1 Roles, Features, and Packages

Roles

The role of Kibana is to provide visualization and dashboard for the analyzed tasks.

Features

- Visualization of data stored in Elasticsearch
- User can create various visualization components such as histograms, graphs, pie charts, and so on.

Packages

The following software packages must be installed on the operating system prior to installation of the software:

- Kibana (version: 6.2.4)

2.7.2 Configuration

How you deploy Kibana largely depends of the use case. If you have a large number of heavy Kibana users, you might need to load balance across multiple Kibana instances that are all connected to the same Elasticsearch instance. It is also recommended to run Kibana separate from the Elasticsearch data or master nodes. For Kibana in a production environment, please visit the Kibana official website (https://www.elastic.co/guide/en/kibana/6.2/production.html)

2.7.3 Configured Values

Use the table below to make note of the values for your installation environment for future reference.

Information	Value
Server name	Kibana

2.8 Server 8 (Web Server)

Installation of this system is supported on the following operating system and version:

- Linux Ubuntu 16.04

2.8.1 Roles, Features, and Packages

Roles

The Web server provides a web platform for users to 1) initiate live captures; 2) upload pcap files to analyze; 3) analyze and reconstruct pcap files generated from live captures or uploaded by users; 4) summarize the pcap file and generate a report

Features

Web Server sends a request to activate the following tasks:

- Time based live captures
- Size based live pcatures
- Non-stop live captures
- Pcap file uploading
- Capture list
- Dashboard for captures
- Analysis for captures
- Reports for pcap files

Packages

The following software packages must be installed on the operating system prior to installation of the software:

- wireshark==2.6.4
- python==2.7.12
- pip=18.1
- amqp==2.3.2
- bcrypt==3.1.5
- billiard==3.5.0.5
- blinker==1.4
- celery==4.2.1
- certifi==2018.11.29
- cffi==1.11.5
- chardet==3.0.4
- Click==7.0
- elasticsearch==6.3.1
- Flask==1.0.2
- Flask-Assets==0.12

- Flask-Bcrypt==0.7.1
- Flask-Caching==1.4.0
- Flask-CeleryExt==0.3.1
- Flask-DebugToolbar==0.10.1
- Flask-Login==0.4.1
- Flask-Mail==0.9.1
- flask-paginate==0.5.1
- Flask-Script==2.0.6
- Flask-SQLAlchemy==2.3.2
- Flask-Uploads==0.2.1
- Flask-WTF==0.14.2
- idna==2.8
- itsdangerous==1.1.0
- Jinja2==2.10
- kombu==4.2.2
- MarkupSafe==1.1.0
- Pillow==5.3.0
- pycparser==2.19
- PyMySQL==0.9.3
- pytz==2018.7
- reportlab==3.5.12
- requests==2.21.0
- six==1.12.0
- SQLAlchemy==1.2.15
- urllib3==1.24.1
- vine==1.1.4
- webassets==0.12.1
- Werkzeug==0.14.1
- WTForms==2.2.1

2.8.2 Hardware Requirements

Random access memory (RAM): 32GB

2.8.3 Configured Values

Use the table below to make note of the values for your installation environment for future reference.

Information	Value
Server name	WebServer

3 Software Installation

3.1 Server 1 (Database)

3.1.1 Prerequisites

All steps in section 2 “Server Configurations” have been performed.

3.1.2 Installation Steps

1. Install MySQL Database (version: 5.7.25)

```
$ sudo apt-get update
$ sudo apt-get install mysql-server-5.7
```

2. Testing MySQL service: Regardless of how you installed it, MySQL should have started running automatically. To test this, check its status using the following command.

```
$ systemctl status mysql.service
```

3. At the command line, log into MySQL as the root user

```
$ mysql -u root -p
```

4. Create a new database named ‘filetsar’ and change the database to use it

```
mysql > create database filetsar;
mysql > use filetsar;
```

5. Create the following tables using InnoDB

```
mysql > create table userinfo (
        user_id char(20) not null,
        password varchar(255) not null,
        primary key(user_id)
) ENGINE=InnoDB;
```

```
mysql > create table task (
        user_id char(20) not null,
        task_id varchar(255) not null,
        year YEAR not null,
        case_number SMALLINT unsigned not null DEFAULT 1,
        option_number int(1) not null,
        capture_interface varchar(50),
        autostop_duration int(11),
        autostop_filesize bigint(20),
        uploaded_file_path varchar(255),
        protocol_selection varchar(100),
        timestamp varchar(100) not null,
        capture_status tinyint(1) not null,
        isanalyzed tinyint(1) not null,
        primary key(user_id, task_id),
        foreign key(user_id) references userinfo(user_id) on update cascade on delete cascade
) ENGINE=InnoDB;
```

```
mysql > create table pcap (
            filename char(20) not null,
            user_id char(20) not null,
            task_id varchar(255) not null,
            sha1 binary(20),
            md5 binary(16),
            foreign key (user_id, task_id) references (user_id, task_id) on update cascade on delete
cascade
) ENGINE=InnoDB;
```

```
mysql > create table reconstructedFile (
            ID int not null AUTO_INCREMENT,
            filename varchar(255) not null,
            user_id char(20) not null,
            task_id varchar(255) not null,
            sha1 binary(20),
            md5 binary(16),
            protocol char(20) not null,
            primary key (ID),
            foreign key (user_id, task_id) references (user_id, task_id) on update cascade on delete
cascade
) ENGINE=InnoDB;
```

6. Create a user and a strong password as you noted in section 2.1.1 and grant access to the user

```
mysql > grant all previlieges on filetsar.* to ‘filetsar_user@%’ identified by ‘password123456’ with
grant option;

mysql > flush privileges;
```

7. Fill in the database configuration information in the system configuration file (config.yml) located on NFS and save the configuration file

3.1.3 Configured Values

Use the table below to make note of the values for your installation environment for future reference.

Information	Value
User account ID	(example) filetsar_user
Account password	(example) Password123456

3.2 Server 2 (Kafka System)

3.2.1 Prerequisites

All steps in section 2 “Server Configurations” have been performed.
All steps in section 3.1 “Software installation, Server 1 (Database)” have been performed.

3.2.2 Installation Steps

1. Install Java: Kafka is written in Java and Scala and requires JRE 1.7 and above to run it. First, we will ensure Java is installed.

```
$ sudo apt-get update
$ sudo apt-get install default-jre
```

2. Install ZooKeeper; ZooKeeper is a centralized service for maintaining configuration information, providing distributed synchronization, and providing group services. Kafka uses ZooKeeper so you need to first start a ZooKeeper server if you don’t already have one. Before starting the servers, you need to modify the configuration files. A replicated group of servers in the same application is called a quorum, and in replicated mode, all servers in the quorum have copies of the same configuration file. In this installation guide, we use the ZooKeeper that comes with Kafka package.

```
First, download kafak 2.11-1.1.0 release and un-tar it

$ tar -xzf kafka_2.11-1.1.0.tgz
$ cd kafka_2.11-1.1.0
```

3. Modify the configuration file (/config/zookeeper.properties) and add properties in the file properly. Here is an example.

Open file in /config/zookeeper file

```
tickTime=2000
dataDir=/var/zookeeper
clientPort=2181
initLimit=5
syncLimit=2
server.1=zoo1:2888:3888
server.2=zoo2:2888:3888
server.3=zoo3:2888:3888
```

4. Start Zookeeper Server

You can use the convenience script packages with Kafka to get a single-node ZooKeeper instance. After typing the command below, you will be connected to the Zookeeper server.

```
$ bin/zookeeper-server.start.sh config/zookeeper.properties
```

5. Start Kafka Server

Before starting the Kafka server in cluster mode, configuration setting should be performed for each server instance. In each server, when you open config/server.properties file, there are 3 properties that have to unique for each broker instance:

(1) broker.id = 0
(2) listerners=PLAINTEXT://:9092
(3) log.dirs=/mnt/kafka-logs

For each server, copy the config/server.properties file and make 3 files for each server instance, and change the above 3 properties for each copy of file so that they are unique.

Server.1.properties

```
broker.id=1
listeners=PLAINTEXT://:9093
log.dirs=/mnt/kafka-logs1
```

Server.2.properties

```
broker.id=2
listeners=PLAINTEXT://:9094
log.dirs=/mnt/kafka-logs2
```

Server.3.properties

```
broker.id=3
listeners=PLAINTEXT://:9095
log.dirs=/mnt/kafka-logs3
```

Also, create the log directories that we configured:

```
$ mkdir /mnt/kafka-logs1
$ mkdir /mnt/kafka-logs2
$ mkdir /mnt/kafka-logs3
```

6. Finally, we can start the broker instances. Run the below three commands on different server in different instances.

```
$ bin/kafka-server-start.sh config/server.1.properties
$ bin/kafka-server-start.sh config/server.2.properties
$ bin/kafka-server-start.sh config/server.3.properties
```

7. Create a topic named ‘packet-json’

   An example of creating a topic named packet-json with 20 partitions and 3 replication factors is as follows:

```
Bin/kafka-topic.sh –create –zookeeper kafka-server-address:2181 –replication-factor 3 –partition
20 –topic packet-json
```

8. Fill in the Zookeeper and Kafka configuration information in the configuration file (config.yml) located on NFS directory and save the file.

3.3 Server 3 (Collector)

3.3.1 Prerequisites

All steps in section 2 “Server Configurations” have been performed.
All steps in section 3.1 “Software Installation, Server 1 (Database)” have been performed.
All steps in section 3.2 “Software Installation, Server 2 (Kafka System)” have been performed.

3.3.2 Installation Steps

1. Open the configuration file (config.yml) and enter the information of the collector and save it
Download filetsar project

2. Run start-collector.sh with the argument of the path of the configuration file (config.yml)

```
$ ./start-collector.sh /path/config.yml
```

If it shows, “start collector…”, it is running successfully.

3.4 Server 4 (Elasticsearch)

3.4.1 Prerequisites

All steps in section 2 “Server Configurations” have been performed.
All steps in section 3.1 “Software Installation, Server 1 (Database)” have been performed.
All steps in section 3.2 “Software installation, Server 2 (Kafka System)” have been performed.
All steps in section 3.3 “Software installation, Server 3 (Collector)” have been performed.

3.4.2 Installation Steps

1. Install Java 8 in order to run
2. Install Elasticsearch with Debian Package (version 6.2.4)

```
$ sudo apt-get install apt-transport-https
$ echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | sudo tee -a
/etc/apt/sources.list.d/elastic-6.x.list
$ wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.2.4.deb
$ wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.2.4.deb.sha512
$ shasum -a 512 -c elasticsearch-6.2.4.deb.sha512
$ sudo dpkg -I elasticsearch-6.2.4.deb
```

3. Run Elasticsearch with system

```
$ sudo /bin/systemctl daemon-reload
$ sudo /bin/systemctl enable elasticsearch.service
```

4. Configure Elasticsearch nodes

Elasticsearch loads its configuration from the /etc/elasticsearch/elasticsearch.yml file by default, and you can modify the file to customize the configuration settings as desired for each node.

Master-eligible node

The master node is responsible for lightweight cluster-wide actions such as creating or deleting an index, tracking which nodes are part of the cluster, and deciding which shards to allocate to which nodes. It is important for cluster health to have a stable master node. A node that has node.master set to true, which makes it eligible to be elected as the master node, which controls the cluster.

To create a dedicated master-eligible node, set:

```
          node.master: true
          node.data: false
          node,ingest: false
          cluster.remote.connect: false
```

Data node

Data nodes hold the shards that contain the documents you have indexed. Data nodes handle data related operations like CRUD, search, and aggregations. These operations are I/O-, memory-, and CPU-intensive. It is important to monitor these resources and to add more data nodes if they are overloaded.

To create a data node, set:

```
node.master: false
node.data: true
node,ingest: false
cluster.remote.connect: false
```

Coordinating node

The coordinating node route requests handle the search reduce phase and distribute bulk indexing. Coordinating only nodes can benefit large clusters by offloading the coordinating node role from master-eligible nodes.

To create a coordinating node:

```
node.master: false
node.data: false
node,ingest: false
cluster.remote.connect: false
```

5. Index FileTSAR templates that allow you to define templates that will automatically be applied when new indices are created.

```
$ ./index-filetsar-template.sh
```

3.4.3 Configured Values

Information	Value
path.data	/mnt/elasticsearch/data (NFS directory)
Description: Every data and master-eligible node requires access to a data directory where shards and index and cluster metadata will be stored. The path of data can be configured through the elasticsearch.yml file	
path.log	/mnt/elasticsearch/log
cluster.name	filetsar-cluster
node.name	${HOSTNAME}
Network.host	_site_

For other parameters such as java heap size, please refer to https://www.elastic.co/guide/en/elasticsearch/reference/6.2/setup.html

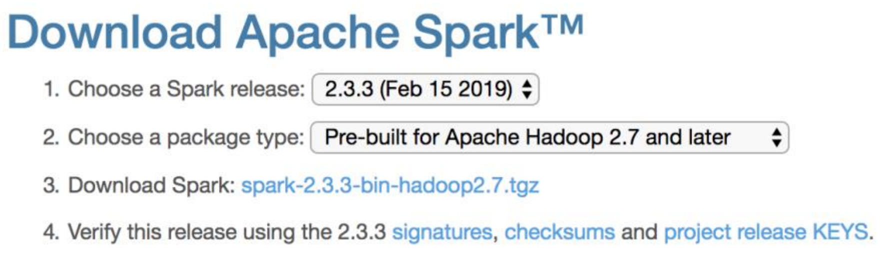
3.5 Server 5 (Spark)

3.5.1 Prerequisites

All steps in section 2 “Server Configurations” have been performed.
All steps in section 3.1 “Software Installation, Server 1 (Database)” have been performed.
All steps in section 3.2 “Software installation, Server 2 (Kafka System)” have been performed.
All steps in section 3.3 “Software installation, Server 3 (Collector)” have been performed.
All steps in section 3.4 “Software installation, Server 4 (Elasticsearch)” have been performed.

3.5.2 Installation Steps

1. Install JAVA (JDK 8)
2. Download Spark 2.3.0 release and un-tar it



Download Apache Spark™

1. Choose a Spark release: 2.3.3 (Feb 15 2019)
2. Choose a package type: Pre-built for Apache Hadoop 2.7 and later
3. Download Spark: spark-2.3.3-bin-hadoop2.7.tgz
4. Verify this release using the 2.3.3 signatures, checksums and project release KEYS.

3. Un-tar it and move the directory

```
$ tar xvf spark-2.3.3-bin-hadoop2.7.tgz
$ mv spark-2.3.3-bin-hadoop2.7 /mnt/spark
```

4. Setting up the environment for Spark. Add the following line to ~/.bashrc file. It means adding the location, where the software file are located to the PATH variable.

```
$ export PATH=$PATH:/mnt/spark/bin
$ source ~/.bashrc
```

5. Launch the master server

```
$ cd /mnt/spark/spark-2.3.3-bin-hadoop2.7/sbin/start-master.sh
```

6. Connect to the slaves; Now you have a master server running. To start the slave servers, you have to type the following command running from your Spark installation folder.

```
$ cd /mnt/spark/spark-2.3.3-bin-hadoop2.7/sbin/start-slaves.sh
```

7. Change the Spark Configuration files
8. Download the parser.jar file in filetsar project
9. Run on a Spark standalone cluster

```
$ cd /mnt/spark/spark-2.3.3-bin/hadoop2.7/bin/spark-submit –class
packetparser.DirectKafkaPackaetParser –conf spark.local.dir=/mnt/spark –master
spark://{master_host} /path/to/parser.jar
```

3.5.3 Configured Values

➢ Spark Application Properties

Information	Value
spark.app.name	PacketParser
spark.driver.cores	(example) 24
spark.driver.memory	(example) 1g
spark.executor.memory	(example) 1g
spark.master	Spark://HOST:PORT

For more parameter settings, please refer to https://spark.apache.org/docs/2.3.3/submitting-applications.html#master-urls

3.6 Server 6 (Analyzer)

3.6.1 Prerequisites

All steps in section 2 “Server Configurations” have been performed.
All steps in section 3.1 “Software Installation, Server 1 (Database)” have been performed.
All steps in section 3.2 “Software installation, Server 2 (Kafka System)” have been performed.
All steps in section 3.3 “Software installation, Server 3 (Collector)” have been performed.
All steps in section 3.4 “Software installation, Server 4 (Elasticsearch)” have been performed.
All steps in section 3.5 “Software installation, Server 5 (Spark)” have been performed.

3.6.2 Installation Steps

1. Download filetsar project
2. Copy “analyzer_install.sh” to filetsar directory
3. Copy “analyzer_run.sh” to filetsar directory
4. Open the filetsar directory
5. Run following commands

```
$ ./analyzer_install.sh
$ ./analyzer_run.sh /path/to/config.yml
```

3.7 Server 7 (Kibana)

3.7.1 Prerequisites

All steps in section 2 “Server Configurations” have been performed.
All steps in section 3.1 “Software Installation, Server 1 (Database)” have been performed.
All steps in section 3.2 “Software installation, Server 2 (Kafka System)” have been performed.
All steps in section 3.3 “Software installation, Server 3 (Collector)” have been performed.
All steps in section 3.4 “Software installation, Server 4 (Elasticsearch)” have been performed.
All steps in section 3.5 “Software installation, Server 5 (Spark)” have been performed.
All steps in section 3.6 “Software installation, Server 6 (Analyzer)” have been performed.

3.7.2 Installation Steps

1. Install Java 8 in order to run
2. Install Kibana with Debian Package (version 6.2.4)

```
$ sudo apt-get install apt-transport-https
$ echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | sudo tee -a
/etc/apt/sources.list.d/elastic-6.x.list
$ wget https://artifacts.elastic.co/downloads/kibana/kibana-6.2.4-amd64.deb
$ shasum -a 512 kibana-6.2.4-amd64.deb
$ sudo dpkg -I kibana-6.2.4.deb
```

3. Run Kibana with systemd

```
$ sudo /bin/systemctl daemon-reload
$ sudo /bin/systemctl enable kibana.service
```

4. Configure Kibana via config file
   Kibana loads its configuration from the /etc/kibana/kibana.yml file by default.
   The format of this config file is explained in Configuring Kibana.

3.7.3 Configured Values

Information	Value
server.host	${host_name}
elasticsearch.url	[“elasticsearch_master_ip:9200”, …]
elasticsearch.hosts	[“localhost:9200”]
path.data	/mnt/kibana
server.name	kibana
Server.port	5601

For more parameters, please refer to https://www.elastic.co/guide/en/kibana/6.2/settings.html

3.8 Server 8 (Web Server)

3.8.1 Prerequisites

All steps in section 2 “Server Configurations” have been performed.
All steps in section 3.1 “Software Installation, Server 1 (Database)” have been performed.
All steps in section 3.2 “Software installation, Server 2 (Kafka System)” have been performed.
All steps in section 3.3 “Software installation, Server 3 (Collector)” have been performed.
All steps in section 3.4 “Software installation, Server 4 (Elasticsearch)” have been performed.
All steps in section 3.5 “Software installation, Server 5 (Spark)” have been performed.
All steps in section 3.6 “Software installation, Server 6 (Analyzer)” have been performed.
All steps in section 3.7 “Software installation, Server 6 (Kibana)” have been performed.

3.8.2 Installation Steps

1. Download filetsar project
2. Copy “webserver_install.sh” to filetsar directory
3. Copy “webserver_run.sh” to filetsar directory
4. Open the filetsar directory
5. Run following commands

```
$ ./webserver_install.sh
$ ./webserver_run.sh /path/to/config.yml
```

4 References

M. Rogers, J. Goldman, R. Mislan, T. Wedge, and S. Debrota, “Computer Forensics field triage process model,” Journal of Digital Forensics, *Security, and Law*, vol. 1(2), pp. 19–38, 2006.

S. Garfinkel, “Network forensics: Tapping the internet,” *IEEE Internet Computing*, vol. 6, pp. 60–66, 2002.

C. Beek. (2016) Introduction to file carving. McAfee. [Online]. Available: http://fliphtml5.com/zmxx/prcw/basic

B. Claise, B. Trammell, and P. Aitken, “Specification of the ip flow information export (ipfix) protocol for the exchange of flow information,” IETF, Tech. Rep., 2013.

E. S. Pilli, R. C. Joshi, and R. Niyogi, “Network forensic frameworks: Survey and research challenges,” *Digital Investigation*, vol. 7, no. 1-2, pp. 14–27, 2010.

5 Definition, acronyms and abbreviations

(1) ESXi: VMware ESXi is an enterprise-class, type-1 hypervisor developed by VMware for deploying and serving virtual computers. As a type-1 hypervisor, ESXi is not a software application that is installed on an operating system; instead, it includes and integrates vital OS components, such as a kernel.

(2) FreeNAS: FreeNAS is a free and open-source network-attached storage software based on FreeBSD and the OpenZFS file system.

(3) Kafka: Apache Kafka is an open-source stream-processing software platform written in Scala and Java. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds. Its storage layer is essentially a scalable message queue designed as a distributed transaction log making it highly valuable for enterprise infrastructures to process streaming data.

(4) Spark: Apache Spark is an open-source distributed general-purpose cluster-computing framework. Spark provides is a unified analytics engine for large-scale data processing and allows for an interface for programming entire clusters with implicit data parallelism and fault tolerance.

(5) Elasticsearch: Elasticsearch is a Java based search engine which provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. Elasticsearch can be used to search all kinds of documents and provides scalable near real-time search feature for large datasets. It is developed alongside a data collection and log-parsing engine called Logstash, and analytics and visualization platform called Kibana.

(6) Packet: A packet is a basic unit of communication over a digital network. Network data consists of multiple packets together being sent and received.

(7) FTP: The File Transfer Protocol (FTP) is a protocol for file transfer between hosts. The primary function of FTP is to transfer files efficiently and reliably, and to allow the convenient use of remote file storage capabilities. A detailed description of FTP is available at https://tools.ietf.org/html/rfc354 where it was originally defined. All previous versions and upgrades made to the protocol are linked on the website.

(8) HTTP: The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. A detailed description of HTTP is available at https://tools.ietf.org/html/rfc2616 where it was originally defined. All previous versions and upgrades made to the protocol are linked on the website.

(9) IMAP: The Internet Message Access Protocol (IMAP) allows a client to access and manipulate electronic mail messages on a server. IMAP permits manipulation of mailboxes and the capability for an offline client to resynchronize with the server. A detailed description of IMAP is available at https://tools.ietf.org/html/rfc3501 where it was originally defined. All previous versions and upgrades made to the protocol are linked on the website.

(10) RTP: As part of Voice over IP (VoIP), the Real-time Transport Protocol (RTP) is a network protocol for delivering audio and video over IP networks. It is used communication and entertainment systems that involve streaming media, such as video teleconferences, television services, and web-based push-to-talk features. A detailed description of RTP is available at https://tools.ietf.org/html/rfc1889 where it was originally defined All previous versions and upgrades made to the protocol are linked on the website.

(11) SIP: As part of Voice over IP (VoIP), The Session Initiation Protocol (SIP) is a communications protocol for managing multimedia communication sessions for voice and video calls, as well as in instant messaging over Internet Protocol (IP) networks. A detailed description of SIP is available at https://tools.ietf.org/html/rfc2543 where it was originally defined. All previous versions and upgrades made to the protocol are linked on the website.

(12) SMTP: The objective of Simple Mail Transfer Protocol (SMTP) is to transfer mail reliably and efficiently across the network. A detailed description of SMTP is available at https://tools.ietf.org/html/rfc821 where it was originally defined. All previous versions and upgrades made to the protocol are linked on the website.

FILE TSAR: TOOLKIT FOR SELECTIVE ANALYSIS & RECONSTRUCTION

Dr. Kathryn Seigfried-Spellar
Department of Computer &Information Technology
Purdue University

Prof. Raymond A. Hansen
Department of Computer Science & Networking
Wentworth Institute of Technology

THE TEAM

WENTWORTH INSTITUTE OF TECHNOLOGY

PURDUE UNIVERSITY

THE CONTRIBUTING TEAM

Dr. Kathryn Seigfried-Spellar: PI & Project Manager
Prof. Raymond Hansen: PI & Architect

Dr. John Springer: Application Development
Dr. Marcus Rogers: Law Enforcement
Dr. Justin Yang: System Administration

Graduate Assistants:

- Siddarth Chowdhury
- Niveah Abraham
- Seunghee Lee
- Nicolas Vukadinovic
- Xiang Lui

PURDUE UNIVERSITY

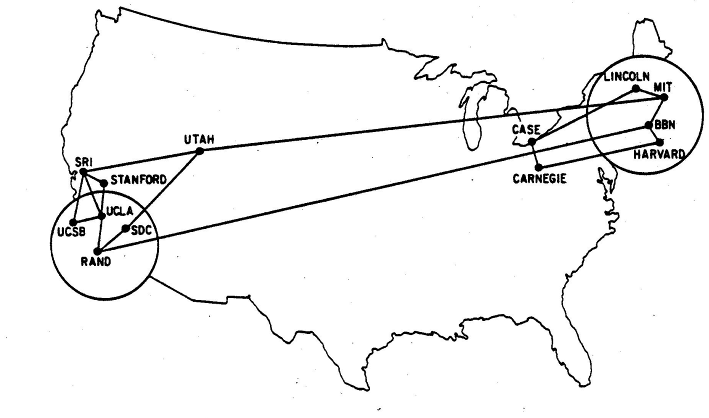
WENTWORTH INSTITUTE OF TECHNOLOGY

BACKGROUND

WENTWORTH INSTITUTE OF TECHNOLOGY

PURDUE UNIVERSITY

NIJ-2016-8976 - DEVELOPING IMPROVED MEANS TO COLLECT DIGITAL EVIDENCE ELIGIBILITY



This project was funded by the National Institute of Justice, Office of Justice Programs, U.S. Department of Justice (#2016-MU-MU-K091). The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the U.S. Department of Justice.

PURDUE UNIVERSITY

WENTWORTH INSTITUTE OF TECHNOLOGY

CHALLENGES

WENTWORTH
INSTITUTE OF TECHNOLOGY

PURDUE
UNIVERSITY

CHALLENGES

DIGGING INTO THE NETWORK FORENSIC ANALYTICS

- Technologies
- Timeliness
- Privacy
- Provenance
- Admissibility
- Courtroom Presentation
- …

BIG DATA ANALYTICS

VAST SOURCES OF DATA

DNS
DOMAIN NAME SERVER

DHCP

syslog

```
Type: AR...
Address Resolution Proto...
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  ... request (0x0001)
  ...4c:00:10:40:...
```

tcpdump

KERBEROS

LDAP

NetFlow

SNMP
Simple Network Management Protocol

WENTWORTH
INSTITUTE OF TECHNOLOGY

PURDUE
UNIVERSITY

BIG DATA ANAYTICS

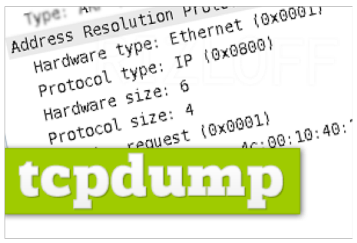
DIGGING INTO THE NETWORK FORENSIC ANALYTICS

The key challenge for this tool kit is to recover & reconstruct files

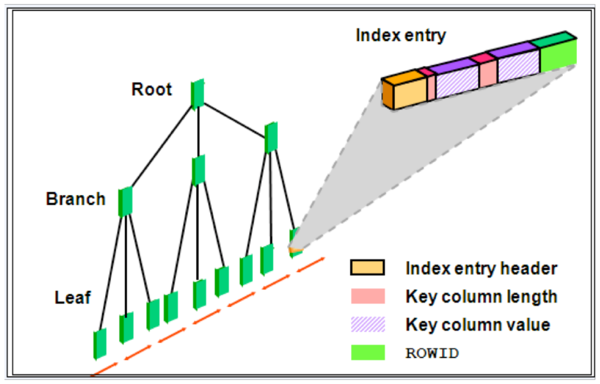
BIG DATA ANALYTICS

DIGGING INTO THE NETWORK FORENSIC ANALYTICS

The key challenge is to recover & reconstruct files from network data



```
Type: AR
Address Resolution Proto
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
      request (0x0001)
                    :00:10:40:
```



PURDUE
UNIVERSITY

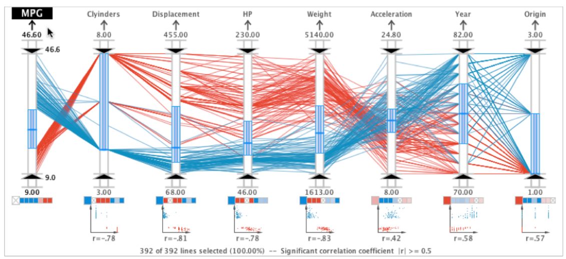
WENTWORTH
INSTITUTE OF TECHNOLOGY

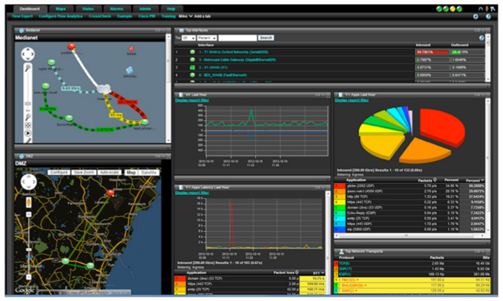
This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

BIG DATA ANALYTICS

DIGGING INTO THE NETWORK FORENSIC ANALYTICS

ANALYTICS & VISUALIZATION





Indexed File Trees

Raw Captures

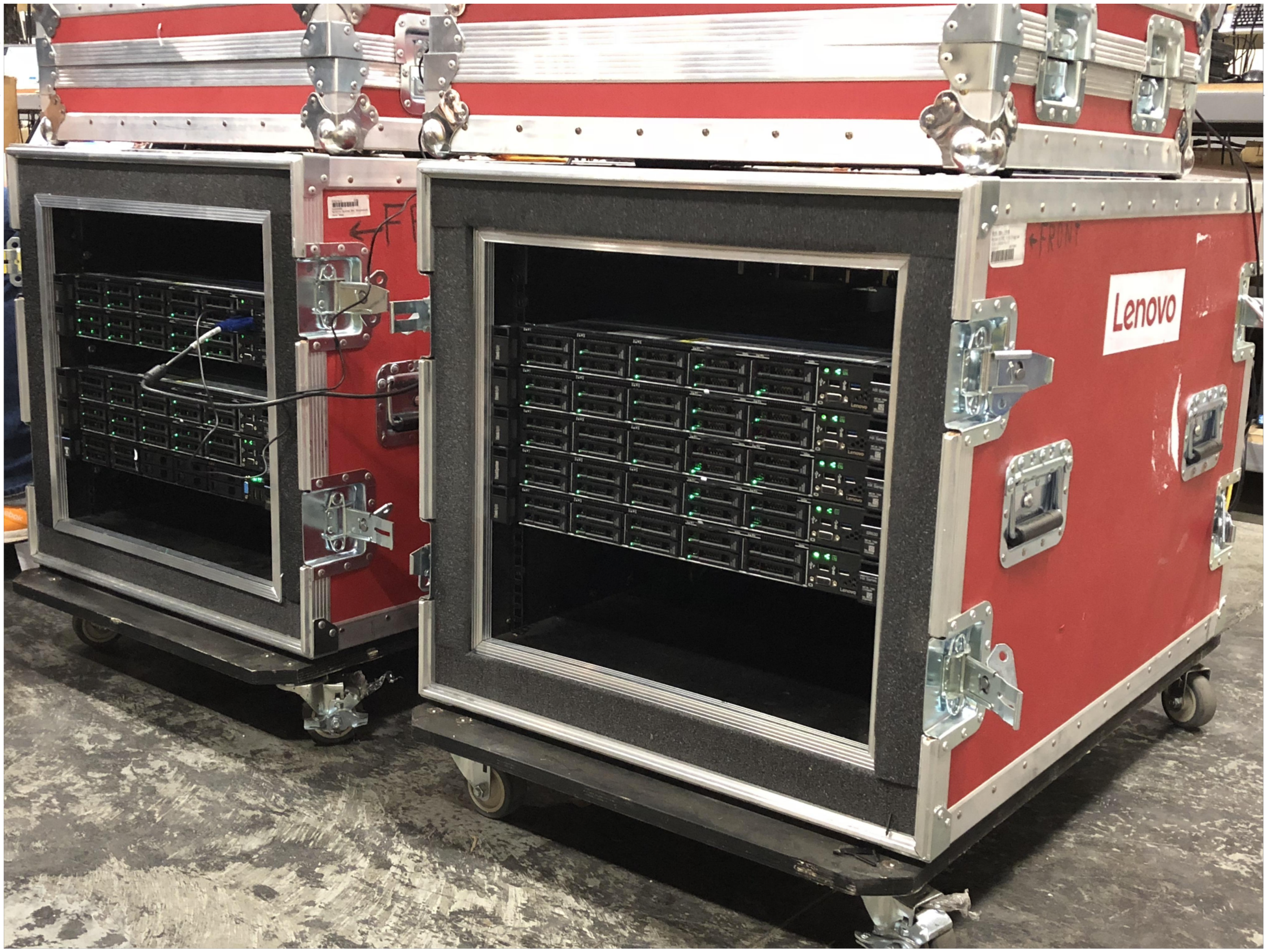
Network Data

PURDUE UNIVERSITY

WENTWORTH INSTITUTE OF TECHNOLOGY

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

FILETSAR CRASH CART



PURDUE UNIVERSITY

WENTWORTH INSTITUTE OF TECHNOLOGY

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

LAW ENFORCEMENT COLLABORATORS

HTCU & NW3C

- A collaboration of local law enforcement in Tippecanoe County, Indiana and the Purdue University Cyberforensics program.
- Provides investigative resources when examining various forms of digital evidence.
- Housed on campus at Purdue's Discovery Park in order to provide students with real world experience, research topics, software development, and digital forensics training.
- Two faculty members are deputized members of the HTCU

- Congressionally funded non-profit corporation providing support for law enforcement working economic and high-tech crime
- Provide training in computer forensics, cyber and financial crime investigations, and intelligence analysis.
- Provide certifications: Certified Economic Crime Forensic Examiner (CECFE) and Certified Cyber Crime Examiner (CCCE)
- Conduct original research on all facets of white collar crime.

PURDUE UNIVERSITY

WENTWORTH INSTITUTE OF TECHNOLOGY

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

LAW ENFORCEMENT COLLABORATORS

TESTING AND FEEDBACK

30-day beta testing

- Examiners were given remote access to the toolkit along with a user manual, providing details on functionality & usage.
- They represented a broad range of knowledge in network forensic investigations, from basic to advanced training.
- The research team incorporated a ticketing system in order to provide technical support to the examiners and log all bugs/requests during testing.

Anonymous Survey

- Internet-based anonymous survey in order to provide feedback on the features and functionality of the toolkit
- Questions assessed: setup and login, user interface, capture options, status management, dashboard, and final reconstruction.
- Results indicated satisfaction with the core functionalities.
- Additional user-friendly features were requested.

PURDUE UNIVERSITY

WENTWORTH INSTITUTE OF TECHNOLOGY

DEMO

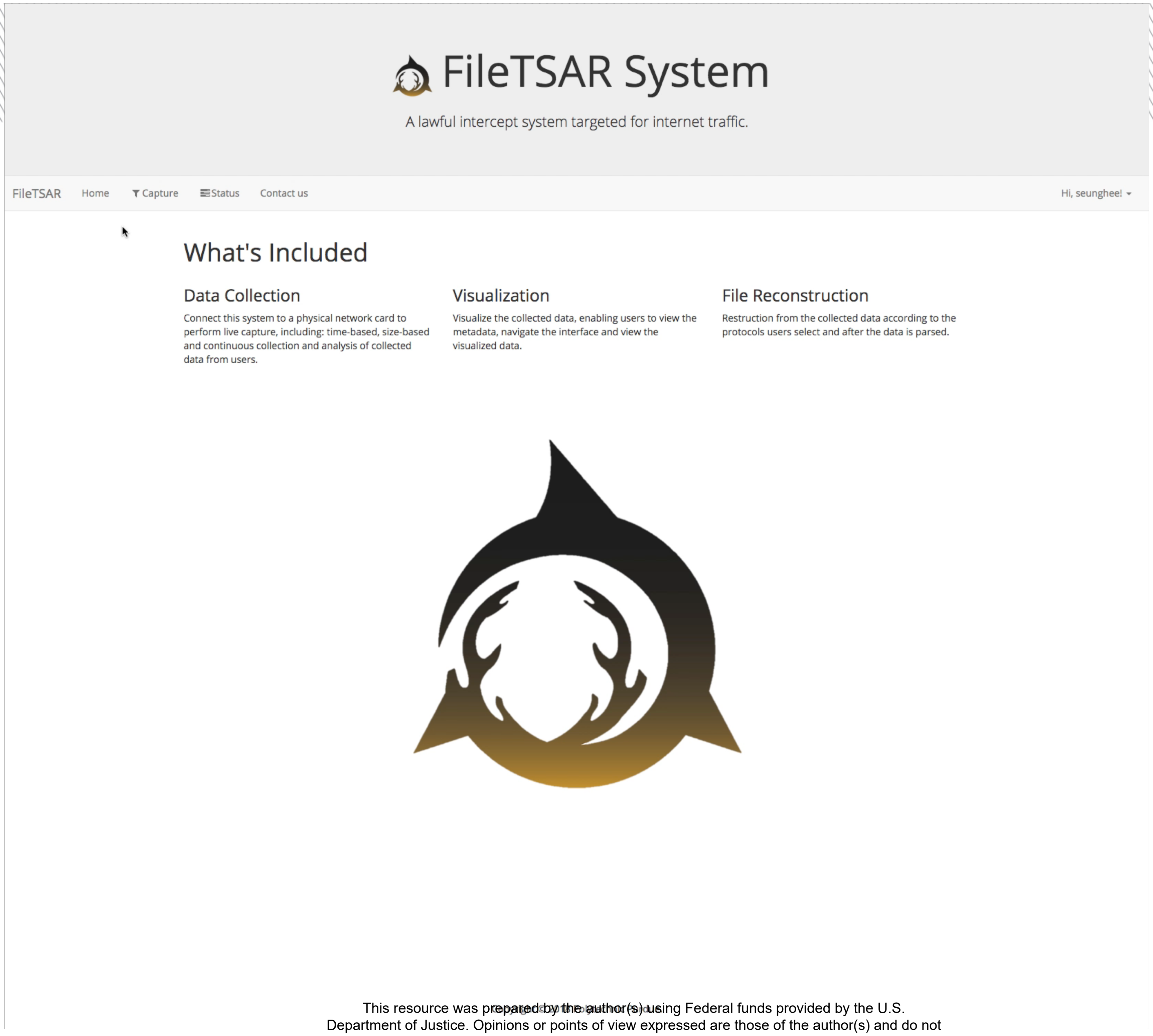
DEMONSTRATION – BASIC SCENARIO

A system administrator has reported that malware has gained access into the organization and infected a handful of workstations. They were uncertain of how the malware gained access to the network, but have determined that the number of systems infected has continued to grow. Their basic investigation has ruled out host-to-host communications within the network. Determine if this malware infection has occurred via the web, email, or other file transfers. This will require analysis of web traffic, email for embedded scripts or attachments, etc.

PURDUE
UNIVERSITY

WENTWORTH
INSTITUTE OF TECHNOLOGY

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.


FileTSAR System

A lawful intercept system targeted for internet traffic.

What's Included

Data Collection

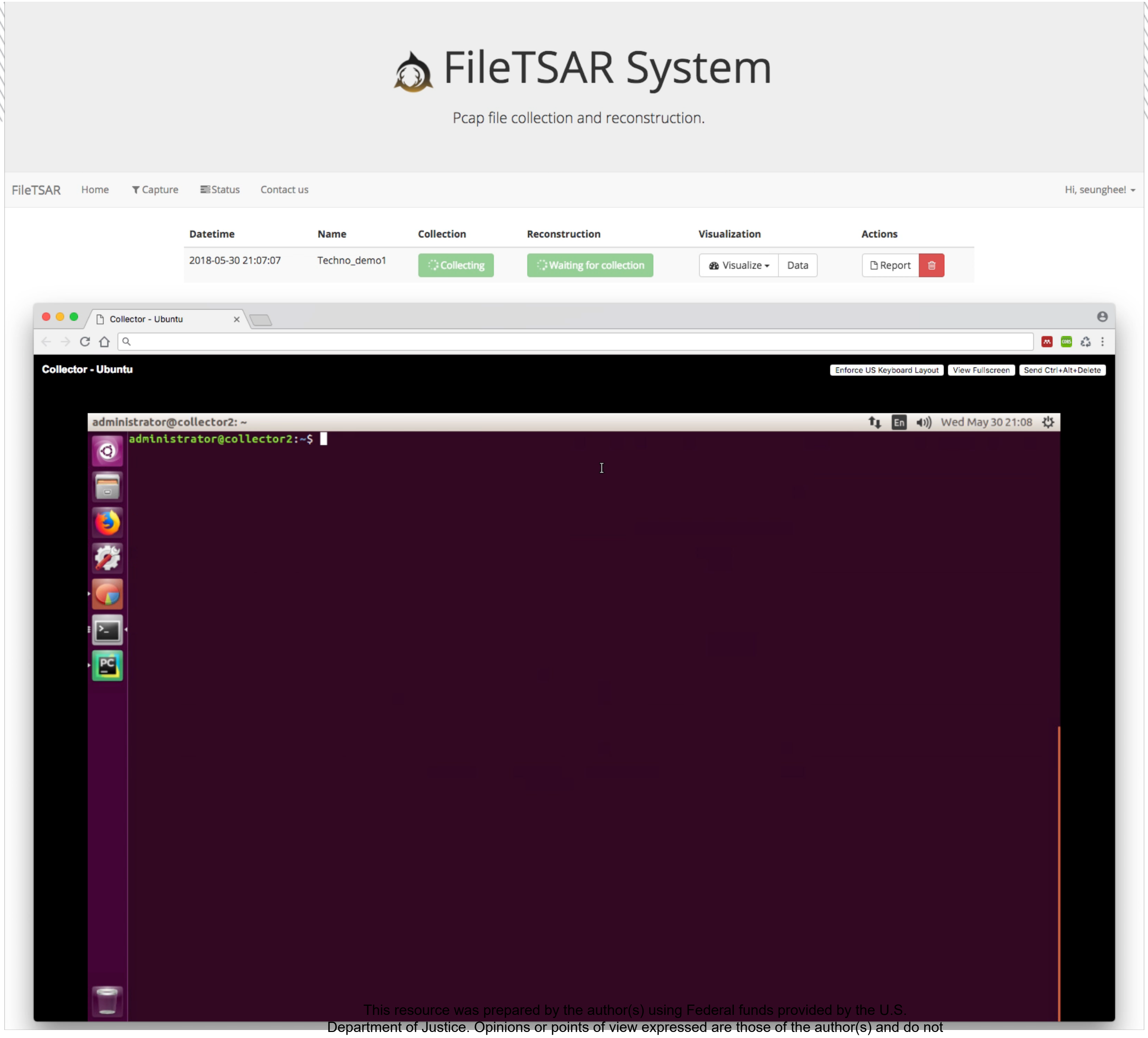
Connect this system to a physical network card to perform live capture, including: time-based, size-based and continuous collection and analysis of collected data from users.

Visualization

Visualize the collected data, enabling users to view the metadata, navigate the interface and view the visualized data.

File Reconstruction

Restruction from the collected data according to the protocols users select and after the data is parsed.

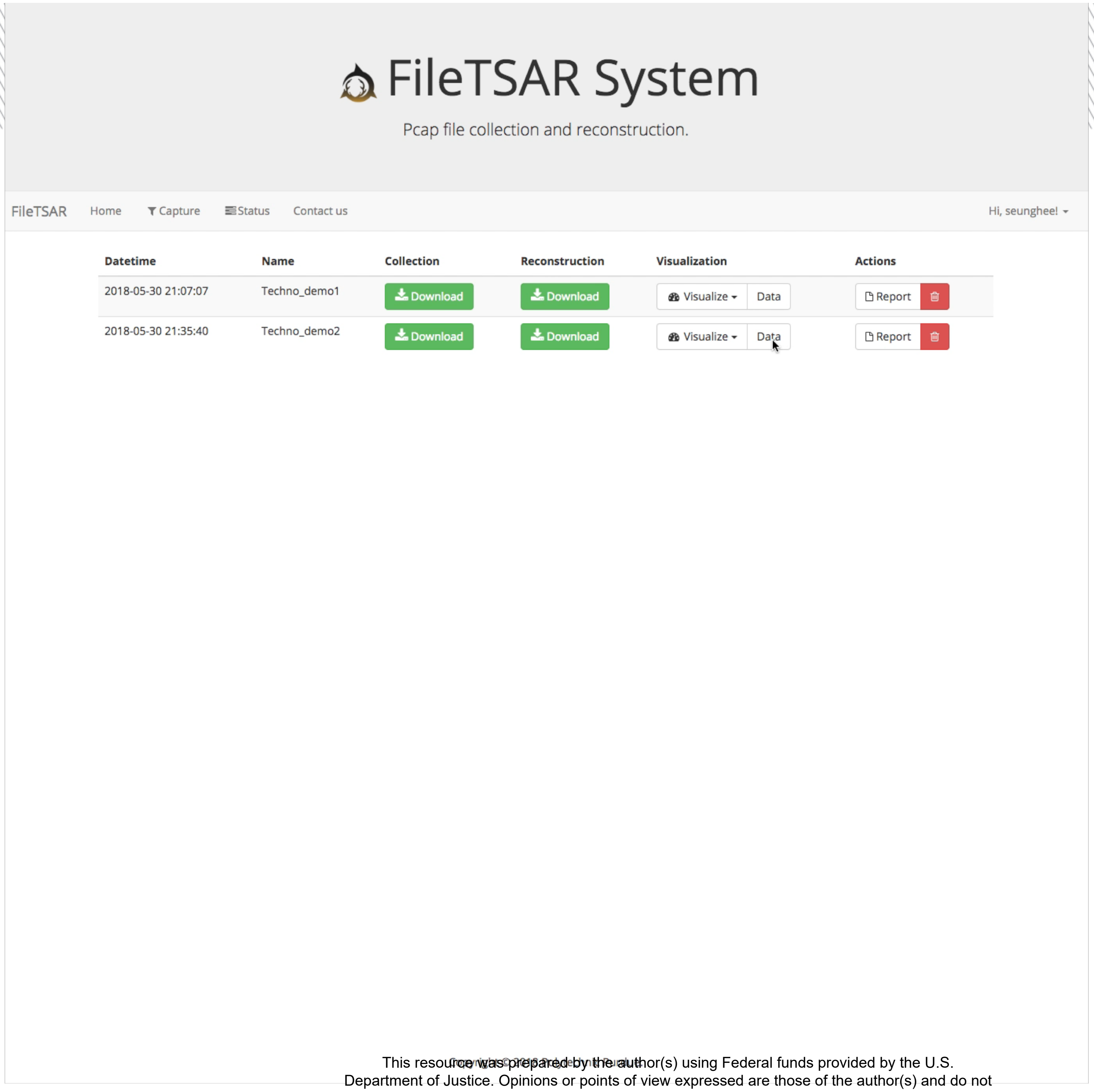


FileTSAR System

Pcap file collection and reconstruction.

Datetime	Name	Collection	Reconstruction	Visualization	Actions
2018-05-30 21:07:07	Techno_demo1	Collecting	Waiting for collection	Visualize Data	Report

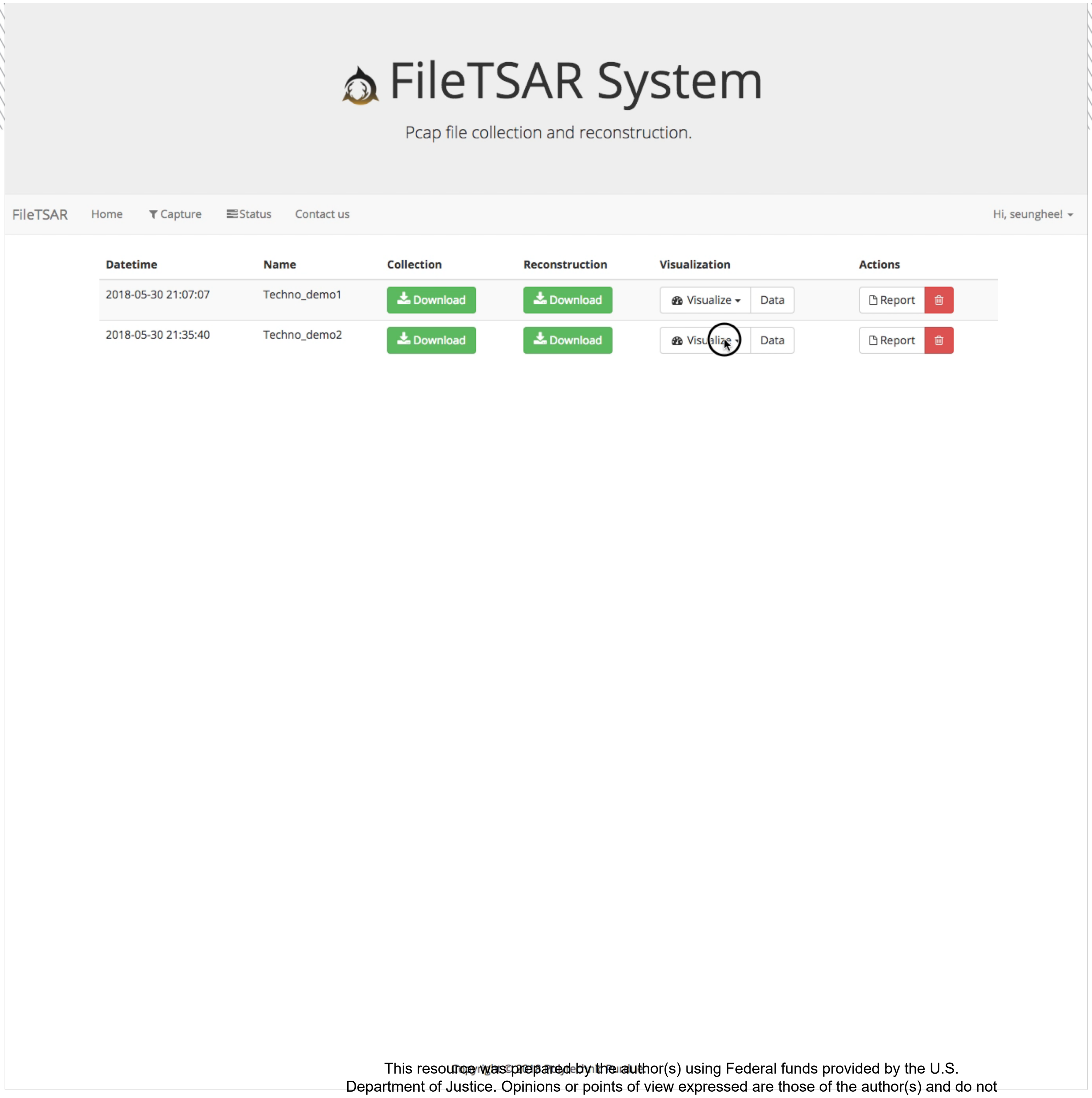
```
administrator@collector2:~$
```


FileTSAR System

Pcap file collection and reconstruction.

Datetime	Name	Collection	Reconstruction	Visualization	Actions
2018-05-30 21:07:07	Techno_demo1	Download	Download	Visualize Data	Report
2018-05-30 21:35:40	Techno_demo2	Download	Download	Visualize Data	Report

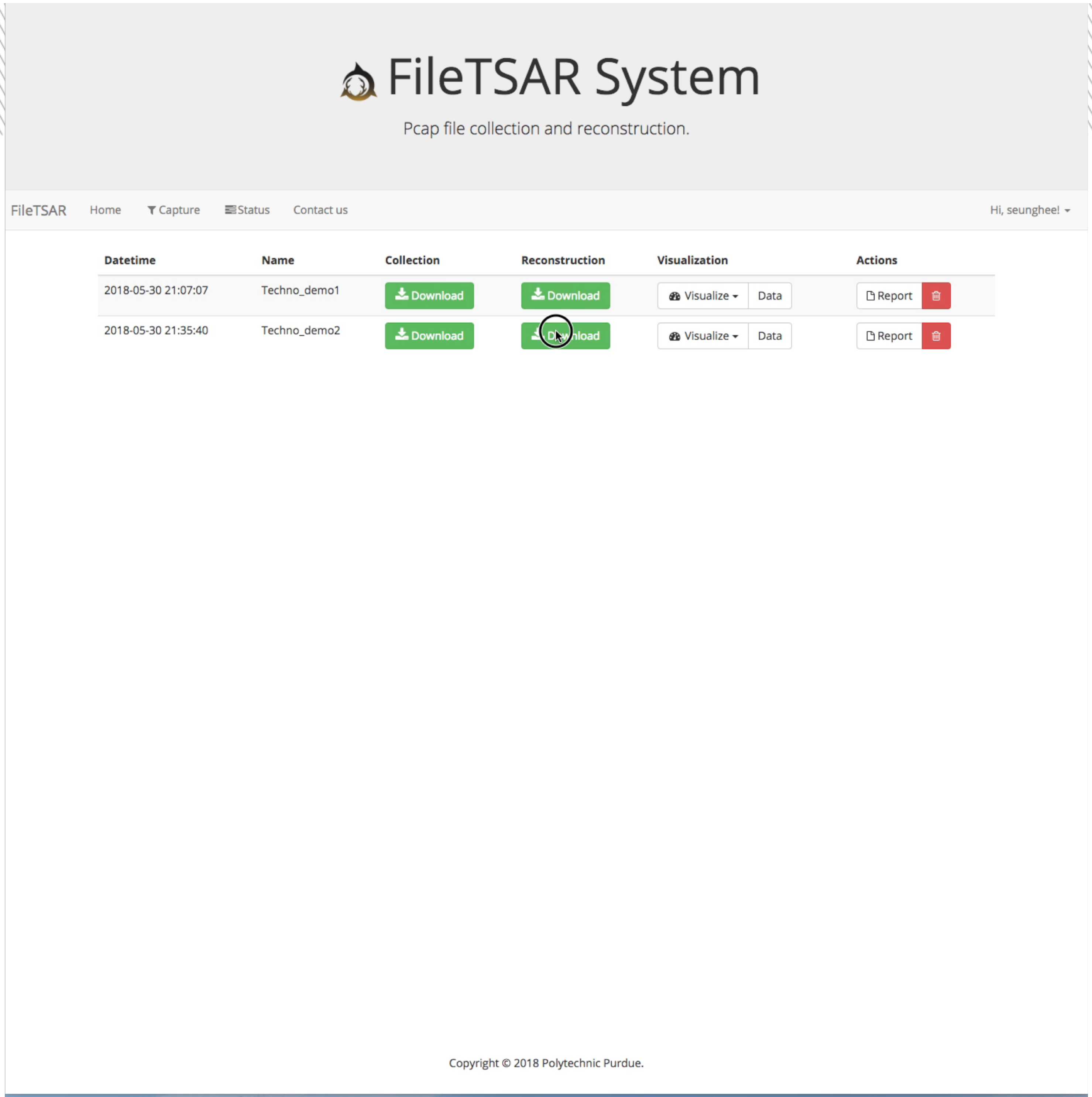
Copyright © 2018 Polytechnic Purdue.



FileTSAR System

Pcap file collection and reconstruction.

Datetime	Name	Collection	Reconstruction	Visualization	Actions
2018-05-30 21:07:07	Techno_demo1	Download	Download	Visualize Data	Report
2018-05-30 21:35:40	Techno_demo2	Download	Download	Visualize Data	Report

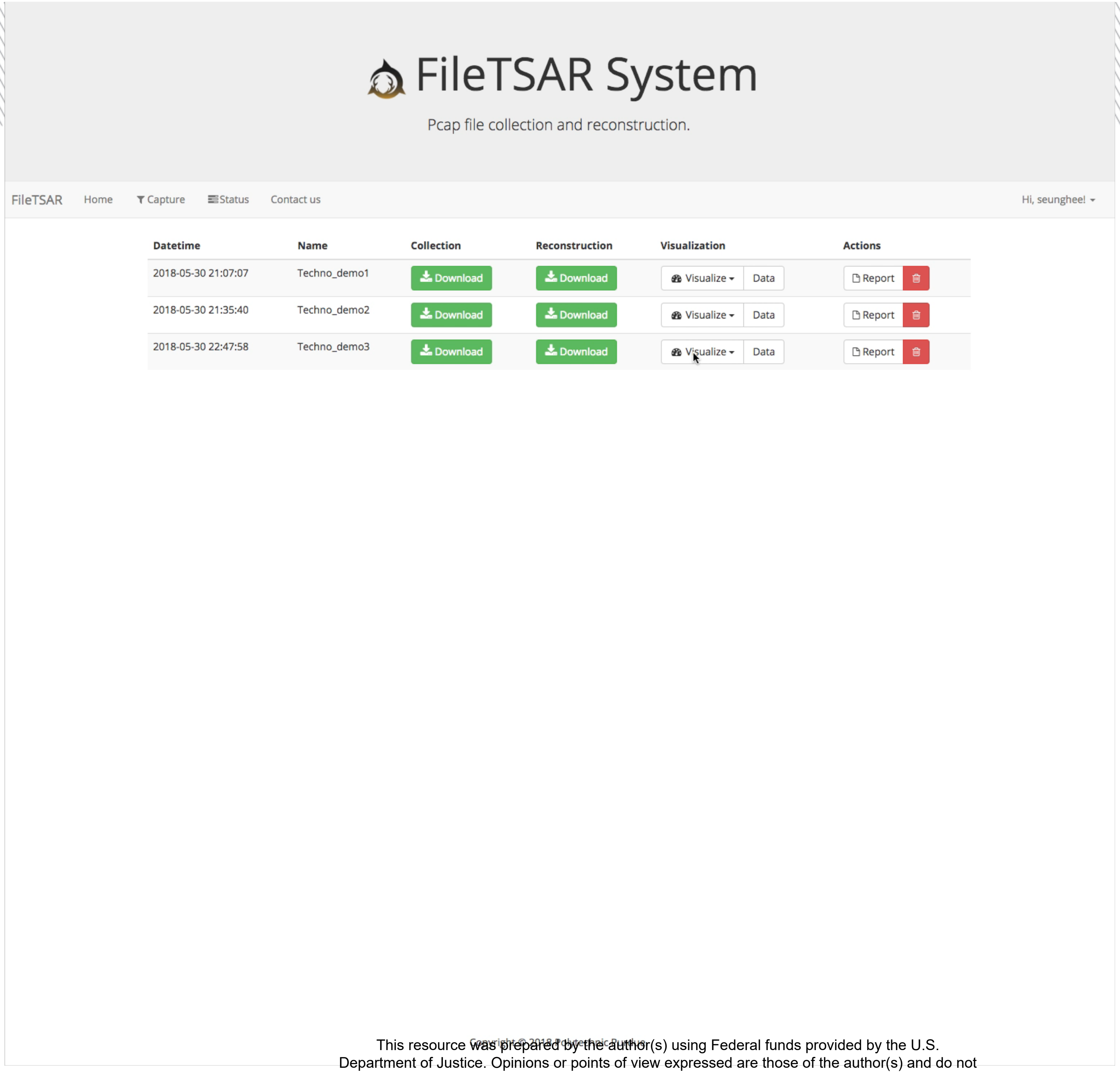


FileTSAR System

Pcap file collection and reconstruction.

Datetime	Name	Collection	Reconstruction	Visualization	Actions
2018-05-30 21:07:07	Techno_demo1	Download	Download	Visualize Data	Report
2018-05-30 21:35:40	Techno_demo2	Download	Download	Visualize Data	Report

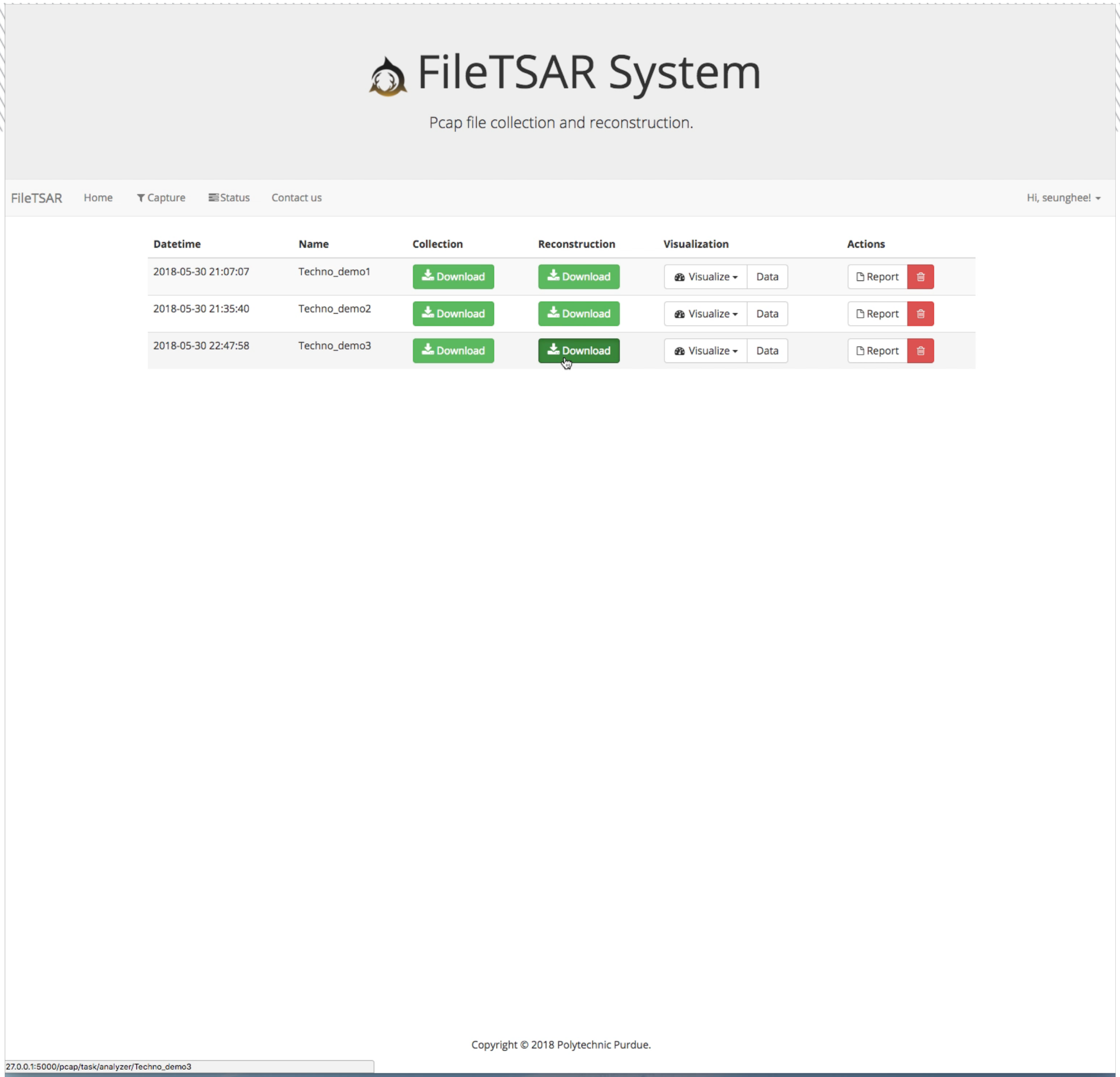
Copyright © 2018 Polytechnic Purdue.



FileTSAR System

Pcap file collection and reconstruction.

Datetime	Name	Collection	Reconstruction	Visualization		Actions	
2018-05-30 21:07:07	Techno_demo1	Download	Download	Visualize	Data	Report	🗑
2018-05-30 21:35:40	Techno_demo2	Download	Download	Visualize	Data	Report	🗑
2018-05-30 22:47:58	Techno_demo3	Download	Download	Visualize	Data	Report	🗑



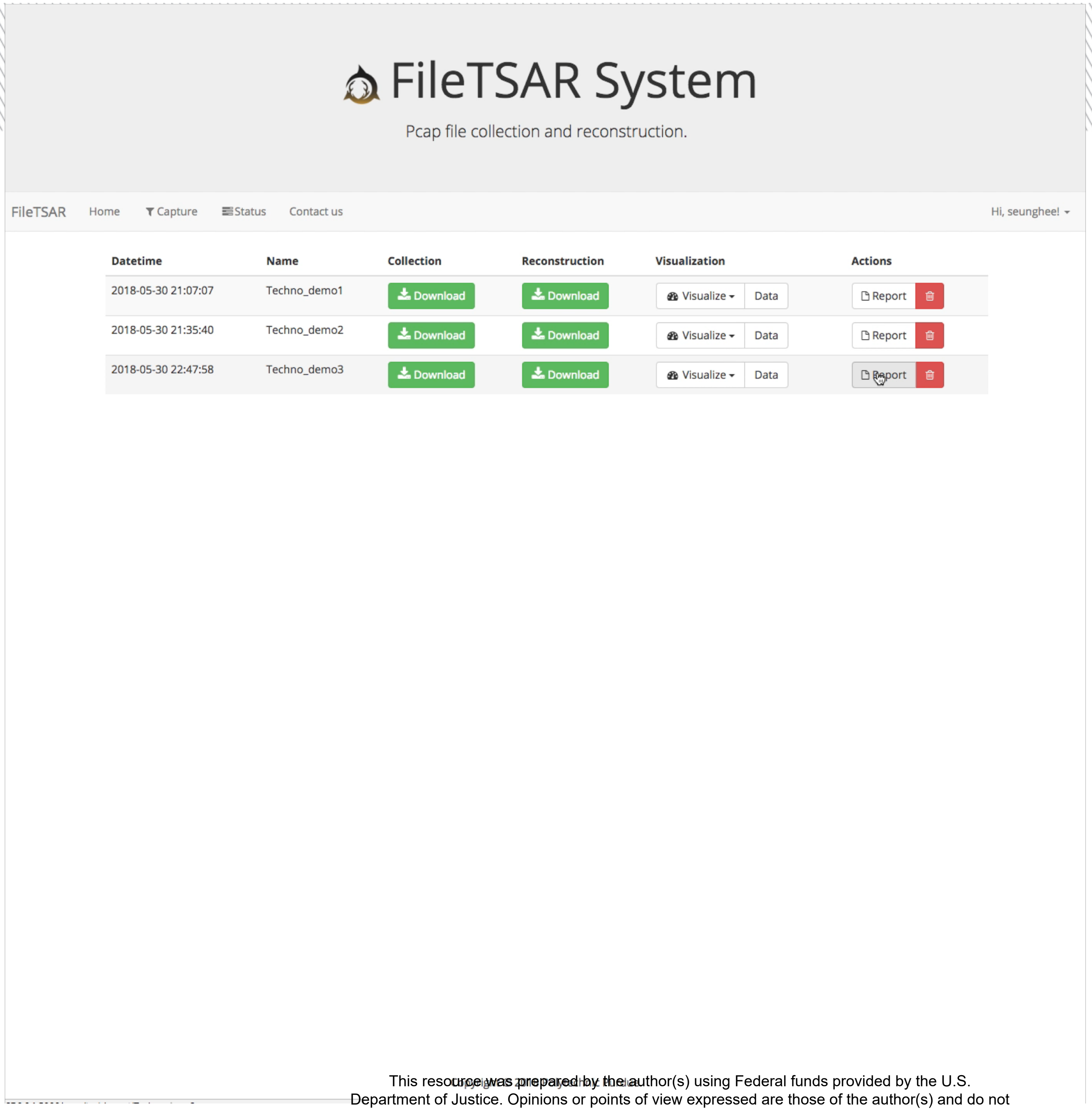
FileTSAR System

Pcap file collection and reconstruction.

Datetime	Name	Collection	Reconstruction	Visualization	Actions
2018-05-30 21:07:07	Techno_demo1	Download	Download	Visualize Data	Report
2018-05-30 21:35:40	Techno_demo2	Download	Download	Visualize Data	Report
2018-05-30 22:47:58	Techno_demo3	Download	Download	Visualize Data	Report

Copyright © 2018 Polytechnic Purdue.

27.0.0.1:5000/pcap/task/analyzer/Techno_demo3



FileTSAR System

Pcap file collection and reconstruction.

Datetime	Name	Collection	Reconstruction	Visualization	Actions
2018-05-30 21:07:07	Techno_demo1	Download	Download	Visualize Data	Report
2018-05-30 21:35:40	Techno_demo2	Download	Download	Visualize Data	Report
2018-05-30 22:47:58	Techno_demo3	Download	Download	Visualize Data	Report

FUTURE PLANS

CONTINGENT UPON $$$

Commercialization

Additional Filetypes

GPUs

UX Refinement

…

PURDUE UNIVERSITY

WENTWORTH INSTITUTE OF TECHNOLOGY

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

FUTURE TRAINING

FILE TSAR LAW ENFORCEMENT TRAINING WORKSHOP (3-DAYS)

Tuesday, September 11th – Thursday, September, 13th

This 3-day training workshop will be hosted at Purdue University (West Lafayette, IN) thanks to funding from the National Institute of Justice.

We have 20 seats available for the 3-day training workshop.

All travel expenses (hotel, flight, etc) will be covered, and a per diem will be provided to each participant to cover meal expenses.

Webinar training and tutorials will be made available at a date TBD

PURDUE
UNIVERSITY

WENTWORTH
INSTITUTE OF TECHNOLOGY

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

QUESTIONS

THANK YOU

Dr. Kathryn Seigfried-Spellar
Assistant Professor
Purdue University
Department of Computer & Information Technology
kspellar@purdue.edu

Prof. Raymond A. Hansen
Associate Professor
Wentworth Institute of Technology
Department of Computer Science & Networking
hansenr2@wit.edu

WENTWORTH INSTITUTE OF TECHNOLOGY

PURDUE UNIVERSITY