PCAPFunnel: A Tool for Rapid Exploration of Packet Capture Files

Juraj Uhlár
Flowmon Networks
Brno, Czech Republic
Email: juraj.uhlar@flowmon.com

Martin Holkovič
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
Email: iholkovic@fit.vutbr.cz

Vít Rusňák
Institute of Computer Science
Masaryk University
Brno, Czech Republic
Email: rusnak@ics.muni.cz

***Abstract*—Analyzing network traffic is one of the fundamental tasks in both network operations and security incident analysis. Despite the immense efforts in workflow automation, an ample portion of the work still relies on manual data exploration and analytical insights by domain specialists. Current state-of-the-art network analysis tools provide high flexibility at the expense of usability and have a steep learning curve. Recent—often web-based—analytical tools emphasize interactive visualizations and provide simple user interfaces but only limited analytical support. This paper describes the tool that supports the analytical work of network and security operators. We introduce typical user tasks and requirements. We also present the filtering funnel metaphor for exploring packet capture (PCAP) files through visualizations of linked filter steps. We have created PCAPFunnel, a novel tool that improves the user experience and speeds up packet capture data analysis. The tool provides an overview of the communication, intuitive data filtering, and details of individual network nodes and connections between them. The qualitative usability study with nine domain experts confirmed the usability and usefulness of our approach for the initial data exploration in a wide range of tasks and usage scenarios, from educational purposes to exploratory network data analysis.**

***Index Terms*—Data analysis, Data visualization, Network traffic analysis, Packet captures**

I. Introduction

Day-to-day monitoring of the network status and intervention, if necessary, is part of the network administrators' duties [1]. Many problems appear, causing the network or its parts to malfunction daily. E.g., communication issues with the webserver, broken connections between distant locations, or a user's computer spreading malware across the network. Except for real-time monitoring, packet capture inspection is a standard method used during network analysis. However, it is nearly impossible to analyze all traffic on a per-packet basis due to the overwhelming amount of transferred data, especially in high-speed networks. Therefore the network administrators work with the aggregated data only, which yields to the crucial challenge: selecting the proper criteria for data aggregation and their presentation. They need to explore multi-dimensional data and gain knowledge by analyzing them on multiple levels of abstraction [2].

Commonly used low-level command-line tools or applications like Wireshark [3] provide broad data processing and filtering capabilities but provide limited data visualization and presentation in a comprehensible way. The previous studies confirmed the usefulness of visualizations for network analysis [4], [5] which outperforms the traditional text-based ones. Novel tools such as NetCapVis [6] bootstrap the network analysis through interactive and easy-to-learn graphical interfaces, which makes them accessible also for novice users. However, less attention is on the support and guidance of the analysts in their exploratory work.

We address these issues in the prototype implementation of a web-based visual analysis tool for packet capture (PCAP) data. *PCAPFunnel* enables to upload packet capture files and performs a rapid exploration through a set of consecutive filters. It improves the initial orientation in the network dataset and allows the export and import of filter configurations to simplify their sharing or reuse. The filtering funnel metaphor guides the analysts and allows quick drill down to the details of individual network nodes or connections between them. We performed a qualitative user evaluation with nine domain experts who considered our approach helpful in a wide range of usage scenarios for exploratory network data analysis or educational purposes. The project has been done in collaboration with Flowmon Networks a.s., a company providing network monitoring solutions.

We contribute to state of the art with: 1) a filtering funnel metaphor applied on an exploration of network traffic datasets, 2) design and implementation of PCAPFunnel, a visual analysis tool for rapid exploration of PCAP files,

Section II covers the related work in network traffic analysis. Section III provides the data abstraction, outlines user tasks, and presents design requirements on analytical tools for our target domain. In Section IV, we describe PCAPFunnel design. Section V summarizes the qualitative user study demonstrating the usability and usefulness of our approach. We present the discussion and outline future work in Section VI and conclude our paper in Section VII.

II. Related Work

We first introduce the network traffic data sources. To better situate our work, we present the three-level workflow of network traffic analysis, including examples of currently used tools. Finally, we discuss existing visualization tools that inform the PCAPFunnel design or provide akin capabilities.

A. *Network Traffic Data Sources*

The network traffic analysis can be either active or passive [7]. The active approach requires generating additional network traffic to check a device's status or link. The passive approach uses only data that are already available in the network. In the remainder, we focus on passive analysis as it is more accessible since it does not require any additional network infrastructure.

Two primary data sources for passive network traffic analysis are packet captures (PCAP) and network flows (NetFlow). A PCAP file [8] contains complete copies of packets transferred over the network. Therefore, it allows analyzing all values from packet headers and payloads. Because storing and processing complete packets is very expensive, it is usually not performed globally but only for a limited network segment. NetFlow [9] represents connections between communicating nodes (i.e., flows). Compared to PCAP, it contains only a limited amount of attributes from each packet. While NetFlow technology is predominantly used in real-time network monitoring, PCAPs are suitable for detailed inspection of network performance issues or forensic analysis of detected cybersecurity incidents [10].

B. *Network Traffic Analysis*

There are three levels network traffic analysis that differ in the level of automation and depth of the required knowledge of the analysts or network operators who perform them:

Automatic diagnostic tools process the captured network traffic, try to identify network issues, and report them [7], [11], [12]. A common strategy is that the administrators validate these reports and take countermeasures if necessary. However, the diagnostic tools heavily rely on the knowledge base they implement, which often limits their capabilities. Although most of the routine tasks are automated, many reported network issues require administrators' attention. An example of such a tool is Flowmon Packet Investigator [13].

The **top-level analysis** is usually performed over the aggregated data. It includes data sorting and filtering according to several different criteria. Although the workflow is straightforward, it gives the administrators the power to explore even massive network traffic and identify context before diving into the details. One of the tools for NetFlow data analysis is NFDump [14].

The **in-depth analysis** allows the administrators to explore the detailed information and content of individual packets [15]. It is powerful but time-consuming and often also prone to losing the "big picture." Therefore, the analysts often switch continually between the top-level and in-depth analyses in their work.

To capture and inspect the traffic, network administrators and security analysts often use command-line utilities (e.g., tcpdump, TShark) or highly flexible tools, such as Wireshark [3] and NetworkMiner [16]. The latter ones also provide statistics as well as details of each captured packet. However, they present the data in tables offering only limited visualization capabilities (e.g., tabular row highlighting or simple static line or bar charts). They are, however, used mainly by skilled users, and the steep learning curve is one of their main disadvantages.

These three approaches are usually combined into a complex workflow of network or security operations (NetOps or SecOps) teams. The automatic diagnostic tools notify the administrators about the network issue, and they initiate the top-level analysis. If these two steps do not provide sufficient information, the analysts continue with in-depth network traffic inspections using proper tools [17].

In our work, we focus wholly on top-level analysis. PCAPFunnel aims to support the analysis bootstrap, reduce the *time-to-first-insight*, and improve the analytical process by providing better guidance and support through interactive visualizations.

C. *Packet Capture Visualizations*

With the growing amount of transferred network data, the design of efficient visualizations gains importance [18]. Visualizations are gradually used in various areas of network security [19]. Over a decade ago, Goodall [5] performed a comparative evaluation of a conventional PCAP file analysis tool with a visualization application. The visualization application outperformed the conventional tool in both task accuracy and completion times. The evaluation also provided insights and the overall preference of study participants. Further, we show several examples of such visualization tools.

GrassMarlin [20] is an open-source tool released by the NSA. Its primary purpose is to help passively map industrial control systems and networks and visualize them in a communication graph.

SNAPS [21] and EventPad [22] represent specialized tools for network analysts and require strong domain knowledge. Both the tools work with PCAP files. The former provides a bottom-up pixel-oriented approach for iterative analysis and parallel filtering options. The latter enables pattern identification and analysis of malware activity using visual analytics methods. Another example of an advanced visualization tool is FloVis [23], a suite of visualization tools intended to complement command-line utilities. It processes NetFlow data and visualizes activity diagrams, communication clustering, and connection details.

The advances in modern web application development open new possibilities for leveraging visual analytics methods in network analysis tools. Several tools provide analytical capabilities as a service. For example, CapAnalysis [24] allows users to review large PCAP files, parse the data streams, filter out ports, protocols, or IP addresses, and associate them with geographical areas. A-packets [25] and PacketTotal [26] provide multiple individual views on PCAP files. Rather than supporting the explorative analysis, both tools provide dataset overview and multiple dashboards focused on individual characteristics extracted from the data (e.g., application, SSL certificates, transferred files). The closest to our approach is NetCapVis [6]. It provides both overview and fundamental analytical support through filtering based on incoming and outgoing IP addresses and port numbers. Its main limitation is only limited details for individual network nodes or connections between them. However, a user can export filter configuration for Wireshark to enable investigation of further details.

In PCAPFunnel, though inspired by these tools, we focused on improving the support and guidance to reduce users' cognitive load during the analytical process. Our goal was to design a tool that will enable rapid initial exploration of the dataset, be useful for skilled professionals, and easy to learn for novice users.

III. Data, Tasks and Requirements

The section provides the data abstraction of PCAP files, followed by an overview of the user tasks. Based on these, we formulate five design requirements for PCAPFunnel.

A. Data Abstraction

PCAP is a binary file format described in [8]. The file structure starts with a global header followed by at least one record for each packet consisting of a packet header and its payload (i.e., content data). There are multiple implementations (e.g., WinPcap, NPcap) that differ in API methods and supported features. Though, we worked with the most common *libpcap*[1], considered a de-facto standard. Moreover, we currently use only a subset of PCAP data needed for the visualizations.

We represent each packet as an object with several attributes listed in Table I to simplify data manipulation.

We also enhanced the packet with the *application name* attribute representing the application or service corresponding to the source or destination port. The information is based on the IANA Registry [27].

TABLE I
List of packet attributes used in PCAPFunnel.

Attribute	PCAP Data Property
index	`frame.number`
timestamp	`frame.time_epoch`
network/transport protocol	`frame.protocols`
source/destination IP	`src/dst` (e.g., `ip.src`, `ip.dst`)
source/destination port	`srcport/dstport` (e.g., `udp.srcport`)
bytes	`frame.len`

B. Tasks

Although NetOps and SecOps teams focus on different goals, they perform similar network analysis tasks on the same input data. Ulmer et al. [6] identified three general tasks performed by both teams: *GT1:* Get a network traffic overview. *GT2:* Find suspicious connections. *GT3:* Determine relevant events for an in-depth analysis.

Based on discussion with domain experts, we have identified five specific tasks (ST), also common to both teams:

ST1: Identify the top N communication sources, based on given criteria;

ST2: Discover unusual patterns in the network traffic (e.g., peaks);

ST3: Identify nodes with which the particular station communicated.

ST4: Identify nodes providing specific services to a network (e.g., DNS server);

ST5: Share the analysis parameters with coworkers.

We derived these tasks from real-world scenarios and the personal experience of several Flowmon Networks employees.

C. User Requirements

We address the user tasks through six design requirements that have driven our work on PCAPFunnel:

R1: Provide descriptive statistics of the network's traffic properties (e.g., traffic volume, number of connections, top N statistics for a typical network (e.g., IP, ICMP) and transport (e.g., UDP, TCP) protocols, ports, or IP addresses).

R2: Enable intuitive and progressive filtering of data by multiple packet properties.

R3: Provide details for individual nodes and connections between them and identify individual packets for further analysis in external packet analyzer software.

R4: Enhance the PCAP data with information from external sources where possible (e.g., network node geolocation, DNS resolving).

R5: Support sharing and reuse of filter configurations.

R6: Allow working with large datasets progressively when loaded without disrupting the user experience.

IV. PCAPFunnel Design

PCAPFunnel is a React-based web application communicating with a Node.js server and external APIs, providing complementary information about individual IP addresses (e.g., resolved DNS names and geolocations). The demo is available at https://pcap-viz.surge.sh.

The main goals of the tool are (a) to reduce the *time-to-first-insight* and (b) to support the users in their explorative drill down of PCAP files. The application follows Shneiderman's "Overview first, zoom and filter, then details on demand" [28]. For data organization and separating different detail levels, the application leverages tabs. The Dataset Overview provides filtering options and an overview of the loaded PCAP dataset. Detailed Views display statistics and attributes of network entities: network nodes (defined by their IP addresses) or connections between them. In the following sections, we discuss these views in detail.

A. Dataset Overview

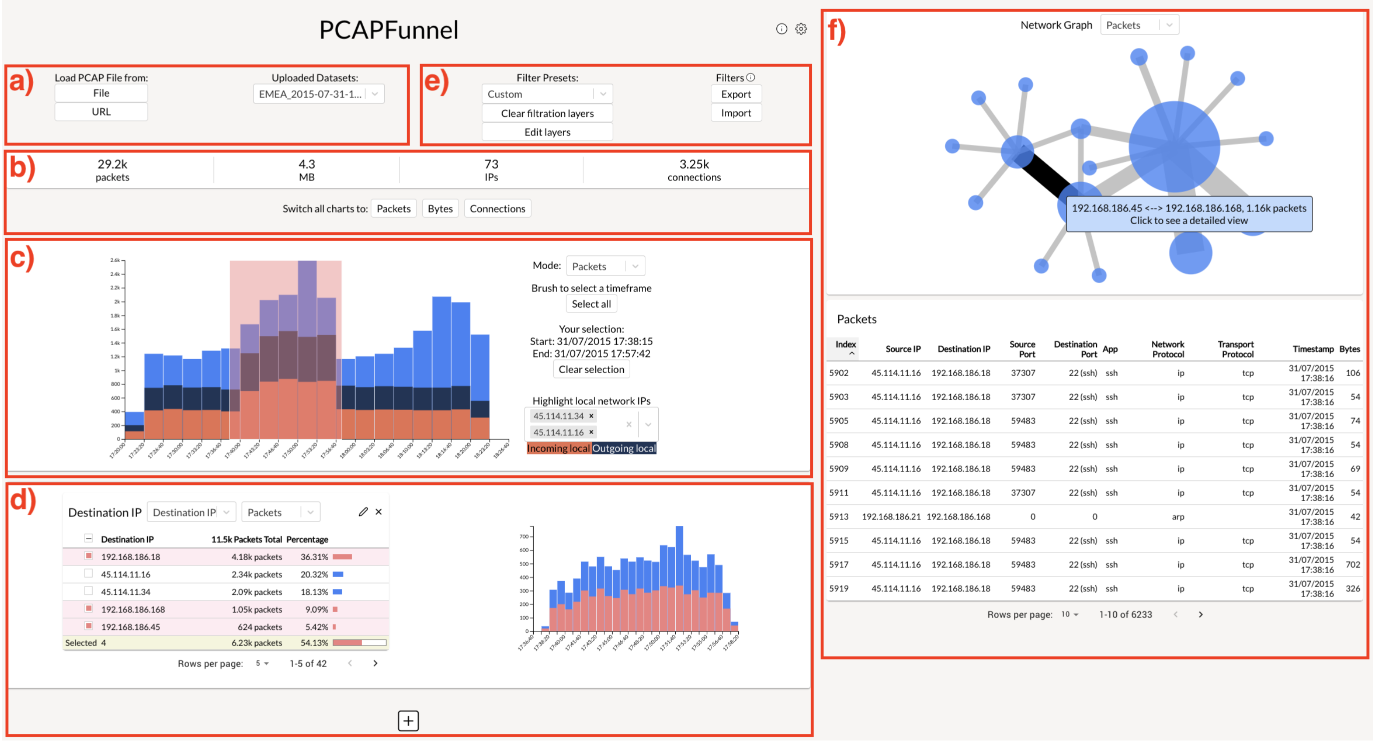
The user interface has two main sections. The left part (Fig. 1(a–e)) serves for data input and filtering, the right one (Fig. 1(f))for data visualization and exploration (**R1**).

File Upload and Summary:

A user can upload a PCAP file from local storage or as a publicly available URL (Fig. 1(a)). The uploaded PCAP file is split into progressively processed batches. The packet attributes from each batch are extracted using TShark and stored as a JSON file on the server. Fig. 2 overviews the data preprocessing workflow.

This approach allows the user to start exploring the data while the PCAP file is still being processed (**R6**). The dataset

[1] https://www.tcpdump.org


PCAPFunnel

a)
Load PCAP File from:
File
URL
Uploaded Datasets: EMEA_2015-07-31-1...

e)
Filter Presets: Custom
Clear filtration layers
Edit layers
Filters
Export
Import

b)

29.2k packets	4.3 MB	73 IPs	3.25k connections

Switch all charts to: Packets Bytes Connections

c)

Mode: Packets
Brush to select a timeframe
Select all

Your selection:
Start: 31/07/2015 17:38:15
End: 31/07/2015 17:57:42
Clear selection

Highlight local network IPs
45.114.11.34 ×
45.114.11.16 ×
Incoming local Outgoing local

d)
Destination IP — Destination IP — Packets

□	Destination IP	11.5k Packets Total	Percentage
■	192.168.186.18	4.18k packets	36.31%
□	45.114.11.16	2.34k packets	20.32%
□	45.114.11.34	2.09k packets	18.13%
■	192.168.186.168	1.05k packets	9.09%
■	192.168.186.45	624 packets	5.42%
Selected	4	6.23k packets	54.13%

Rows per page: 5 1-5 of 42

f)
Network Graph Packets

192.168.186.45 <--> 192.168.186.168, 1.16k packets
Click to see a detailed view

Packets

Index	Source IP	Destination IP	Source Port	Destination Port	App	Network Protocol	Transport Protocol	Timestamp	Bytes
5902	45.114.11.16	192.168.186.18	37307	22 (ssh)	ssh	ip	tcp	31/07/2015 17:38:16	106
5903	45.114.11.16	192.168.186.18	37307	22 (ssh)	ssh	ip	tcp	31/07/2015 17:38:16	54
5905	45.114.11.16	192.168.186.18	59483	22 (ssh)	ssh	ip	tcp	31/07/2015 17:38:16	74
5908	45.114.11.16	192.168.186.18	59483	22 (ssh)	ssh	ip	tcp	31/07/2015 17:38:16	54
5909	45.114.11.16	192.168.186.18	59483	22 (ssh)	ssh	ip	tcp	31/07/2015 17:38:16	69
5911	45.114.11.16	192.168.186.18	37307	22 (ssh)	ssh	ip	tcp	31/07/2015 17:38:16	54
5913	192.168.186.21	192.168.186.168	0	0		arp		31/07/2015 17:38:16	42
5915	45.114.11.16	192.168.186.18	59483	22 (ssh)	ssh	ip	tcp	31/07/2015 17:38:16	54
5917	45.114.11.16	192.168.186.18	59483	22 (ssh)	ssh	ip	tcp	31/07/2015 17:38:16	702
5919	45.114.11.16	192.168.186.18	59483	22 (ssh)	ssh	ip	tcp	31/07/2015 17:38:16	326

Rows per page: 10 1-10 of 6233

Fig. 1. Dataset Overview visualizes network traffic from packet capture files: (a) One can upload a PCAP file or choose from the previously uploaded datasets (a). Dataset summary (b) shows the sum of packets, IP addresses, connections, and dataset size. Communication profile (c) is the starting point for the analysis. The user can progressively configure up to six filter steps (d) to filter out the data according to multiple consecutive attributes. Users can also export and import the filter configurations or use one of four predefined presets (e). The right part (f)shows the filtering results as a network graph and in tabular form.

summary—Fig. 1(b)—provides an overview of the loaded dataset parameters: a sum of packets, IP addresses, connections and the file size.

Filtering Funnel: The network traffic analysis often requires a combination of multiple filter steps. A common approach is, therefore, to chain multiple filters into one filtering command. We implemented such chained filters using the *filtering funnel metaphor*, a fundamental concept of PCAPFunnel.

PCAP file analysis starts with a statistical overview of traffic properties (e.g., traffic volume, peaks in communication, primary sources or used protocols, and distribution). Filtering is then based on a chain of filter steps where each step allows to filter data using a different property. The output of the preceding filter step becomes the input of the following one. So the data is progressively filtered, resembling a funnel as depicted in Fig. 3. Depending on the analytical goal, the steps can be combined almost arbitrarily from top-down or bottom-up manner, i.e., from application to network-level or vice versa (**R2**).

The traffic volume over time is displayed as a bar chart. The chart also serves as the first filter step, allowing the user to select the time range (Fig. 1(c)). Only the traffic information from the selection is passed to the filter steps below. The user can also indicate local network IP addresses by selecting them in the drop-down dialog.

The filter step (Fig. 1(d)) consists of a table and the traffic volume chart. The table shows packet property statistics (e.g.,

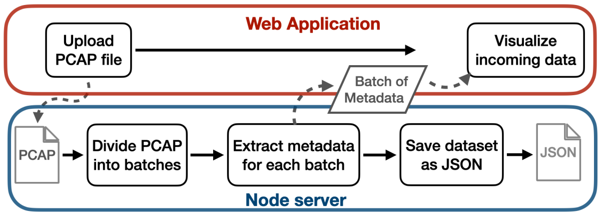


Fig. 2. PCAP file data preprocessing workflow.

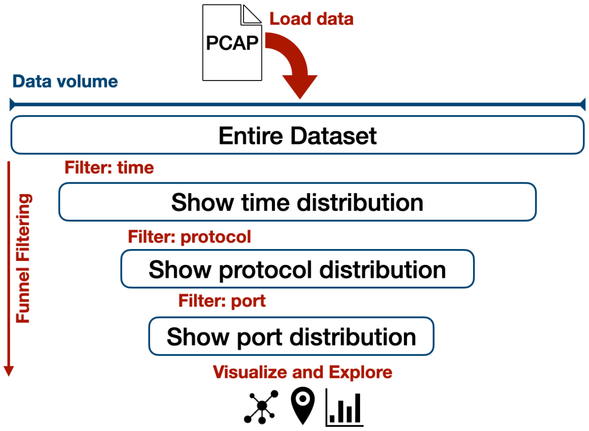


Fig. 3. The *filtering funnel metaphor* illustration. Step-by-step filtering allows to drill down in the data progressively.

a destination IP filter contains a table of all the IP addresses from the selection). The + button allows to add up to six different filter steps corresponding to the following packet properties: IP address (source, destination, or both), network protocol, transport protocol, port (source, destination, or both), and application name. The applied filter steps remain visible. Since they are linked, the change in one instantly affects others.

In addition to an arbitrary combination of filter steps, the user can select from four presets and export or import filtering configurations (Fig. 1(e)). There are two presets for a top-down analysis (i.e., from an application to an IP address) and two in reversed order. The exported files are in JSON format to simplify their sharing with co-workers or reuse on similar datasets (**R5**).

Filtering Results: Communications that pass through all the steps are displayed in the DATASET OVERVIEW right part as a node-link diagram (Fig. 1(f)). The table below lists all the packets and their attributes. Any changes in filtering steps directly affect the displayed results.

The node-link diagram visualizes the communication topology of the filtered data. Each node corresponds to an IP address. Its size is proportional to the volume of send and received data. The link width represents the volume of communication between two nodes.

Clicking on a node or table row opens a new detailed view tab with information concerning the given IP address. Analogically, clicking on a link opens a detailed view tab of the traffic between the two IP addresses. The user can also switch between visualizing packets, bytes, or connections (uni- or bi-directional).

B. Detail View

The DETAIL VIEW (Fig. 4) provides further details of incoming and outgoing communication of the selected IP address or addresses in case of connections (**R3**). The view has four sections: *Communication Profile*, *Network Profile*, *Packet Property Statistics*, and *Raw Data*.

Communications profile: The section contains four mirrored charts, each displaying incoming (top) and outgoing (bottom) traffic (Fig. 4(a)). The charts display the traffic volume over time in numbers of packets, bytes, and flows. The last one is a packet size histogram.

Network Profile: The section (Fig. 4(b)) visualizes the proportional comparison of incoming and outgoing traffic, geolocalized IP addresses, and the list of the countries based on the traffic volume. The geolocalization (**R4**) is provided through GeoLite2 IP [29].

A Sankey diagram displays incoming (left) and outgoing (right) connections based on the number of packets, bytes, or flows. The same color signifies the same source/destination IP. The user can also display only the top 5, 10, 15, 20, or 30 connections. "Others" aggregates the remaining ones.

The minimap shows all geo-localized IP addresses. If both connection endpoints are localized, there is also a link between them. Circle size represents an IP address and is proportional to the sum of incoming and outgoing traffic.

The table on the right side lists the top countries by traffic volume. The user can also switch between the table and choropleth visualization.

All the visualizations are zoomable and draggable. Clicking on a node or connection either in the Sankey diagram or geovisualization opens a new Detailed View tab with the corresponding IP address or connection. So the user can continue further in the drill-down analysis.

Packet Property Statistics: The third section (Fig. 4(c)) presents the most frequent (i.e., top 5/10/15/20/30) values for packet properties: source and destination IP addresses, network protocol, transport protocol, source and destination ports, and the application name. Each property data is visualized in one of seven bar charts.

Raw Data: The last section (Fig. 4(d)) presents the raw data in two tables: *Connections* and *Packets*.

The *Connections* table lists collections of packets with the same transport protocol, source IP and port, destination IP, and port. The table also displays the first packet's timestamp and overall packet count in the connection. The user can switch between uni-directional and bi-directional connections.

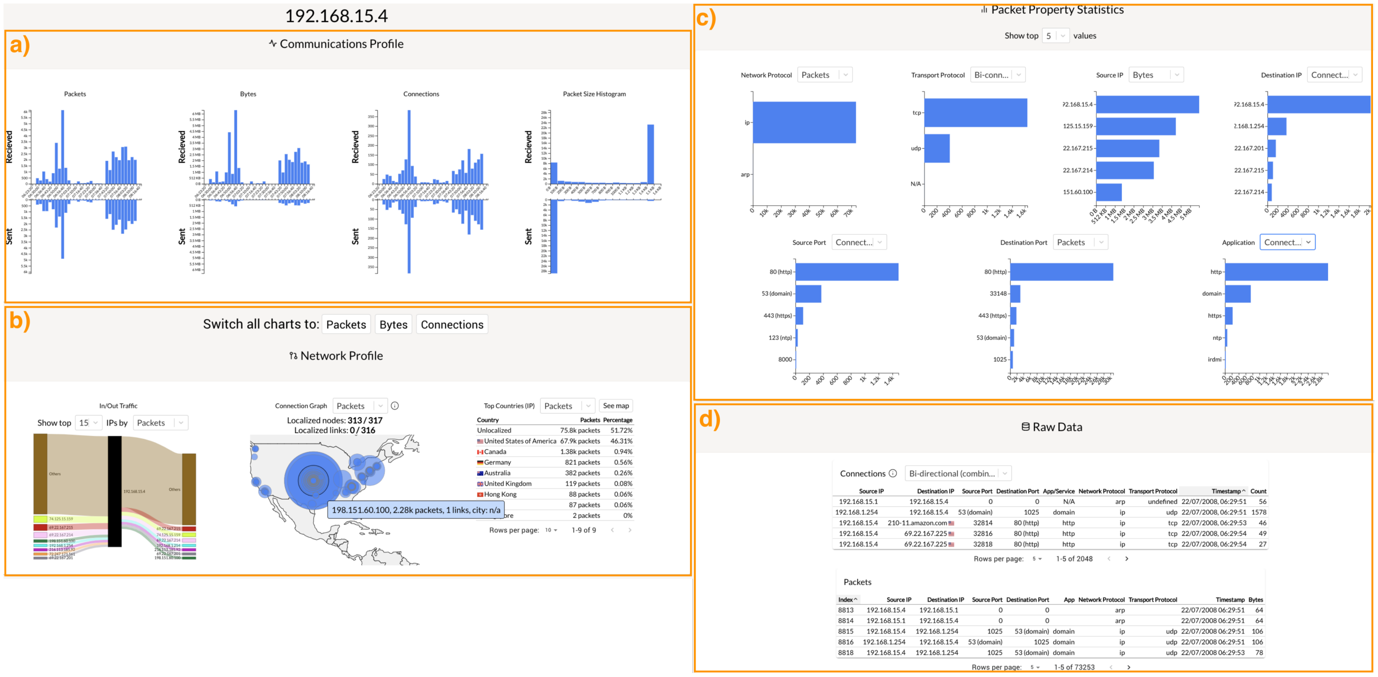
A country flag is shown in the source and destination IP columns for localized IP addresses. Another enhancement is DNS translation which converts IP addresses to resolved domain names (**R4**). IP addresses and hostnames are clickable and open new DETAIL VIEW tab of the clicked IP address. The *Packets* table displays the attributes of corresponding packets that provide the underlying information for all the visualizations in the DETAIL VIEW. Both tables are sortable by clicking on the column headings. A click on the IP address invokes a new DETAIL VIEW tab with corresponding data.

V. USER STUDY

We conducted a qualitative user study of PCAPFunnel with the following goals: a) collect qualitative feedback on the usability and usefulness, b) identify limitations of the tool, and c) assess the fulfillment of the user requirements. Instead of formal usability evaluation or measuring task performance, we sought to observe what aspects of the tool provide the most value to domain experts. The approach is commonly used in projects like ours [30], [31]. Due to Covid-19 pandemic restrictions, we held the study online.

A. Method

Participants: We recruited nine domain experts, all males (26–42 yo). Three of them work in academia as cybersecurity researchers. Two are managers, and four employees in the private sector. All participants have previous experience with computer network security (six over ten years, two 5–10 years, one less than five years). Six participants participated over video calls (Google Meet) which were also recorded. Three (P7–P9) worked asynchronously due to their time restrictions. In both cases, the study design was equal. Measured on a five-point Likert scale (1=novice, 5=expert), all the participants considered themselves as experienced with packet analysis (mean 3.8) and Wireshark (mean 3.7). The participants had



Country	Packets	Percentage
Unlocalized	75.8k packets	51.72%
United States of America	67.9k packets	46.31%
Canada	1.38k packets	0.94%
Germany	821 packets	0.56%
Australia	382 packets	0.26%
United Kingdom	119 packets	0.08%
Hong Kong	88 packets	0.06%
[illegible]	87 packets	0.06%
[illegible]ore	2 packets	0%

Rows per page: 10 1-9 of 9

Connections Bi-directional (combin...

Source IP	Destination IP	Source Port	Destination Port	App/Service	Network Protocol	Transport Protocol	Timestamp ^	Count
192.168.15.1	192.168.15.4	0	0	N/A	arp	undefined	22/07/2008, 06:29:51	56
192.168.1.254	192.168.15.4	53 (domain)	1025	domain	ip	udp	22/07/2008, 06:29:51	1578
192.168.15.4	210-11.amazon.com	32814	80 (http)	http	ip	tcp	22/07/2008, 06:29:53	46
192.168.15.4	69.22.167.225	32816	80 (http)	http	ip	tcp	22/07/2008, 06:29:54	49
192.168.15.4	69.22.167.225	32818	80 (http)	http	ip	tcp	22/07/2008, 06:29:54	27

Rows per page: 5 1-5 of 2048

Packets

Index ^	Source IP	Destination IP	Source Port	Destination Port	App	Network Protocol	Transport Protocol	Timestamp	Bytes
8813	192.168.15.4	192.168.15.1	0	0		arp		22/07/2008 06:29:51	64
8814	192.168.15.1	192.168.15.4	0	0		arp		22/07/2008 06:29:51	64
8815	192.168.15.4	192.168.1.254	1025	53 (domain)	domain	ip	udp	22/07/2008 06:29:51	106
8816	192.168.1.254	192.168.15.4	53 (domain)	1025	domain	ip	udp	22/07/2008 06:29:51	106
8818	192.168.15.4	192.168.1.254	1025	53 (domain)	domain	ip	udp	22/07/2008 06:29:53	78

Rows per page: 5 1-5 of 73253

Fig. 4. Detail View showing communication details of the IP address 192.168.15.4.

also experience with other PCAP analysis tools such as TShark, tcpdump, Moloch, SELK, pyshark, Microsoft Network Monitor, or Netfox Detective.

Procedure and Tasks: We prepared a Google Form that served as the evaluation guide. The form also included the prerecorded video presenting PCAPFunnel features, description of the tasks, and demographic and post-study questionnaires. The three participants who performed the study asynchronously used the form to list the steps they performed to complete each task and for written feedback. Despite being recorded, their responses still provided helpful feedback, and therefore we included them in the evaluation results.

The procedure has three phases: introduction phase, task performing phase, and debriefing phase. In the introduction phase, the participants consented and filled the demographic questionnaire. Next, they watched the presentation of PCAPFunnel features and have a couple of minutes to interact with the tool using a dummy dataset.

The task performing phase (∼40 minutes) consisted of five tasks:

Task 1: Identify the application with the most significant number of connections.

Task 2: Identify two 2 IP addresses that are the most significant sources of the most prominent peak in the traffic.

Task 3: Given the IP address of an infected network station, find out which country the station has communicated with using the HTTPS application protocol.

Task 4: Which network nodes provide DNS service on other ports than 53?

Task 5: Import the provided filter configuration. What is the location of the destination IP address in the resulting connection?

We developed specific tasks rather than open-ended exploration, so the participants tried most PCAPFunnel user interface features. Most of the tasks, however, were possible to accomplish in several ways. Each task was introduced by a brief description providing the context and preparation steps that included, e.g., loading a new dataset for a given task. We asked participants of video calls to think aloud [32] and to rate the perceived difficulty.

In the debriefing phase (10–15 minutes), the participants filled closed-question post-study questionnaire and shared their suggestions and opinions on our tool.

The participants used their computers. Their screen resolutions were FullHD (6×), QHD (2×), or UHD (1×). Since the PCAPFunnel is a web application, they used either recent Google Chrome (7×) or Firefox (2×) browsers.

B. Results

The participants provided helpful feedback about the PCAPFunnel design. Overall, the results are encouraging in the usability and usefulness of the tool.

Tasks: As we can see from Table II, the participants were largely successful when solving the tasks and considered them relatively easy.

Overall, the participants engaged well with the tasks, and we neither received any reservations regarding their purpose or realism level. The participants solved tasks without the interventions of the observers, so the perceived task difficulty was not necessarily correlated with its success rate. Tasks 1 and 5 achieved high success rates and low perceived difficulty. The participants perceived Task 2 as the easiest, despite most of

TABLE II
PERCEIVED TASK DIFFICULTY (7-POINT LIKERT SCALE—HIGHER IS EASIER) AND SUCCESS RATE (RED CELLS INDICATE FAILED TASKS).

Participant	Task 1	Task 2	Task 3	Task 4	Task 5
P1	6	6	4	7	7
P2	7	6	5	5	7
P3	1	7	6	1	7
P4	6	6	5	6	7
P5	6	7	7	7	7
P6	2	6	1	4	7
P7	6	6	1	6	3
P8	5	6	4	7	3
P9	7	7	6	7	7
Mean success rate	89%	44%	89%	66%	100%
Mean task difficulty	5.1	6.3	4.3	5.6	6.1

them submitted only partial answers. Typically, the participants forgot to change one or more filter steps (e.g., they forgot to switch to connections instead of packets), resulting in a slightly different set of resulting IP addresses.

The participants P6 and P7 considered Task 3 as very difficult since they got confused with the choropleth visualization in the DETAIL VIEW. Based on this feedback, we favored table view as the primary and choropleth as the secondary option.

Some confusion arose around the correct interpretation of Task 4, resulting in a lower success rate. It was caused by misinterpretation of the task. Since we revealed the task assignment flaw (there were two possible interpretations) after a couple of sessions, we decided to finish the rest without the change.

Usability and Usefulness: The participants worked with PCAPFunnel without major issues after being given a brief introduction and quickly grasped the filtering funnel metaphor principle.

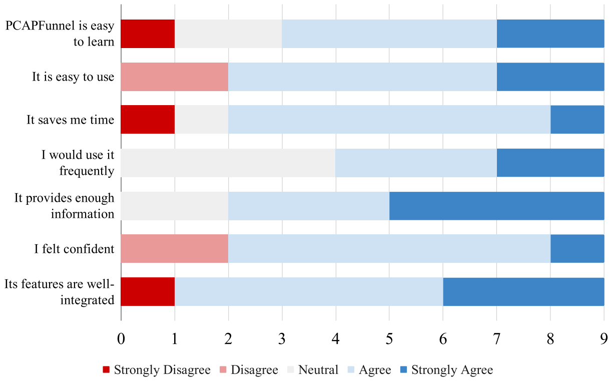


Fig. 5. Answers from the post-study questionnaire.

They were mostly positive (Fig. 5) about its features and perceived it as easy to learn (mean 3.7) and use (mean 3.8). Most of them felt confident when solving the tasks. However, some participants expressed the need for more time to become acquainted (P3: "I would need some more practice with the tool"). Most of them think that PCAPFunnel provides them with enough information (mean 4.2) and that its features are well-integrated (mean 4.0). P4 remarked that "comparing to other tools he uses is [PCAPFunnel] very intuitive." Most of them also indicated they would like to use the application more often (mean 3.8) since it could save their time (mean 3.7). P2 also remarked that "[he] likes its minimalistic design."

VI. DISCUSSION AND FUTURE WORK

The usability study confirmed our initial assumptions that our approach to multi-step filtering could improve the user experience of the top-level PCAP analysis process. Our observations from the study provide several implications for future research.

Desired Features and Improvements: The majority of participants considered the tool as well-designed. However, we also received several suggestions on improvements, some of which we have already integrated into the final design—for example, opening new DETAIL VIEW either by clicking on the table row or in visualizations. We also revealed and fixed few minor bugs in the implementation and slightly changed the terminology used for several visualizations.

Few participants often expressed the need for more filtering capabilities, such as allowing negative filters by using the logical NOT operator (P3) or the possibility to enter the filter as a text input (P5). Additional configuration features include optional DNS resolving (P1) or anonymization of imported data (P2).

The truth is that we intentionally omitted many of these features to keep the prototype compact, but we plan to integrate them in the next versions of the tool.

Using PCAPFunnel for Training and Education: Few participants remarked that the tool is so easy to work with that it could be used during the network administration training. As we already mentioned, current tools for network analysis usually provide only limited visualization capabilities and are hard to learn. PCAPFunnel could support teachers in explaining various phenomena, identifying communication patterns, or demonstrating network analysis tasks during the classes. Moreover, the use of familiar visualizations makes the tool also suitable for novice users.

Generalizability: Currently, our prototype works only with PCAP files. However, the filtering funnel metaphor is generic and can be used with other data sources (e.g., NetFlow or logs). We would also like to extend export formats to PDF files to document or present the analysis results. However, making PCAPFunnel part of the analysts' toolkit would require its integration with other deep packet inspection tools. The trivial way is to create new filter export in a format comprehensible for Wireshark or similar applications. The more ambitious way is to provide integration with deep packet inspection tools and extend the PCAPFunnel user interface, so the user does not need to switch the tools. Such integration would allow seamless transition from top-level to in-depth analysis and back.

Performance Improvements: One of the current weaknesses of PCAPFunnel is the need to upload the entire PCAP file to the server before the progressive data processing starts. We plan to improve our preprocessing to start as soon as the first packets are uploaded, similarly to the approach used by NetCapVis [6].

VII. Conclusion

The traditional network traffic analysis tools have a steep learning curve and only limited visualization capabilities. To facilitate and speed up the process, we designed PCAPFunnel—a tool for packet capture analysis. Inspired by previous work and based on tight collaboration with domain experts, we defined user tasks and design requirements.

We proposed the *filtering funnel metaphor* to support filtering and data analysis based on linking several independent filter steps where the output from one is another's input. All the filter steps are permanently visible to the user, who can interactively modify filters' parameters. With this new approach, a domain expert can quickly check the data's content, determine its structure, analyze network actors' behavior, or reveal the cause of network issues. The user interface leverages linked views and conventional visualizations, so it is also suitable for novice users.

We further performed the user study with nine domain experts in the field of network data analysis. The participants appreciated that PCAPFunnel is easy to learn and use and considered the proposed method flexible and supportive during the initial packet capture data analysis. Likewise similar user studies, we admit that even our study has several limitations. Namely, a low number of participants and its qualitative focus. A more extensive deployment in a real-world setup could provide new insights. Also, a comparative study with the commonly used tools could be valuable. We hope that PCAPFunnel will also inspire novel approaches for network traffic analysis based on interactive visualizations.

Acknowledgment

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

References

[1] R. Wang *et al.*, "Knight's Tour-based Fast Fault Localization Mechanism in Mesh Optical Communication Networks," *Photonic Network Communications*, vol. 23, no. 2, pp. 123–129, 2012.

[2] R. Ball, G. A. Fink, and C. North, "Home-centric Visualization of Network Traffic for Security Administration," in *International Workshop on Visualization for Cyber Security*, C. E. Brodley, P. Chan, R. Lippmann, and W. Yurcik, Eds. ACM, 2004, pp. 55–64.

[3] G. Combs *et al.*, "Wireshark: A Network Protocol Analyzer," 2008. [Online]. Available: http://www.wireshark.org/

[4] E. W. Bethel, S. Campbell, E. Dart, K. Stockinger, and K. Wu, "Accelerating Network Traffic Analytics Using Query-Driven Visualization," in *2006 IEEE Symposium On Visual Analytics Science And Technology*, 2006, pp. 115–122.

[5] J. R. Goodall, "Visualization is Better! A Comparative Evaluation," in *6th International Workshop on Visualization for Cyber Security*, 2009, pp. 57–68.

[6] A. Ulmer, D. Sessler, and J. Kohlhammer, "NetCapVis: Web-based Progressive Visual Analytics for Network Packet Captures," in *2019 IEEE Symposium on Visualization for Cyber Security (VizSec)*, 2019, pp. 1–10.

[7] M. łgorzata Steinder and A. S. Sethi, "A Survey of Fault Localization Techniques in Computer Networks," *Science of computer programming*, vol. 53, no. 2, pp. 165–194, 2004.

[8] G. Harris and M. Richardson, "PCAP Capture File Format," Working Draft, IETF Secretariat, Internet-Draft draft-gharris-opsawg-pcap-01, December 2020. [Online]. Available: http://www.ietf.org/internet-drafts/draft-gharris-opsawg-pcap-01.txt

[9] B. Claise, "Cisco Systems NetFlow Services Export Version 9," Internet Requests for Comments, RFC Editor, RFC 3954, October 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3954.txt

[10] G. A. Pimenta Rodrigues *et al.*, "Cybersecurity and Network Forensics: Analysis of Malicious Traffic Towards a Honeynet with Deep Packet Inspection," *Applied Sciences*, vol. 7, no. 10, p. 1082, 2017.

[11] C. Guo *et al.*, "Pingmesh: A large-scale System for Data Center Network Latency Measurement and Analysis," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 139–152.

[12] M. Holkovič and O. Ryšavý, "Using Rule-Based Decision Trees for Automatic Passive Diagnostics of the Network Problems," *International Journal on Advances in Networks and Services*, vol. 2020, no. 1, pp. 1–10, 2020.

[13] Flowmon Networks a.s., "Flowmon Packet Investigator," 2020. [Online]. Available: https://www.flowmon.com/cs/products/software-modules/packet-investigator

[14] P. Haag, "Nfdump – netflow processing tool," 2007. [Online]. Available: https://github.com/phaag/nfdump

[15] V. Ndatinya *et al.*, "Network Forensics Analysis Using Wireshark," *International Journal of Security and Networks*, vol. 10, no. 2, pp. 91–106, 2015.

[16] Netresec AB, "NetworkMiner." [Online]. Available: https://www.netresec.com/?page=networkminer

[17] M. Solé *et al.*, "Survey on Models and Techniques for Root-cause Analysis," *arXiv preprint arXiv:1701.08546*, 2017.

[18] H. Shiravi, A. Shiravi, and A. A. Ghorbani, "A Survey of Visualization Systems for Network Security," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 8, pp. 1313–1329, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/tvcg/tvcg18.html#ShiraviSG12

[19] H. Tang, C. Han, and J. Ge, "Applications of Visualization Technology for Network Security," in *TrustCom/BigDataSE/ICESS*. IEEE Computer Society, 2017, pp. 1038–1042. [Online]. Available: http://dblp.uni-trier.de/db/conf/trustcom/trustcom2017.html#TangHG17

[20] NSA, "GRASSMARLIN," 2017. [Online]. Available: https://github.com/nsacyber/GRASSMARLIN

[21] B. C. M. Cappers and J. J. van Wijk, "SNAPS: Semantic Network Traffic Analysis Through Projection and Selection," in *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, L. Harrison, N. Prigent, S. Engle, and D. M. Best, Eds. IEEE Computer Society, 2015, pp. 1–8.

[22] B. C. M. Cappers *et al.*, "Eventpad: Rapid Malware Analysis and Reverse Engineering using Visual Analytics," in *2018 IEEE Symposium on Visualization for Cyber Security (VizSec)*. IEEE, 2018, pp. 1–8.

[23] T. Taylor, D. Paterson, J. Glanfield, C. Gates, S. Brooks, and J. McHugh, "FloVis: Flow Visualization System," in *2009 Cybersecurity Applications Technology Conference for Homeland Security*, 2009, pp. 186–198.

[24] Costa, Gianluca, "CapAnalysis," 2018. [Online]. Available: http://www.capanalysis.net/

[25] A-Packets, "Online PCAP file analyzer designed to visualize HTTP, Telnet, FTP," 2020. [Online]. Available: https://apackets.com/

[26] PacketTotal, "Simple, free, high-quality PCAP analysis." [Online]. Available: https://packettotal.com/

[27] "Service Name and Transport Protocol Port Number Registry." [Online]. Available: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

[28] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," in *Proceedings 1996 IEEE Symposium on Visual Languages*, 1996, pp. 336–343.

[29] "GeoLite2 Free Downloadable Databases." [Online]. Available: https://dev.maxmind.com/geoip/geoip2/geolite2/

[30] S. Carpendale, *Evaluating Information Visualizations*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 19–45. [Online]. Available: https://doi.org/10.1007/978-3-540-70956-5_2

[31] H. Lam *et al.*, "Empirical Studies in Information Visualization: Seven Scenarios," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1520–1536, 2012.

[32] C. Lewis and J. Rieman, *Task-Centered User Interface Design*. University of Colorado, Boulder, Department of Computer Science, 1993.