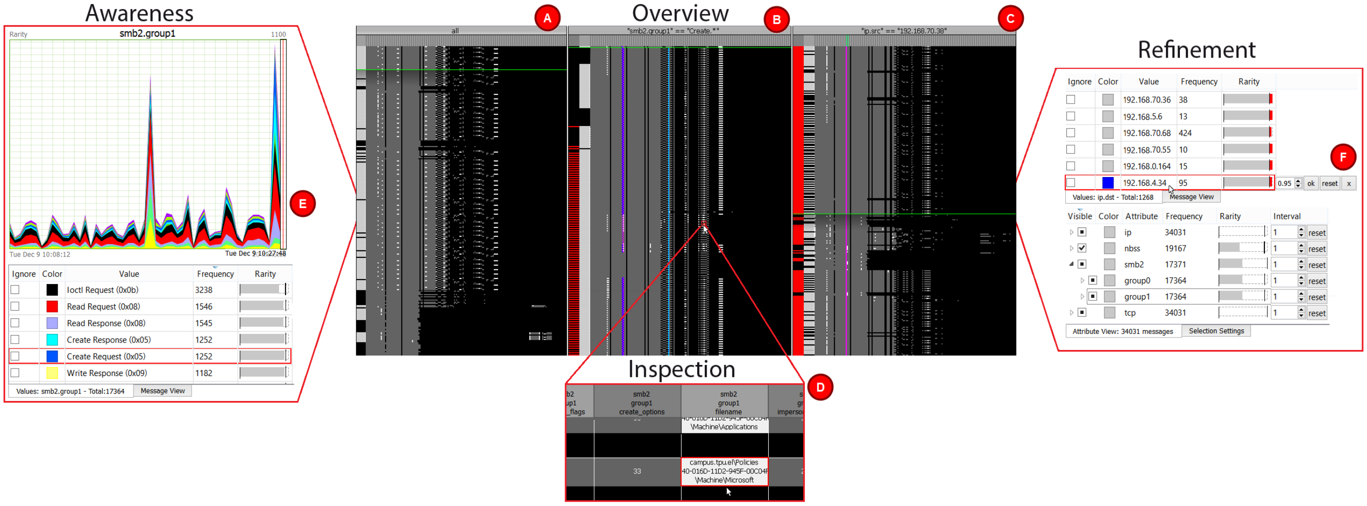
SNAPS:
Semantic Network traffic Analysis through Projection and Selection

Bram C.M. Cappers, Jarke J. van Wijk



Ignore	Color	Value	Frequency	Rarity
□	⬛	Ioctl Request (0x0b)	3238	
□	🟥	Read Request (0x08)	1546	
□	🟪	Read Response (0x08)	1545	
□	🟦	Create Response (0x05)	1252	
□	🟦	Create Request (0x05)	1252	
□	🟨	Write Response (0x09)	1182	

Values: smb2.group1 - Total:17364 | Message View

Ignore	Color	Value	Frequency	Rarity				
□	■	192.168.70.36	38					
□	■	192.168.5.6	13					
□	■	192.168.70.68	424					
□	■	192.168.70.55	10					
□	■	192.168.0.164	15					
□	🟦	192.168.4.34	95		0.95	ok	reset	x

Values: ip.dst - Total:1268 | Message View

Visible	Color	Attribute	Frequency	Rarity	Interval	
▷ ■	■	ip	34031		1	reset
▷ ☑	■	nbss	19167		1	reset
◢ ■	■	smb2	17371		1	reset
▷ ■	■	group0	17364		1	reset
▷ ■	■	group1	17364		1	reset
▷ ■	■	tcp	34031		1	reset

Attribute View: 34031 messages | Selection Settings

Fig. 1. Network traffic exploration at the level of semantics through the creation of three selections of interest in parallel.

Abstract—Most network traffic analysis applications are designed to discover malicious activity by only relying on high-level flow-based message properties. However, to detect security breaches that are specifically designed to target one network (e.g., Advanced Persistent Threats), deep packet inspection and anomaly detection are indispensible. In this paper, we focus on how we can support experts in discovering whether anomalies at message level imply a security risk at network level. In SNAPS (Semantic Network traffic Analysis through Projection and Selection), we provide a bottom-up pixel-oriented approach for network traffic analysis where the expert starts with low-level anomalies and iteratively gains insight in higher level events through the creation of multiple selections of interest in parallel. The tight integration between visualization and machine learning enables the expert to iteratively refine anomaly scores, making the approach suitable for both post-traffic analysis and online monitoring tasks. To illustrate the effectiveness of this approach, we present example explorations on two real-world data sets for the detection and understanding of potential Advanced Persistent Threats in progress.

Index Terms—Anomaly detection, network traffic analysis, multivariate analysis, streaming data, interaction, parse data analysis.

1 Introduction

One of the main challenges in the area of network traffic analysis is how to detect when a network is being exploited. Especially for critical infrastructures, such as power plants [4], hackers nowadays are willing to design complex viruses to maximize the damage in one specific infrastructure. The main difficulty with Advanced Persistent Threats (APTs) [25] is the involvement of domain knowledge such that their traffic can no longer be distinguished from regular activity by simple inspection of high-level properties, such as message length and destination address. Current methods [7, 11, 12, 17] focus on the analysis of these properties, because in practice they have shown to be sufficient for the discovery of traditional attacks [8, 16]. The fact that these techniques consider traffic content as a black box makes them unaware of anomalies at the level of semantics.

To ensure that systems and data in the network are secure from APTs, the content of the traffic has to be taken into account. For example, we are not only interested in which host sends packets to a particular host, but we are also interested in whether the action inferred by these messages represents the access to an uncommon function call or file in the network. The heterogeneity and abstraction level of the data makes it very difficult to decide if a message is truly malicious. We believe that the greatest insights can be obtained by comparing anomalies to similar parts of traffic and try to understand how they differ from each other with respect to context and structure. In order to gain this insight, we propose a new approach that enables security experts to discover high-level security risks, starting from a collection of automatically classified low-level anomalies, through the use of selection and projection. More specifically, our main contributions are:

- A novel exploration method for the analysis of raw network traffic, enabling the expert to inspect and compare specific parts of the traffic in parallel while preserving context;
- A tight coupling of machine learning and visualization that assists experts in detecting malicious traffic, through iterative refinement of classifier parameters;
- The ability to gain statistical insight in how messages differ from regular traffic and why a message was classified as malicious.

- *Bram C.M. Cappers is with the Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands. E-mail: b.c.m.cappers@tue.nl.*
- *Jarke J. van Wijk is with the Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands. E-mail: j.j.v.wijk@tue.nl.*

978-1-4673-7599-3/15/$31.00 ©2015 IEEE

The paper is structured as follows. First, related work is discussed in Section 2. Next, the scope and approach for traffic analysis is discussed in Section 3. In Sections 4, 5, and 6 an overview of the system is presented after which visualization, classification, and interaction are described. In Sections 7 and 8 we provide two example explorations on real-world data sets and discuss the limitations of the approach. Conclusions and future work are presented in Section 9.

2 Related Work

Network traffic analysis is an extensively studied topic, covering a wide range of techniques. We give a broad overview first, followed by a detailed discussion on pixel-based visualization techniques.

2.1 Data

From a data perspective, current analysis techniques can be grouped into two categories: *byte-oriented* and *attribute-oriented* analysis.

In byte-oriented analysis, network messages are considered as a sequence of bytes enabling visualization techniques to analyze the full payload of a network message. These visualizations typically provide insight in the traffic by encoding the byte sequences in text or pixels. The binary rainfall [7], digraphs [6] and malware images [19] are well-known examples in this category. Anomaly detection systems for this type of analysis typically rely on byte distributions and pattern matching to discover undesired content. Since byte sequences do not contain any information about which bytes together represent an attribute in a message, these detection methods often work poorly for anomalies at the level of attributes.

In attribute-oriented analysis, messages are dissected according to their *protocol* structure, thereby gaining knowledge about the actual values that were sent in the message. The result of dissecting a message typically is a collection of attributes and values. The presence of an attribute or value is determined by the type of network message, thereby significantly increasing the heterogeneity of the data. Current methods often limit their analysis to high-level protocols such as TCP and IP, thereby only relying on common flow-based attributes such as IP addresses, port numbers, and message lengths [11, 15, 27]. For a more complete overview, see the survey paper of Shiravi et al. [23].

There are also examples where both byte structure and attribute analysis are taken into account. For instance, the open source application Wireshark [5] is an extensive protocol analyzer that can dissect network packets and display the payload in a (hierarchical) textual representation. Especially for debugging applications, the wealth of information provided by Wireshark can help the expert to analyze traffic in great detail. The software unfortunately does not assist the expert in finding anomalies and can become a burden when analyzing or monitoring large network samples.

2.2 Visualization

In SNAPS we use a pixel-based visualization that conveys the global structure of network messages as well as anomalies in that structure. A message is displayed as a horizontal sequence of pixels. Pixel-based visualizations have been used often for network traffic analysis, some examples are:

- Binary rainfall [7] by Conti et al. visualizes network messages as a single line of pixels where pixels are colored based on protocol type, various byte encodings and frequency. They showed that the visual encoding of network traffic does not have to be complex in order to discover nontrivial patterns. Their byte-oriented approach unfortunately makes the method unsuitable for the detection of APTs.
- PortVis [17] by McPherson et al. uses a color-based grid visualization to visualize the amount of network activity between port numbers. By using a zoom lens, the user can obtain port number information to trace back the cause of the anomaly.
- IDS rainstorm [1] by Abdullah et al. visualizes Stealthwatch [13] intrusion detection alerts by showing the severity of alerts over time using a set of rectangular regions that represent a large continuous range of IP addresses.

Previous methods construct an image to represent the values for one or two attributes in the data. To cover the wide variety of attributes, in SNAPS we construct an image to represent the full range of attributes.

A method specifically designed for multivariate data exploration and closest to our technique is the Pixel Carpet visualization by Landstorfer et al. [14]. In this visualization every log record is visualized as a stack of pixels, where every pixel denotes the frequency of a value in that record. By means of filtering, uninteresting records can be removed from the data, after which the frequencies of the remaining records are updated. Although our visualization method is similar to the stacked pixel approach in the Carpet visualization, there are differences. First, the Carpet visualization is limited to a single view, indicating that it is impossible for experts to zoom in on a specific subset without losing context of other activities over time. We provide a time view to maintain awareness of temporal patterns and enable experts to duplicate pixel views before applying new filters. Second, the tight integration of filtering and recomputing statistics causes the frequency analysis to overfit the data when filters become too specific. In SNAPS, experts can refine classifiers when necessary or train a new classifier on a subset of the data. Third, Landstorfer et al. already indicate that their method is designed to work for a low number of attributes, while our approach is designed to work for hundreds of attributes. Finally, our selections of interest enable experts to construct more complex queries using boolean search and regular expressions.

In summary, current methodologies are either focused on the visualization of high-level message attributes or the visualization of unstructured low-level representations. Current methods that do consider message attributes typically consider only a few flow-based attributes.

3 Problem statement

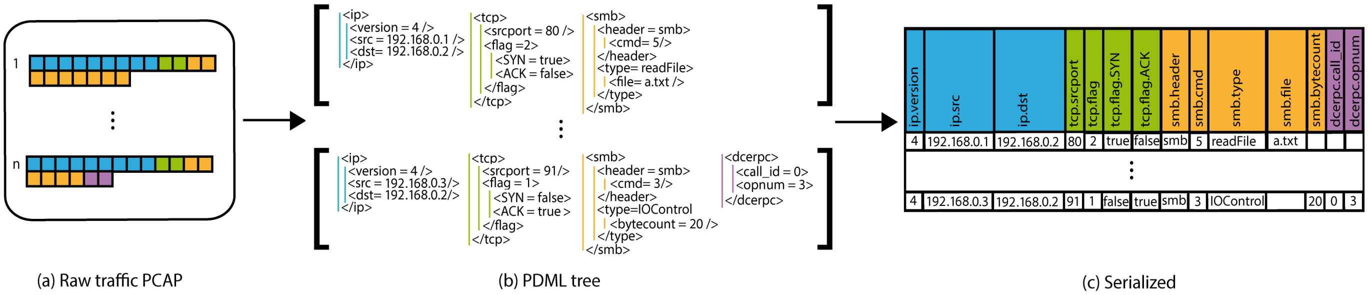
With the vast amount of information that is sent over networks, one of the main concerns is to know when something undesired is being sent. Especially for critical infrastructures, the presence of malicious traffic can have severe if not life-threatening consequences.

The involvement of domain knowledge in APTs makes the infiltration of these viruses (typically through social engineering) in networks nearly impossible to prevent. Once the threat is established, we can analyze unencrypted internal network traffic for anomalies that arise during the APTs exploitation phase. As a consequence, traffic from or to external sources is considered outside this scope.

In practice it is possible for messages to consist of an arbitrary number of protocols, where one protocol can even occur multiple times. Since semantic attacks happen at the application layer of the network protocol, we restrict the analysis of messages to the following protocols: (DCE)RPC, SMB2 and S7 [24]. (DCE)RPC are application protocols to send remote procedure calls over a network. SMB2 is typically used for file management in a network, whereas S7 is a classified industrial control protocol by Siemens for controlling low-level hardware components. We use ETH, IP, and TCP protocol information to trace the anomalies back to physical entities in the network. To avoid the significant increase in attribute space due to protocols occurring multiple times, analysis of messages is limited to the first occurrence of every protocol.

3.1 Data acquisition

Before we can analyze network traffic in greater detail, we first need to analyze network traffic with protocol semantics. We use the Wireshark dissector to convert a raw network message to a so called Packed Detail Markup Language (PDML) parse tree, describing on a per protocol basis the values and attributes that are present in that message. Figure 2b shows an illustration of how PDML trees are structured. Attributes in a protocol are structured hierarchically. In general, network messages consist of multiple protocols, each with their own purpose and different level of abstraction. Depending on the protocol semantics, attributes in protocols can represent numerical ranges (e.g., `tcp.srcport`), strings (e.g., `ip.src`), or boolean values (e.g., `tcp.flag.SYN`). The presence of a protocol, attribute, or value not only depends on the type of message, but also depends on the context in which the message was sent.



(a) Raw traffic PCAP

```
<ip>
 <version = 4 />
 <src = 192.168.0.1 />
 <dst= 192.168.0.2 />
</ip>
<tcp>
 <srcport = 80 />
 <flag =2>
  <SYN = true>
  <ACK = false>
 </flag>
</tcp>
<smb>
 <header = smb>
  <cmd= 5/>
 </header>
 <type= readFile>
  <file= a.txt />
 </type>
</smb>

⋮

<ip>
 <version = 4 />
 <src = 192.168.0.3/>
 <dst= 192.168.0.2/>
</ip>
<tcp>
 <srcport = 91/>
 <flag = 1>
  <SYN = false>
  <ACK = true >
 </flag>
</tcp>
<smb>
 <header = smb>
  <cmd= 3/>
 </header>
 <type=IOControl
  <bytecount = 20 />
 </type>
</smb>
<dcerpc>
 <call_id = 0>
 <opnum = 3>
</dcerpc>
```

(b) PDML tree

ip.version	ip.src	ip.dst	tcp.srcport	tcp.flag	tcp.flag.SYN	tcp.flag.ACK	smb.header	smb.cmd	smb.type	smb.file	smb.bytecount	dcerpc.call_id	dcerpc.opnum
4	192.168.0.1	192.168.0.2	80	2	true	false	smb	5	readFile	a.txt			
⋮													
4	192.168.0.3	192.168.0.2	91	1	false	true	smb	3	IOControl		20	0	3

(c) Serialized

Fig. 2. Data acquisition by dissecting PCAP packets (a) to PDML trees (b) after which they are serialized into one (sparse) multivariate table (c).

More formally, let S represent the set of attributes that the expert wants to analyze. Furthermore, let $PDML(m)$ represent the PDML tree of message m. Attribute $A \subseteq PDML(m)$ if and only if there exists a path $P = [p_0, p_1, \ldots, p_n]$ from root to leaf in $PDML(m)$ such that $p_0.p_1.\cdots.p_n = A$. A is also referred to as the *serialization* of P. Finally, let $val(A)$ represent the value stored in p_n of A. We can now model a message m as a set of (A, v) pairs such that:

$$A \in S \text{ , and} \tag{1}$$

$$v = \begin{cases} val(A) & \text{if } A \subseteq PDML(m) \\ \textbf{undefined} & \text{otherwise} \end{cases} \tag{2}$$

Since the set of all possible attributes is too large to analyze, in SNAPS we use domain knowledge and a large sample of network traffic to determine which attributes are worth analyzing. The resulting table after serializing the collected PDML trees can be found in Figure 2c. Since the set of possible attributes is much larger (500 or more) than the set of attributes contained in a message (order of 10s), the data can become quite sparse, increasing the complexity of visualizing the payload data.

Using our previous model, we can now formulate the analysis task for the detection of APTs as trying to gain insight in the presence, description, temporal behavior, and rarity of these (attribute,value) pairs. The serialization of PDML trees to one multivariate table enables us to compare differences and similarities between messages at the level of attributes. Since the presence of one attribute depends on the presence of other attributes, showing values of multiple attributes simultaneously enables us to gain insight in these dependencies. In order to make the visualization of alerts and patterns feasible for a large number of attributes, we chose for a pixel-based visualization approach.

4 SNAPS: Selection and Projection

Network traffic exploration is a challenge due to the large amount of data that is being generated in a relatively short period of time. Furthermore, the heterogeneity and complex structure of the traffic content adds a new dimension to the analysis of network traffic. We cannot expect the expert to know the meaning of every dissected value or attribute. However, the expert should be able to determine the severity or cause of an anomaly by *inspecting* and *comparing* similar type of messages in different contexts. To support this, we need a scalable interactive method to simultaneously explore low-level anomalies, while maintaining a high-level *overview*.

We tackle the scalability problem by visualizing network traffic using a pixel map [10]. The high "data-to-space" ratio of pixel maps enables us to visualize large amounts of attributes and network messages in a limited amount of screen space. Furthermore, to maximize the speed of analyzing traffic, we aim for a computationally cheap classification method using histograms. The level of granularity in which traffic is analyzed is determined by filtering on attributes or values in the traffic. In the world of relational algebra [2], these operations are referred to as *projection* and *selection* respectively. To enable the simultaneous exploration of traffic in local and global contexts, we do not limit the exploration to one selection, but to a number of selections of interest (see Section 4.1.2) enabling the expert to:

- *Drill down*: inspecting alerts against different subparts of the network, while remaining aware of the rest of the traffic, or
- *Scatter*: creating multiple views to keep an eye on critical or suspicious entities in the network (e.g., hosts, files etc.).

To tackle the problem of dealing with large amounts of false positive alerts, we use a human-in-the-loop approach [21] that enables the expert to inspect and *refine* classification results on a per selection basis. By means of *color rules*, the expert is able to highlight specific events in the traffic for which the severity is already known. Figure 3 shows a schematic overview of the SNAPS exploration process. When trying to find potential virus attacks, time is of the essence. The earlier anomalies in the network can be detected, the faster we are able to manage the attack. For this reason, we designed the system in such a way that it is suitable for both post-traffic analysis and live monitoring. Although the traffic dissection by Wireshark is rather computationally intensive to be used for real-time monitoring, there are (more complex) alternatives, like the Bro dissector [20], that are suitable for obtaining near-real time dissections. To assist the expert in exploring and explaining traffic alerts, we use five coordinated views as depicted in Figure 5. For each view we describe its functionality and design decisions. For the demonstration of the functionality in practice, we refer to the supplementary video [1].

4.1 Pixel viewer

For every selection of interest, the pixel viewer visualizes message payload by creating an image where the horizontal axis represents the attribute space the expert is interested in and the vertical axis represents the collection of network messages. The result is that every message corresponds to a single line of pixels, where the brightness of a pixel p_{ij} represents the rarity of message i at attribute j according to Section 5. Since it is hard to distinguish colors for small objects [26], we use a discrete gray scale map (Figure 5) consisting of three colors: pixels are colored black if the message does not contain the corresponding attribute, gray if the value in that message is not considered rare and white if the value in the message is considered to be rare. The rarity of a message as a whole is visualized by prepending the image with an additional column. Values and attributes in messages become visible by inspecting pixels with a zoom lens (Figure 5b). Besides the grey shades that indicate rarity, a subtle hue can be added to message attributes to indicate different protocols (Figure 4). Besides coloring attributes, we enable the expert to discover patterns by coloring pixels according to their value or more complex expressions. To improve the distinction between pixels and prevent pixel colors from spreading to their neighboring cells, tiles of 2 by 2 physical pixels are used instead. As soon as an incoming message adheres to some color rule r, SNAPS creates a marker in front of the pixel view whose color corresponds to r. In situations where multiple color rules apply, SNAPS creates a marker according to the first matching color rule. Figure 5 shows an example how coloring is applied.

[1] https://www.youtube.com/watch?v=aYywTOYjYDA

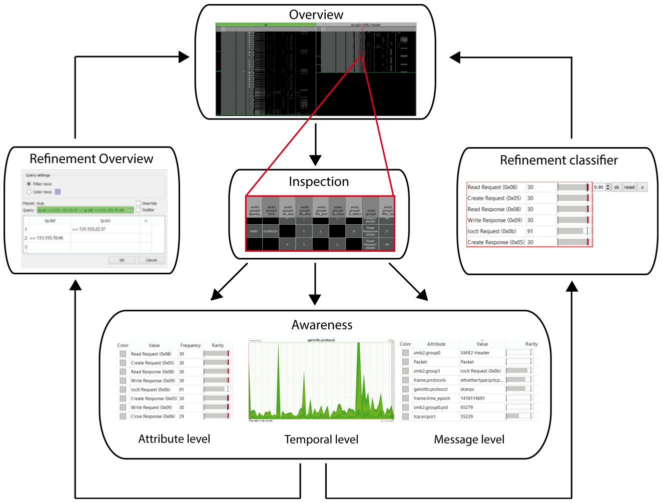


Fig. 3. The SNAPS workflow model for network traffic exploration. The expert uses the overview to monitor the presence of alerts in the selections of interest. Upon the discovery of an alert, the expert tries to gain more insight by inspecting its location and value. Depending on the familiarity and rarity of the inspected value(s) and attribute(s), the expert assesses the severity of the alert by iteratively inspecting: the occurrence of other values in the message; the value distribution of an attribute; or the presence of a message value over time. Depending on his findings, the expert either decides to ignore the alert, prevent the alert from happening by refining the classifier, or refining his selections of interest to analyze the alert in a different context.

Similar to the pixel visualization by Conti et al. [7], we use the notion of a radar to replace old messages with new ones. In contrast to traditional scrolling, previous messages are not shifted at the arrival of new data, thereby making the visualization more stable when analyzing traffic at a larger pace. A direct consequence of visualizing messages in a sequential fashion is that the inter arrival time between messages is no longer visible. To make the expert aware of these changes in flow, the radar creates a green marker in front of the image whenever the timestamp difference between messages is larger than a second. The distance between green markers is an indicator for the amount of traffic that is being sent in between timestamps. For more detailed information about temporal behavior, the expert can use the time view (Section 4.2).

4.1.1 Attribute ordering

When combining multiple PDML trees into one attribute space, the ordering in which attributes should be positioned is not uniquely defined. To illustrate the latter, consider two message m_1 and m_2 with attribute sequences $[\ A, B, C\ ]$ and $[\ A, B, D\ ]$ respectively. Although the ordering of attributes in a message depends on its structure, when combining the attribute space of two messages into one, it is undefined whether attribute C should precede D or vice versa. Although this ordering does not influence the classification result of a message, it can help the expert to localize attributes in the visualization more quickly. One way to solve this is to sort attributes alphabetically. Since the hierarchy is implicitly stored in the attributes, sorting attributes alphabetically causes the attributes with the same PDML paths to be grouped together. Any logical ordering between siblings (e.g., header attributes before payload), unfortunately, may no longer be preserved. To solve the second issue, we sort the siblings within each group according to their frequency. The effect of sorting attributes with and without frequency analysis is illustrated in Figure 4.

4.1.2 Selections and Projections

As mentioned earlier, selection and projection enable the expert to inspect anomalies against different parts of traffic while focusing on a specific subset of attributes. A downside of applying filters, is that the expert is no longer aware of alerts that were present in previous settings. One can imagine, however, that an expert wants to keep an eye on specific entities in the network (e.g., a critical host), while staying aware of other activities. To prevent the expert from losing context, we enable the expert to create multiple selections of interest by creating multiple pixel views in parallel.

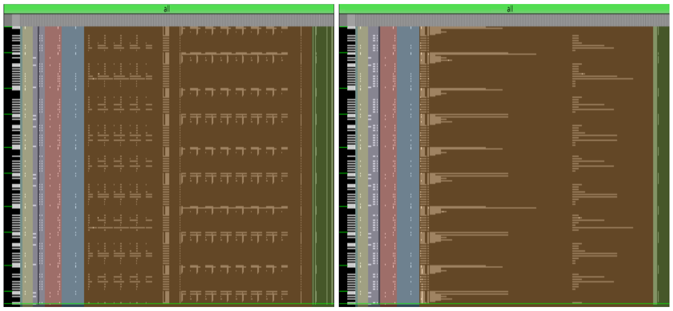


Fig. 4. Attribute ordering without (left) and with (right) frequency clustering. Attributes are colored according to their protocol.

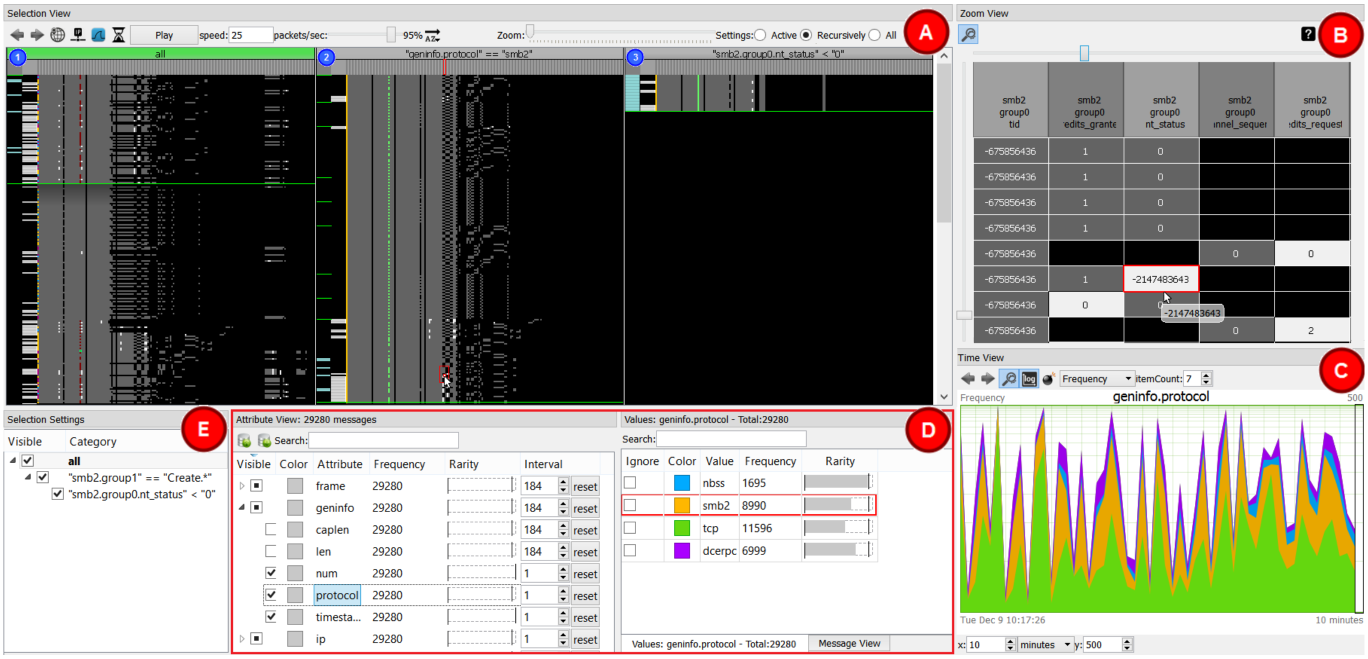
When creating a new view B, by default, projection, selection, and color settings are inherited from view A where the filtering was initiated. This enables the expert to continue the exploration without having to reapply every setting in the new view. The histograms for view B are constructed by revisiting the network traffic within data window ω, only considering messages that are valid with respect to the current selection. Network messages in B are visualized in the pixel view if and only if these messages are also visually present in A. This way, messages in B initially are always a subset of the messages in A enabling the expert to see the impact of applying a new selection of interest. By means of the time view, the expert can revisit earlier parts of traffic that are outside the scope of the pixel view.

In order to gain insight in anomalies within a specific selection of interest, the expert is enabled to train a separate classifier for that selection. Since highly specific selections may result in inaccurate alerts due to overfitting, by default, alerts with respect to the histograms of the parent view are shown in the visualization. Since both anomaly scores are maintained in parallel, the expert can toggle between *local* and *global* anomaly scores. An overview of the current selections is shown by means of a tree structure (see Figure 5e). The expert can add, remove, show or hide selections whenever necessary. We enable the expert to apply new settings simultaneously to all views, the selected view or the selected view along with its descendants.

4.2 Time view

The time view shows an overview of the number of messages that are sent over the last n time units. By selecting an attribute A, the expert is enabled to inspect the distribution of the values of A over various periods in time. Depending on the selected pixel view, only messages that are valid with respect to that selection are shown in the time view. Upon the arrival of new data, the line chart is shifted to the left, causing messages older than n time units to be no longer visible. To prevent the chart from cluttering, only the top m values in A are shown that are either most frequent, most rare, or selected by the expert. Remaining values are grouped in a miscellaneous category.

The time view enables the expert to scroll back to earlier parts of traffic. To make the expert aware of the time interval that is spanned between the oldest and the newest message in the selected pixel view, a black window is rendered in the time view. Since the time interval of the pixel view depends on the inter arrival time between messages in that view, the width of the visualization window may vary over time. The expert can scroll back to earlier parts of traffic by dragging the visualization window along the time axis after which the selected view and all descendants are updated. Here traditional scrolling is preferred over a radar, since experts can determine their own rate in which the views have to be updated.



speed: 25 packets/sec: 95% Zoom: Settings: ○ Active ● Recursively ○ All

"geninfo.protocol" == "smb2"

"smb2.group0.nt_status" < "0"

smb2 group0 tid	smb2 group0 edits_grante	smb2 group0 nt_status	smb2 group0 innel_sequer	smb2 group0 dits_request
-675856436	1	0		
-675856436	1	0		
-675856436	1	0		
-675856436	1	0		
-675856436			0	0
-675856436	1	-2147483643		
-675856436	0	0		
-675856436			0	2

Frequency itemCount: 7

x: 10 minutes y: 500

Visible	Category
☑	all
☑	"smb2.group1" == "Create.*"
☑	"smb2.group0.nt_status" < "0"

Search:

Visible	Color	Attribute	Frequency	Rarity	Interval	
▣		frame	29280		184	reset
▣		geninfo	29280		184	reset
□		caplen	29280		184	reset
□		len	29280		184	reset
☑		num	29280		1	reset
☑		protocol	29280		1	reset
☑		timesta...	29280		1	reset
▣		ip	29280		1	reset

Search:

Ignore	Color	Value	Frequency	Rarity
□	🟦	nbss	1695	
□	🟧	smb2	8990	
□	🟩	tcp	11596	
□	🟪	dcerpc	6999	

Fig. 5. Graphical user interface of the implemented system: A) Pixel view showing parts of the network traffic within the scope of selections and projections of interest. Settings with respect to the scan speed and coloring and ordering of attributes are adjusted using the controls in A. B) Lens view for the inspection of message values and alerts in the pixel view. C) Depending on the selected pixel view and attribute, the time view shows the attribute distribution over some specified time range. Settings with respect to axis scaling, coloring of values, and time range are adjusted using controls in C. D) Attribute view enabling the expert to inspect value distributions of attributes and refine the classifier by modifying rarity thresholds or removing predominating values from the histograms. E) Selection view showing an overview of current selections.

When scanning messages sequentially, there is a choice between *message-oriented* versus *time-oriented* scanning. In message-oriented scanning, messages are scanned at a fixed rate causing the pixel visualization to be updated at a constant rate. Since the inter arrival time between messages is not taken into account, the update rate of the time view varies over time. In time-oriented scanning, messages are grouped in fixed time intervals such that the time view is updated at a regular pace. Since multiple messages can adhere to the same time interval, the refresh rate of the pixel views is no longer constant. In case of for instance traffic bursts, message-oriented scanning may be preferred over time-oriented scanning if the data contains samples of malicious activity (e.g., file-scan). If a data burst is not of interest, the expert can switch to time-oriented scanning to analyze these messages at a higher rate.

4.3 Attribute view

The attribute view (Figure 1d) enables the expert to inspect the frequency and rarity of attributes and values that are present within data window ω. Depending on the selected pixel view, only messages that are valid with respect to the pixel view selection settings are visible in the attribute view. The tree structure on the left shows an overview of all attributes in ω by taking the union of all PDML paths in the PDML trees of messages in ω. The expert can adjust the projection settings of the pixel view by changing the visibility of attributes using check boxes. Only attributes that occur in the projection settings are taken into account during classification. When selecting an attribute, the table on the right shows the value distribution of that attribute. The expert can inspect the frequency of values by means of sorting and filtering. Besides frequency, the rarity of a value (see Section 5) is visualized using a bar.

One major problem of anomaly detection is that there is no intrinsic difference between a malicious value and a new incoming value. One can imagine however that the creation of a new file in the network is not necessarily harmful. To keep the number of false alerts in such attributes minimal, the expert can adjust the rarity thresholds of the histograms on a per value basis (or bin basis for numeric attributes). Dominating values can be removed from the histogram by means of a checkbox in front of the value. The rarity threshold that is applicable to a certain value is shown as a vertical bar in the previously mentioned rarity bars. The expert can modify the threshold by either dragging the threshold in the bar visualization or filling an exact value in a popup (Figure 1f). To prevent the expert from having to adjust every threshold manually, the expert is enabled to select multiple values at the same time or to specify a global rarity threshold at the level of an attribute or pixel view. In other words, if there is no threshold set for a value v in (A, v), the threshold for A is used instead. Alerts for specific values and attributes can be ignored during their classification by setting the rarity threshold to 100%. The effect of modifying a threshold is immediately reflected in the brightness of the pixels. The expert can save and load thresholds on a per pixel view basis through import and export functionality.

5 Classification

Due to the large number of attributes per message, it is difficult for the expert to manually spot anomalous values in the traffic. To assist the expert in finding these anomalies, a simple but effective histogram-based classifier is used. Histograms in general are computationally cheap to maintain, easy to understand and can be applied to both numerical and categorical attributes. Their ability to be updated in an incremental fashion makes them both suitable for offline and online analysis. Anyhow, the SNAPS approach is independent of the chosen classifier, and developing better classifiers is a topic for future work.

5.1 Model

Anomalies in general can be classified into the following three categories [3]: point, contextual, and collective anomalies. Network messages are considered point anomalies whenever they are anomalous with respect to the entire data set (e.g., the invocation of a deprecated function call). Messages that are only anomalous in a specific context (e.g., the access of a restricted file by an unauthorized user) are contextual anomalies. Collective anomalies are collections of messages that together are anomalous with respect to the entire data set. Since automatic collective anomaly detection methods are rather error prone for highly heterogeneous and time-dependent data, they are considered outside the scope of this work. Instead, we provide the expert a time view where collective patterns can be visually inspected over different periods in time.

In our online classification approach, network traffic is considered to be relevant within time window ω. Upon the arrival of new data at current time t, messages older than $t-\omega$ are removed from the window and replaced by new ones. For every incoming message, the classifier determines the rarity of values in that message after which the histograms are updated. When training a classifier on a new subset of traffic, the minimum size of the training set $\mathscr{T}$ with respect to that subset is determined using Yamane's sample size formula [9].

5.2 Anomalies

In order to decide whether a network message is a point anomaly, we first have to define when a value in the message is considered to be anomalous. Let $T=(A,v)$ be an attribute value pair in message m. Let $\#A$ denote the number of messages in the data set with attribute A other than **undefined** and let $\#v$ denote the number of messages with (A,v). T is considered to be *rare* if and only if:

$$1-\frac{\#v}{\#A}>\tau \tag{3}$$

where τ represents a rarity threshold defined by the expert. In case where $\#A$ is smaller than Yamane's sample size with respect to $\mathscr{T}$, every value is considered to be rare, since the number of samples in this attribute is too low to build an accurate histogram. **undefined** values are excluded from the histograms, since they predominate the distribution of sparse attributes. To minimize the number of false positive alerts, values for numeric attributes are binned. By default, the bin size s of a numeric attribute is computed by applying Scott's rule [22] on the training set after removing outliers:

$$s=\frac{3.5\sigma}{n^{1/3}}$$

Scott's rule is chosen for its simplicity, since we expect bin sizes to be refined during exploration.

For the detection of contextual anomalies, the expert is enabled to train a new classifier on a selection of interest (see Section 4.1.2). This enables the expert to inspect distributions and look for anomalies on a smaller subset of the traffic.

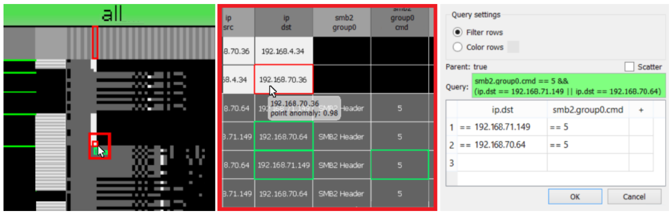
6 Interaction

To enable querying in SNAPS, three operations are supported:

- inspecting values;
- color messages according to rules;
- creation of selections of interest.

For the inspection of values, the notion of a lens is used, showing an enlarged part of the pixel visualization where additional information such as the values and attributes of pixels become visible. Upon the detection of an alert, the expert can stop the message scanning and lock the lens to inspect values in more detail. The rarity score of a value is shown by means of a popup (Figure 6). The expert is enabled to inspect the contents of a pixel in even more detail by switching to the Wireshark interface with one click of a button.

Visual coherence between views is achieved by using color. Hovering the mouse over a value highlights all messages in the pixel view with that value. Similarly, hovering the mouse over a message reveals the location of that message in other pixel views (Figure 1b).



ip src	ip dst	smb2 group0	smb2 group0 cmd
8.70.36	192.168.4.34		
68.4.34	192.168.70.36		
8.70.64	192.168.71.149	SMB2 Header	5
3.71.149	192.168.70.64	SMB2 Header	5
8.70.64	192.168.71.149	SMB2 Header	5
3.71.149	192.168.70.64	SMB2 Header	5

192.168.70.36
point anomaly: 0.98

Query settings

● Filter rows

○ Color rows

Parent: true □ Scatter

Query: smb2.group0.cmd == 5 &&
(ip.dst == 192.168.71.149 || ip.dst == 192.168.70.64)

	ip.dst	smb2.group0.cmd	+
1	== 192.168.71.149	== 5	
2	== 192.168.70.64	== 5	
3			

Fig. 6. Multiselection of values and corresponding query.

Experts can create selections of interest using default, text-based and table-based filtering. To improve interaction speed, SNAPS provides default filtering functionality when the expert selects a pixel, value, or attribute. By means of context menus, the expert can choose to filter the traffic by the presence or absence of the selected value, sending or receiving IP address, or by the conversation in which the message occurred. For more complex queries, a textual interface is provided to assist the expert in creating a query. When writing the query, the expert is instantly notified if the query is syntactically correct. Depending on the part of the query that is being constructed, the expert receives a list of possible attributes, operations, or values that were found in the selected view.

Since the usage of brackets in a query can affect the readability of a query in a negative way, an alternative form of visual querying similar to Excel's advanced criteria filtering [18] is introduced. Queries can be represented as a table where the columns represents attributes and a cell represents a condition (op, reg) where reg represents a regular expression and op the corresponding operation (either $==$, $!=$, $<(=)$, or $>(=)$). Since a message can have at most one value per attribute, the query is constructed by taking the conjunction of all non empty conditions in a row, after which a disjunction is taken over all rows in the table. Figure 7 shows an example of the resulting encoding. We can use the same encoding to enable the selection of multiple pixels into a new filtering condition. To prevent the selection border of two neighboring pixels from occluding each other, selection borders are drawn using a contour algorithm. For a more concrete overview of the interaction, we refer to the supplementary video in Section 4.

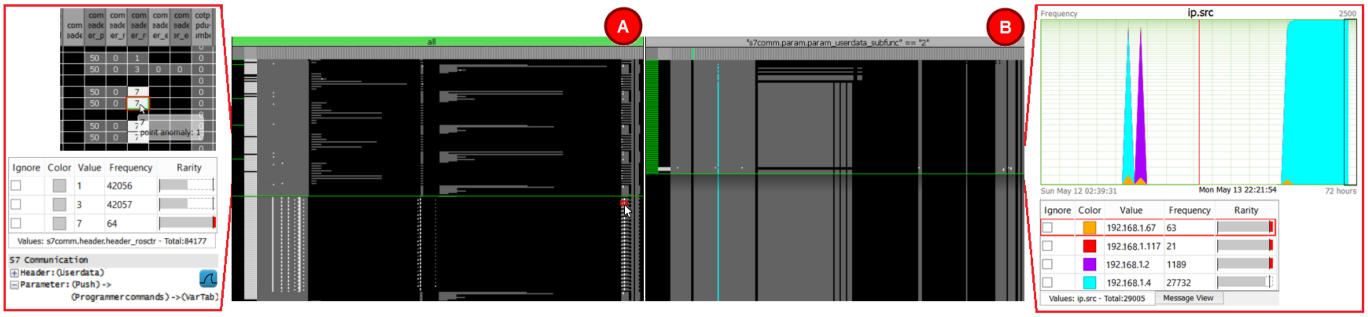
7 Use cases

To illustrate the effectiveness of our approach, we tested the application on two types of data sets. The first data set is obtained by recording one day of internal network traffic from a university. The data consists of approximately 400,000 messages and 500 attributes sent by 25 hosts, initially training the classifier on 30,000 messages corresponding to one hour of traffic. The second data set consists of approximately 500,000 messages, 650 attributes, 10 to 15 hosts representing one week of S7 traffic from a governmental industrial control system. For this data set, the classifier is initially trained on 100,000 messages corresponding to one day of traffic. In the use cases, ω is set to half a day and three days of network traffic respectively.

7.1 University

We initially start the exploration by scanning messages without any selections of interest. The network data contains a wide variety of TCP, DCERPC, and SMB2 traffic (view 1, Figure 5a). Since we are interested to find anomalies at application layer, we create a selection of interest B only containing SMB2 traffic (view 2, Figure 5a). To obtain more reliable classification results for SMB2 traffic, we switch to the local classifier trained on B. Finally, predominating values in the data such as `Ioctrl` request are ignored during classification and the rarity thresholds of attributes like `smb2.file_name` were raised to reduce the number of false positives in the visualization.

After scanning 30 minutes of traffic data, in approximately 5 minutes, we noticed a group of anomalies in B. The lens view showed that the alert was raised by a non-zero value in attribute `smb2.nt_status` (Figure 5b). To receive automatic notification of this alert, a color rule for this condition is applied (indicated by the cyan markers). By creating a new selection of interest C for which `smb2.nt_status` <0 and specifying that the pixels showing the source IP address of such messages should be colored green, we can see that these, what turned out to be SMB2 buffer size warnings, were coming from the same IP address (view 3, Figure 5a). We close C and continue exploration.



Ignore	Color	Value	Frequency	Rarity
□	▨	1	42056	
□	▨	3	42057	
□	▨	7	64	

Values: s7comm.header.header_rosctr - Total:84177

```
S7 Communication
+Header:(Userdata)
-Parameter:(Push)->
        (Programmercommands)->(VarTab)
```

Ignore	Color	Value	Frequency	Rarity
□	🟧	192.168.1.67	63	
□	🟥	192.168.1.117	21	
□	🟪	192.168.1.2	1189	
□	🟦	192.168.1.4	27732	

Values: ip.src - Total:29005

Fig. 7. In view a) the sudden change in S7 traffic raises alerts in attribute `s7.rosctr`. The Wireshark interface shows that alerts are coming from commands to reprogram the PLC. Creating view b) only containing these commands shows that four machines were responsible for sending these commands in the last 72 hours.

Around 10:27 AM, an SMB2 burst was detected as depicted in Figure 1a. Selecting the `smb2.cmd` attribute in the time view of *A* shows an increased number of files being created in the network (Figure 1e). By creating a selection of interest *D* only considering file creations, we obtain more frequent patterns (depicted in Figure 1b). During the burst, a group of alerts was spotted in the `smb2.file_name` attribute. Although it is common for that attribute to generate alerts (e.g., creation of a file), the fact that these alerts were occurring quite fast after one another in *D* was suspicious. The IP address responsible for sending these messages was found by means of coloring. Creating a separate selection of interest for this address with respect to *D* and selecting the `ip.dst` attribute in the attribute viewer, we obtain a list of all locations where these files have been created (Figure 1c). Hovering the mouse over address `192.168.4.34` highlights all locations of this value in the pixel views, showing that most files in the burst period were created at this location (Figure 1f). The values corresponding to the alerts represented a large collection of Microsoft group policy files being accessed in the network (Figure 1d). Since policy files store authorization access at network level, they can only be modified by network administrators. The interesting part, however, is that none of the users in the data set are administrators.

7.2 Industrial control system

In contrast to the office network data, S7 traffic in the governmental control system shows very regular patterns, suggesting that entities in the system send traffic within a particular ordering (Figure 7a). Based on the shape and values that arise from these “vertical histogram” patterns, we can see that the monitoring system `192.168.0.13` reads sensor values from components at a fixed pace. On May 13th 9:00 the pattern becomes disturbed, raising a large collection of alerts in attribute `s7comm.rsotv`. When switching to the Wireshark interface, it becomes clear that these messages correspond to commands to reprogram the PLC. Since we did not expect this behavior, we create a new selection of interest only containing these program commands (Figure 7b). When selecting the `ip.src` attribute in the time view and attribute view, they show that four IP addresses were sending these commands at very specific moments in time over the last 72 hours. Although it is not strange for the main controller to send these commands, the presence of the other IP addresses was unexpected.

8 Discussion and limitations

The SNAPS approach consists of 5 basic steps: 1) create an overview of potential threats at payload level of a message; 2) use the notion of a lens to inspect alerts in more detail; 3) try to gain insight in the alert by inspecting distributions, temporal patterns and co-occurrence of other alerts in the traffic; 4) create selections of interests to either drill down or analyze traffic from different angles; 5) use close cooperation between machine learning and expert to minimize the number of false positive alerts in the visualization.

Rare values are indicated by the SNAPS classifier, and the additional color rules enable experts to define and reuse insights in suspicious behavior. Another plus is that the reuse of existing visualization techniques and tight integration to the trusted environment Wireshark makes the approach relatively easy to learn. Interaction is kept simple and minimal so that the expert can focus entirely on the traffic data. Views for instance are automatically updated when inspecting values through hovering while the wide range of default selections and the use of auto completion enables the expert to create/refine selections with minimal effort. The integration between machine learning and visualization makes the system flexible enough to be configured for different environments.

The approach, however, also has some limitations. First, the scalability in the number of attributes and number of selections is limited to the size of the screen. The more attributes that are of interest, the fewer selections can be shown in parallel. Although we provide the expert functionality to hide and scroll between pixel views, this only solves the problem partly. Second, the number of histograms that have to be maintained in parallel linearly increases with the number of selections of interest. For the cases we studied, we found that up to four selections of interest were sufficient for the expert to answer their questions and understand the complexity of their selections. If the number of selections becomes large, however, updating all histograms in parallel becomes too computationally and memory intensive. Third, one disadvantage of the current data acquisition approach is that the quality of the payload analysis highly depends on the dissector. Since the S7 protocol is classified, the Wireshark dissector for S7 was constructed by means of reverse engineering and therefore produces an abstract attribute space. Although we were able to discover some interesting events, the interpretation of alerts in S7 attributes becomes difficult, even with Wireshark.

Finally, some remarks with respect to the classifier. We used a simple and straightforward classifier, and will consider alternatives in the future. We used an online classifier, which suffers from producing suboptimal classification results in the presence of traffic bursts. Especially when the data window ω is set too small, traffic bursts can predominate the presence of regular traffic. A partial solution would be to use an offline classifier, but this would require to maintain a separate histogram model for the classifier and data window, thereby significantly increasing the complexity of the approach.

9 Conclusions and future work

We presented a novel approach for domain experts to discover anomalies in network traffic by combining deep packet inspection, machine learning and visualization into one coherent system. The ability to create multiple selections in parallel enables the expert to drill down or to focus on specific entities while still maintaining an overview of the state in the network. The time view enables experts to detect patterns and trends over time, while the pixel, attribute and lens viewer together enables the expert to detect outliers. Furthermore, the ability to train and refine classifiers on multiple selections of interest makes the approach flexible enough to be optimized for very specific environments. We have shown the effectiveness of SNAPS on two real-world data sets. Since the approach only relies on the structure of parse data in general, the proposed method is suitable to be applied in other domains as well.

For future work it is interesting to study how we can analyze network traffic at higher levels of abstraction by grouping messages based on context and structure. This would enable the expert to discover more complex collective anomalies such as file scans or replay attacks. Furthermore, there is still an open question about how the speed of the radars affects the detection rate of the expert. Finally, evaluation is necessary to study the effectiveness and scalability of the approach in larger network environments.

Acknowledgments

This work is funded by SpySpot, a project in the Cyber Security program of Netherlands Organisation for Scientific Research (NWO). The authors wish to thank Stef van den Elzen from SynerScope, Erik Boertjes and Sander de Kievit from TNO, colleague Paul van der Corput, and in particular SpySpot members Sandro Etalle, Ömer Yüksel, and Jerry den Hartog for their valuable comments throughout this work.

References

[1] K. Abdullah, C. Lee, G. Conti, J. Copeland, and J. Stasko. IDS Rainstorm: Visualizing IDS Alarms. In *Proceedings of the IEEE Workshops on Visualization for Computer Security*, VIZSEC '05, pages 1–, Washington, DC, USA, 2005. IEEE Computer Society.

[2] K. Bratbergsengen. Hashing methods and relational algebra operations. In *Proceedings of the 10th International Conference on Very Large Data Bases*, pages 323–333. Morgan Kaufmann Publishers Inc., 1984.

[3] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15:1–15:58, 2009.

[4] M. Clayton. Stuxnet malware is weapon out to destroy Irans Bushehr Nuclear Plant. *Christian Science Monitor*, 21, 2010.

[5] G. Combs et al. Wireshark-network protocol analyzer: http://www.wireshark.org/. *Version 0.99*, 5, 2008.

[6] G. Conti and E. Dean. Visual forensic analysis and reverse engineering of binary data. *Black Hat USA*, 2008.

[7] G. Conti, J. Grizzard, M. Ahamad, and H. Owen. Visual exploration of malicious network objects using semantic zoom, interactive encoding and dynamic queries. In *Visualization for Computer Security, 2005.(VizSEC 05). IEEE Workshop on*, pages 83–90. IEEE, 2005.

[8] C. Cowan, C. Pu, D. Maier, J. Walpole, P. Bakke, S. Beattie, A. Grier, P. Wagle, Q. Zhang, and H. Hinton. Stackguard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks. In *Usenix Security*, volume 98, pages 63–78, 1998.

[9] G. D. Israel. *Determining sample size*. University of Florida Cooperative Extension Service, Institute of Food and Agriculture Sciences, EDIS, 1992.

[10] D. Keim et al. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, 2000.

[11] H. Koike and K. Ohno. Snortview: Visualization system of snort logs. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, VizSEC/DMSEC '04, pages 143–147, New York, NY, USA, 2004. ACM.

[12] H. Koike, K. Ohno, and K. Koizumi. Visualizing cyber attacks using IP matrix. In *Visualization for Computer Security, 2005.(VizSEC 05). IEEE Workshop on*, pages 91–98. IEEE, 2005.

[13] Lancope. Stealthwatch therminator (2015). http://www.lancope.com/products/.

[14] J. Landstorfer, I. Herrmann, J. Stange, M. Dork, and R. Wettach. Weaving a carpet from log entries: A network security visualization built with co-creation. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pages 73–82. IEEE, 2014.

[15] F. Mansmann, D. Keim, S. North, B. Rexroad, D. Sheleheda, et al. Visual analysis of network traffic for resource planning, interactive monitoring, and interpretation of security threats. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1105–1112, 2007.

[16] D. Maynor. *Metasploit toolkit for penetration testing, exploit development, and vulnerability research*. Elsevier, 2011.

[17] J. McPherson, K. Ma, P. Krystosk, T. Bartoletti, and M. Christensen. Portvis: a tool for port-based detection of security events. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 73–81. ACM, 2004.

[18] Microsoft. Excel advanced criteria filtering (2015). https://support.office.com/en-ie/article/Filter-by-using-advanced-criteria-4c9222fe-8529-4cd7-a898-3f16abdff32b/.

[19] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath. Malware Images: Visualization and Automatic Classification. In *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, VizSec '11, pages 4:1–4:7, New York, NY, USA, 2011. ACM.

[20] V. Paxson et al. Guide, Bro Quick Start. http://www.bro-ids.org.

[21] B. Rogowitz and A. Goodman. Integrating human-and computer-based approaches to feature extraction and analysis. In *IS&T/SPIE Electronic Imaging*, pages 82910W–82910W. International Society for Optics and Photonics, 2012.

[22] D. Scott. Scott's rule. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):497–502, 2010.

[23] H. Shiravi, A. Shiravi, A. Ghorbani, et al. A survey of visualization systems for network security. *IEEE Transactions on Visualization and Computer Graphics*, 18(8):1313–1329, 2012.

[24] Siemens. Industrial Communnication (2012). https://support.industry.siemens.com/cs/document/78028908?dti=0&lc=en-US.

[25] R. Sloan. Advanced Persistent Threat. *Engineering & Technology Reference*, 1(1), 2014.

[26] M. Stone. In Color Perception, Size Matters. *IEEE Computer Graphics and Applications*, 32(2):8–13, Mar. 2012.

[27] T. Zhang, Q. Liao, and L. Shi. Bridging the gap of network management and anomaly detection through interactive visualization. In *Pacific Visualization Symposium (PacificVis), 2014 IEEE*, pages 253–257. IEEE, 2014.