Semantic Network Traffic Analysis using Deep Packet Inspection and Visual Analytics

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

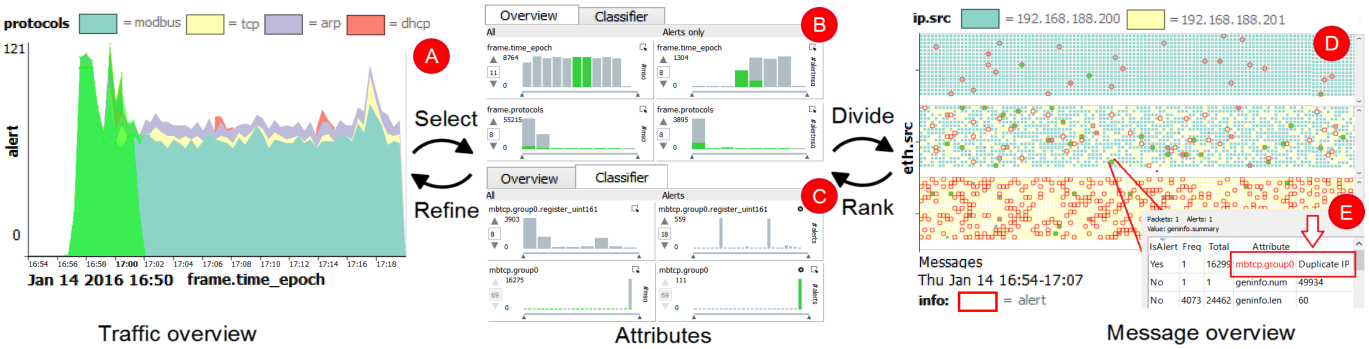

Fig. 1. The discovery of Man-in-the-Middle behavior in network traffic meta-data using deep packet inspection and contextual analysis.

Abstract— For the protection of critical infrastructures against complex virus attacks, automated network traffic analysis and deep packet inspection are unavoidable. Even with the use of network intrusion detection systems, the number of generated alerts is still too large to analyze manually. In addition, the discovery of domain-specific multi stage viruses (e.g., Advanced Persistent Threats) is typically not captured by a single alert. The result is that security experts are overloaded with low-level technical alerts where they must look for evidence that supports the presence of an APT. In this paper we propose an alert-oriented visual analytics approach for the exploration and analysis of network traffic content. In our approach CoNTA (Contextual analysis of Network Traffic Alerts), experts are supported to discover threats in large alert collections through interactive exploration using selections and attributes of interest. Finally, we show the effectiveness of the approach by applying the approach to real world and artificial data sets.

Index Terms—Anomaly detection, Visual analytics, Network traffic analysis, Security visualization, Interactive Data Visualization.

1 Purpose

The aim of network forensics is to discover malicious activity inside logs of network traffic. Especially for critical infrastructures, such as power plants, the presence of malicious activity can lead to the malfunction or even destruction of the underlying system. Forensics can no longer limit their analysis to high-level message properties (e.g., message length, destination address) due to the existence of Advanced Persistent Threats (APTs) [6]. These complex viruses are designed to hide their malicious activity inside the content of messages thereby making them invisible to current flow-based analysis techniques [4]. Since manual inspection of network traffic is impossible due to size and complexity, network forensics use network Intrusion Detection Systems (IDS) [1] to assist them in finding areas of interest. Although these systems automate the analysis of network traffic, the number of (false) alerts is often too large to analyze one by one.

Current methodologies typically visualize alerts by focusing on structural properties such as what, when, or where they occurred in the network. However we believe that an alert as a result of a complex attack does not stand on its own. Instead, we are interested whether the occurrence of an alert is implicitly related to messages or alerts that were sent in the past. For this we need to be able to inspect message collections for the presence of correlations between message attributes (i.e., inter-attribute analysis) and inspect trends in these attributes over time, (i.e., intra-attribute analysis). To enable the simultaneous exploration of message-level phenomena (e.g., field misuse) and traffic-level phenomena (e.g., bursts), our exploration method focuses on a tight interaction scheme between well-established visualization techniques.

In this paper we propose an exploration method that enable experts to gain insight in traffic content by visually exploring and correlating network traffic alerts defined at message-level. In summary, our main contributions are a visual analytics approach to network forensics, enabling experts to:

- explore and analyze network traffic on both attribute and temporal level using alerts as a ground truth, and
- identify correlations between network messages and alerts using selection-based relevance metrics and conversation analysis.

We refer to the work of Cappers et al. [2] for more details.

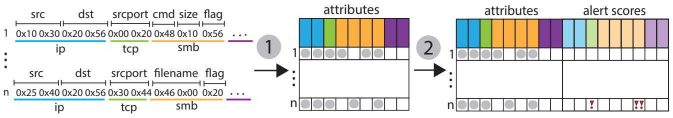

Fig. 2. 1) Serialization of PCAP data with WireShark. 2) Machine learning produces scores per attribute whether these are suspicious or not.

2 Methods

In our system CoNTA we achieve semantic network traffic analysis by enriching raw network packets with protocol semantics using WireShark [3]. The result is a multivariate table where rows correspond to messages and columns correspond to attributes. Depending on the type of message specific attributes are present (Figure 2). Typically, the set of possible attributes is much larger than the set of attributes in a message (order of 100s vs. 10s). For the classification of alerts we use a probabilistic-based IDS as defined by Yüksel et al. [7]. Messages are considered to be malicious if they have at least one alert value.

CoNTA uses four linked views to assist experts in analyzing their alert collection. The time table enables experts to inspect an attribute over time by grouping the traffic over at most two attributes of choice. Grouping the traffic results into a table of small multiples that can be used to for instance inspect traffic over time on a per user or daily basis (Figure 3a). Additionally, an attribute can be selected to act as colormap. Experts can switch between line charts and heatmaps inside the table cells to inspect trends or compare network behavior between one or more groups. For the inspection of individual messages, a pixel map is used where every message is represented as a rectangle (Figure 1d). Malicious messages are indicated by a red box.

The context view (Figure 3b) enables users to store selections of interest for further reuse. Experts can use these contexts to inspect traffic from different perspectives. When creating a new context, a new attribute is added to the data separating the selected messages from the non-selected for further analysis.

For the discovery of correlations between two or more attributes, an attribute viewer is used, showing the frequency of attribute values as interactive widgets. The histograms on the left show the value distribution of an attribute with respect to all messages in the current context, whereas the histograms on the right only considers malicious messages. Alternatively, the histograms can be used to inspect the number of times values are marked as an alert. This enables experts

- *Bram C.M. Cappers and Jarke J. van Wijk are with Eindhoven University of Technology. E-mail: b.c.m.cappers@tue.nl and j.j.v.wijk@tue.nl.*

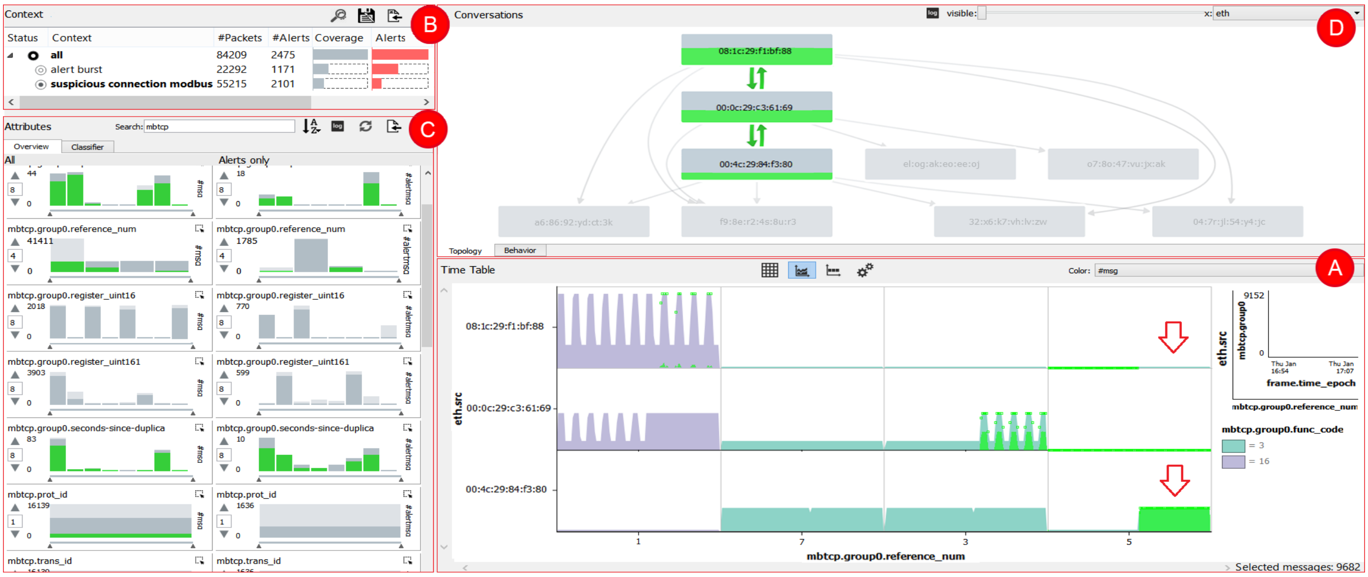

Fig. 3. Graphical user interface of the implemented prototype + components: a) time table, b) context view, c) attribute view, d) conversation view.

to identify bottlenecks in the IDS and filter alerts during investigation. Attributes can be sorted by frequency, number of alerts and relevance [5] to discover potential correlations. The conversation view (Figure 3d) enables experts to inspect sequential patterns inside network conversations for any attribute of choice. In case of IP or MAC addresses, the node-link diagram represents a topology of the network showing the extent to which hosts communicate with each other. Experts can use this graph to filter the traffic on conversations and hosts.

Interaction plays a key role during exploration. To keep brushing and linking consistent and understandable over all views, in CoNTA there was decided to use messages as a central concept. Messages of interest are obtained by selecting and deselecting visual elements (i.e., bars, nodes, edges, or series) across all views. Whenever the selection is modified, every visual element is instantly filled with a green color proportional to the fraction of the selected messages in that item.

3 Results

We tested the effectiveness of our method on one artificial and one real world data set. The first data set represents the simulation of a fully functional artificial water plant consisting of 5 hosts, 80,000 messages and 170 attributes. The data set was designed by an external security company that is specialized in the detection of malicious activity in industrial control systems. To show the practical existence and impact of APTs, they injected an APT attack to damage the facility. The second data set is obtained by recording 3 days of internal office traffic from a university corresponding to approximately 800,000 messages, 400 attributes, and 20 hosts. We describe the use case of the water plant facility in three phases. For a better experience of the interaction and other use cases in practice, we refer to the supplementary video [1].

Discovery: We start exploration by inspecting the number of alerts in the network over time using a line chart. We select the burst period between 16:55 and 17:10 PM in the line chart and save the messages in a new context called "alert burst" (Figure 1a). The conversation view shows that most traffic was created by three nodes: the water tank (...:80), the SCADA system monitoring the plant (...:88), and a router in between (...:69) (Figure 3d). After selecting the new context and sorting the attributes by number of alerts, we see that attribute mbtcp.reg_uint16 has many alerts. Our eye was caught by the attribute mbtcp.group0 whose right histogram shows that there are 50 malicious messages with the value duplicate IP (Figure 1e).

Identification: To verify whether devices are using multiple IP addresses, we group the traffic by MAC-address and switch to the pixel map in the time table. Here we see that most alerts are present in the water tank (Figure 1d). Coloring the messages by IP reveals that the router uses the same IP addresses as the other nodes suggesting man-in-the-middle activity. We select the conversations between the three nodes using the edges in the conversation view, filter the traffic by Modbus, and create a new context for them for further investigation.

Confirmation: Now that we know that this router is suspicious, the next step is to find out what it is aiming for. We select all malicious messages that were sent by the router using the right histogram eth.src in the attribute viewer. Filtering this view by only Modbus fields and sorting the widgets by relevance reveals that most alerts were caused when reading particular registers (Figure 3c). Since each register in the plant stores its own data, we group the traffic per register by setting the time table X-axis to mbtcp.ref_num and table cell Y-axis to mbtcp.reg_uint16. Figure 3a now shows how the register values sent by the water tank are actually perceived by the SCADA system and vice versa. Register 1 shows the status of the tank's valve where the height of the register value describes the extent to which the valve is open. Register 5 represents the overflow flag the tank raises when the water level exceeds a certain threshold. Note that the close valve commands that are sent to the router (Figure 3a, in purple) are not forwarded to the tank. Further note that the tank's overflow flag is suppressed by the router (red arrows, Figure 3a). The result is that the tank overflows while the SCADA system is unaware of this situation.

4 Conclusion

We presented a novel approach for domain experts to explore large message collections using alerts and interaction as a solid basis. The ability to switch from traffic-level overviews to message-level details enable experts to investigate the relationship between high-level traffic phenomena and low-level message fields while staying aware of other concepts such as conversations. The combination of attribute-based widgets and selection-based relevance metrics enable experts to search through large attribute collections. We have shown the effectiveness of the approach on real world and artificial data sets clearly illustrating the complexity network analysts have to deal with.

Acknowledgments

This work is funded by SpySpot, a project in the Cyber Security program of Netherlands Organisation for Scientific Research (NWO).

References

[1] S. Axelsson. Intrusion detection systems: A survey and taxonomy. Technical report, Technical report, 2000.

[2] B. Cappers and J. van Wijk. Understanding the context of network traffic alerts. In *Visualization for Cyber Security (VizSec), 2016 IEEE Symposium on*, pages 1–8. IEEE, 2016.

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

[4] B. Li, J. Springer, G. Bebis, and M. H. M.H. Gunes. A survey of network flow applications. *Journal of Network and Computer Applications*, 36(2):567–581, 2013.

[5] J. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

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

[7] O. Yüksel, J. den Hartog, and S. Etalle. Reading Between the Fields: Practical, Effective Intrusion Detection for Industrial Control Systems. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC '16, pages 2063–2070. ACM, 2016.

[1] https://www.youtube.com/watch?v=yOXDZYKCKZ0