A Tool for Rapid Visual Interrogation & Triage of Alerts

Peter Curtis
pdc46@msstate.edu

Nathan Phillips
ncp38@msstate.edu

Daniel Simpkins
dms226@msstate.edu

T.J. Jankun-Kelly*
tjk@acm.org

Department of Computer Science & Engineering
Bagley College of Engineering
Mississippi State University
Mississippi State, MS 39765

ABSTRACT

We present a tool to assist in the rapid browsing and interrogation of network alert data from monitoring systems like Snort. The tool are designed for time-sensitive applications where further analysis is offloaded after initial identification; thus, rapid exploration and at-a-glance summaries are paramount. Towards these ends, our primary tool uses simple visualization and interaction for identifying what alerts need processing.

Categories and Subject Descriptors

I.3.3 [**Computer Graphics**]: Picture/Image Generation; I.3.8 [**Computer Graphics**]: Applications—*Visualization*; K.6.m [**Management of Computing and Information Systems**]: Miscellaneous—*Security*

Keywords

Security Visualization, Network Alerts, Information Visualization, Web-based Visualization

1. INTRODUCTION

The security of computer networks is a fundamental issue today. The ability to monitor what is occurring on networks and understand when and how attacks are happening is crucial. However, due to the enormity of data generated on networks and the lack of good monitoring methods, this understanding is difficult to achieve. In some settings, understanding is further complicated by time constraints—analysts may have to process large number of alerts in a limited time frame. This leads to security triage where high threat alerts must be identified and processes before low threat alerts. Thus, systems which support such rapid identification of potential anomalies for later, more in-depth processing are needed. This abstract presents such a system.

*Corresponding Author

Poster Presentation
VizSec '14, November 10 2014, Paris, France
.

2. DESIGN & IMPLEMENTATION

Our design focused around three pillars: Simple Representations (complex visualization would require too much time per alert to be operational), In-Depth Interaction (interaction would be used to interrogate data, bridging limitations from the minimal representation), and Rapid Processing (aggregates of alerts can be handled). These are embodied in our alert triage system (Figure 1).

The *alert list* provides the succinct overview of events. In its expanded view (Figure 1, middle) is displays the age of the alert, the involved IPs, the priority, and a brief event summary. All information other than the text summary is graphically encoded for quick visual reads. First, the age of the event uses a progress bar metaphor—the slider moves to the right as newer events are viewed. In addition, age is dual encoded by the shading—a darker fill is used for older events. Source and destination IPs are displayed next using the friendliness classification colormap: Sensitive hosts are color a saturated blue, local hosts a less saturated blue, yellow is used for hosts with no classification, and increasing saturated red are used for suspicious and hostile hosts; these color choices mirror the blue team/red team metaphor often used in security exercises. Finally, the severity or priority of the event uses an orange saturation ramp, with the most saturated orange indicating the most severe or highest priority. All of these elements are interactive, as will be discussed later.

When brushing an individual alert, the *alert description pane* is populated with metadata for interrogation (Figure 1, right). This includes the primary information of the alert list (redundantly color coded), and expanded packet and protocol information. For example, the frequency of occurrence of each IP is included; a high number is a possible indicator of anomalous behavior, a clue for the analyst to follow.

The final visualization component of the main display is the *stream histogram* (Figure 1, lower right). Bins indicate the frequency of events, with a larger bin possibly indicating a sustained attack for further investigation. In addition, a light green rectangle indicates the position of the current alert list within the body of alerts; if there are many alerts, this falls back to a "spike" to indicate where the current view

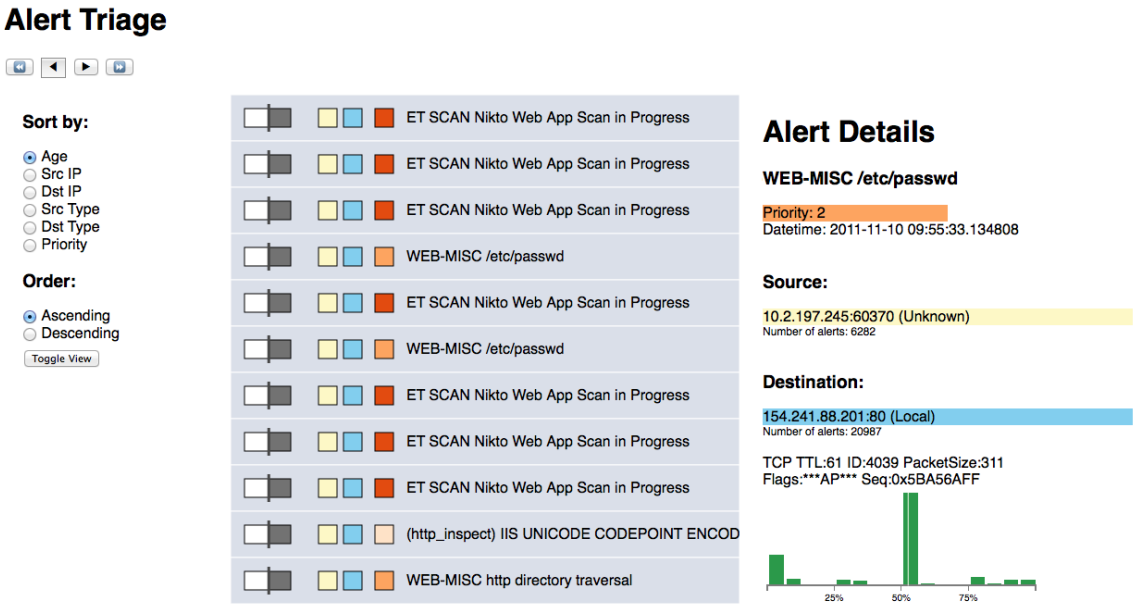

Figure 1: The Alert Triage System. Events information is summarized in the center, with specific alert data and the event histogram to the right, navigation to the left.

is in time.

There are two classes of interaction supported by the triage system—navigation and querying. For navigation, we provide various controls to sort and paginate through the alerts (Figure 1, left). Sorting can be done by time (the default), source/destination IP address, source/destination friendliness classification, or the alert priority/severity. Interaction for the triage tool primarily focuses on interrogating the alert list. First, when an individual IP is brushed, a red border highlights all other occurrences of it within the alerts; the IP address and friendliness classification are also displayed. Highlights can be pinned down for comparison on different pages; the pinned and one other IP can then be highlighted together. This dual highlight allows the discovery of correlated alerts which may be a flag for further processing. Finally, if the analyst has determined that an address has been compromised or malicious, it can be marked as unsafe as part of the triage (Figure 2). Marking updates the IPs friendliness classification system-wide and further exploration of the data maintains the mark. If the analyst determines the mark was in error or the issue has been resolved, the mark can be removed.

This project was implemented using a variety of tools. On the backend, an AMP (Apache-MySQL-PHP) environment was used. On the fronted, d3.js [1] and jQuery were used for the visualizations and interaction; these were written in Coffeescript that was compiled to Javascript. Interaction with the visualization (e.g., navigation, populating the histogram, etc.) is accomplished via dynamic SQL queries via PHP; the entire dataset is never loaded into the browser for maximum performance.

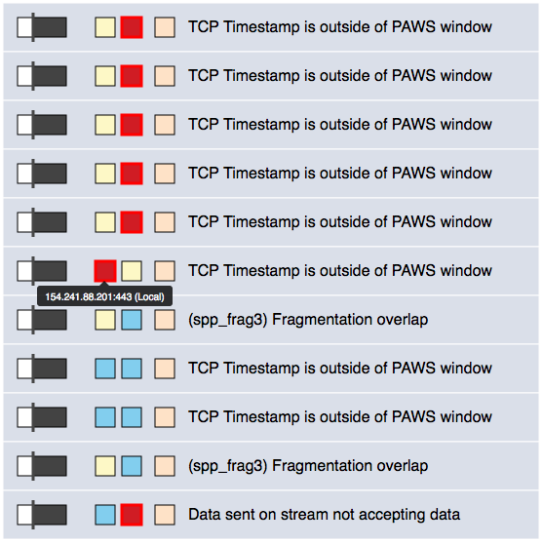

Figure 2: Marking. Compromised and malicious IPs can be marked in red for triage and later processing.

3. ACKNOWLEDGMENTS

This work grew out of a collaboration with the Adelphi Center of the Army Research Laboratory and the Network Attack Characterization and Simulation Testbed (NACMAST) who graciously provided previous funding; the Department of Defense provided additional funding. Yagneshwara Lanka and Chris Lewis worked on earlier prototypes of this project.

4. REFERENCES

[1] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.