Network Access Log Visualization & Sonification

Benjamin deButts
ben@debutts.com
Mentor: Lane Harrison

Abstract

As the worldwide web gets larger and larger, the variety and number of network incidents swells alongside it. Security analysts need to be able to detect, analyze, and respond to incidents faster than ever. However, access logs are both extensive and exhaustive in detail. Visualization, a field that bases itself around making clunky, hard-to-decipher data manageable and understandable, is an obvious method of aiding the detection/response process. By visually displaying access logs on an interactive timeline, analysts will be able orient themselves in a notification stream more quickly and efficiently than with other text-based tools. Additionally, by mapping distinct notification sounds to anomalous incidents, analysts will be able to monitor live network streams without constant visual attention. Upon the recognition of an "incident" tone, they will be able to interact with the timeline, spotting the incident and visually reviewing history to unearth more.

Introduction

Access logs, by their very nature, are narrative stories of what requests have been made on a network. These narratives run constantly, at all hours, and include accesses both harm*less* and harm*ful*. As a tool for security experts, this paper and the tool derivative of it will focus primarily on helping analysts identify harmful traffic.

"Harmful" traffic is a relatively ambiguous term. More often than not, distinguishing "bad" traffic from "good" is the entire game that analysts are playing as they monitor networks. However, there are unique and distinct requests and accesses that can without question be labeled as "attacks" or "incidents". For these select accesses—and more as more types of attacks are uncovered—there will be distinct sounds.

Why implement audio flags in addition to visual? The reasons are fundamentally two: first, and most obviously, audio flags can be passively monitored—much as many people listen to music while working, analysts would be passively aware that a certain sound uttered from the program means a certain access has been made. Two, and less obvious, is that audio data is processed differently than visual. Contrary to visual, humans process audio linearly through time. This means that where a visual schematic could be "read" many different ways by the viewer,

(certain flags being noticed at different times), audio can only be understood one way, and furthermore, in *one* sequence (Skau). Applied to our access logs, this minimizes the delay of visually searching a file into a trivial: heard sound vs. didn't hear sound. Upon hearing the sound, *then* arises the advantage of visualization as it can quickly orient the analyst in the narrative that has been unrolling beneath the non-harmful accesses.

To the Community/Applications

Network security is a bustling field and new attacks occur by the second. Analysts tasked with monitoring these networks have quite an undertaking and would be vastly helped by having the filtering between harmless and possibly harmful traffic done for them. By issuing distinct sounds for distinct incidents, analysts can passively monitor networks without constant visual attention. After hearing an unfamiliar sound, a log-visualization allows the analyst to build a story out of the missed traffic, isolate and observe specific attacks, how they are executed, and from what IP address the attack originated.

Summary

This program aims to expedite and facilitate the network monitoring process by applying visualization and auditory techniques. In this first implementation, visualization serves to place the analyst in the contextual narrative of what requests have been made on the network. Alongside it, the sonification of the program capitalizes on our natural ability to filter out and notice specific sounds out of the many, ultimately allowing passive monitoring of network traffic.

The Supporting Material:

I am planning to implement the above program on Processing in Java. Sound files will be selected for their distinctness and brevity as well as their ability to be processed and played from a computer. Outside of the tones the analyst has selected to be chimed, the program will run silently. For this paper at least, rather than demo with a live-stream, (something that can be implemented in later versions, trivially if I code this right) I plan on utilizing a snippet of a honeypot access log for demonstration of log parsing, detection, and visualization. A video will be attached showing the product's usage and capabilities.

Works Cited

Hadhazy, A. (2014, April 30). Heavenly Sounds: Hearing Astronomical Data Can Lead to Scientific Insights. *Scientific American*.

Humphries, C. (2013). ELVIS: Extensible Log VISualization. *VizSec '13 Proceedings of the Tenth Workshop on Visualization for Cyber Security,* (978-1-4503-2173-0), 9-16. Retrieved October 26, 2014, from http://delivery.acm.org/10.1145/2520000/2517959/p9-humphries.pdf?ip=130.64.151.89&id=2517959&acc=ACTIVE SERVICE&key=AA86BE8B6928DDC7.4579F4D1C4C67060.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=438857745&CFTOKEN=54606407&__acm__=1414176752_7bc53e5491a8eeddf31a8190e2769af0

Phan, D. (2007). Progressive multiples for communication-minded visualization. *GI '07 Proceedings of Graphics Interface 2007,* (978-1-56881-337-0), 225-232. Retrieved October 26, 2014, from http://graphics.stanford.edu/papers/progressive_multiples/progressive.multiples.gi2007.pdf

Skau, D. (2014, October 14). How Audio Can Help Communicate Time Data. Retrieved October 27, 2014.