Digital Investigation 12 (2015) S1–S11

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

DFRWS 2015 Europe

Hviz: HTTP(S) traffic aggregation and visualization for network forensics

David Gugelmann [a, *], Fabian Gasser [a], Bernhard Ager [a], Vincent Lenders [b]

[a] ETH Zurich, Zurich, Switzerland
[b] Armasuisse, Thun, Switzerland

Keywords:
Network forensics
HTTP(S)
Event reconstruction
Aggregation
Visualization
Incident investigation

A B S T R A C T

HTTP and HTTPS traffic recorded at the perimeter of an organization is an exhaustive data source for the forensic investigation of security incidents. However, due to the nested nature of today's Web page structures, it is a huge manual effort to tell apart benign traffic caused by regular user browsing from malicious traffic that relates to malware or insider threats. We present Hviz, an interactive visualization approach to represent the event timeline of HTTP and HTTPS activities of a workstation in a comprehensible manner. Hviz facilitates incident investigation by structuring, aggregating, and correlating HTTP events between workstations in order to reduce the number of events that are exposed to an investigator while preserving the big picture. We have implemented a prototype system and have used it to evaluate its utility using synthetic and real-world HTTP traces from a campus network. Our results show that Hviz is able to significantly reduce the number of user browsing events that need to be exposed to an investigator by distilling the structural properties of HTTP traffic, thus simplifying the examination of malicious activities that arise from malware traffic or insider threats.

Introduction

Network traces are one of the most exhaustive data sources for the forensic investigation of computer security incidents such as online fraud, cyber crime, or data leakage. By observing the network traffic between an internal network and the outside world, an investigator can often reconstruct the entire event chain of computer security breaches, helping to understand the root cause of an incident and to identify the liable parties. In particular, the investigation of HTTP traffic is becoming increasingly important in digital forensics as HTTP has established itself as the main protocol in corporate networks for client-to-server communication (Palo Alto Networks, November 2012). At the same time, malware, botnets and other types of malicious activities nowadays extensively rely on HTTP communication (Dell SecureWorks, March 2014), possibly motivated by the ubiquitous access to the Web even in locations where Internet access is otherwise strictly policed.

Manually analyzing HTTP traffic without supportive tools is a daunting task. Traffic of a single workstation can easily account for millions of packets per day. Even when the individual packets of an HTTP session are reassembled, the traffic may exhibit an abundant number of requests. This high number of requests results from how Web pages are built today. When a browser first loads a Web page from a server, dozens to hundreds of additional HTTP requests are triggered to download further content, such as pictures (Pries et al., 2012; Butkiewicz et al., 2011). These requests may be addressed to the same server as the original page. However, today's common practice of including remote elements, such as advertisements or images hosted on

* Corresponding author.
E-mail address: gugelmann@tik.ee.ethz.ch (D. Gugelmann).

http://dx.doi.org/10.1016/j.diin.2015.01.005

CDNs, results in numerous requests to third-party servers as well. Consequently, finding suspicious activities in a network trace oftentimes resembles the search for a needle in a haystack.

We present Hviz (HTTP(S) traffic visualizer), a traffic analyzer that reconstructs and visualizes the HTTP and HTTPS traffic of individual hosts. Our approach facilitates digital forensics by *structuring, aggregating,* and *correlating* HTTP traffic in order to reduce the number of events that need to be exposed to the investigator.

Hviz reduces the number of HTTP events by combining data aggregation methods based on frequent item set mining (Borgelt, 2012) and domain name based grouping with heuristics to identify main pages in HTTP requests (Ihm and Pai, 2011; Xie et al., 2013). To support the investigator at finding traffic anomalies, Hviz further exploits cross-computer correlations by highlighting traffic patterns that are unique to specific workstations. Hviz visualizes the aggregated events using a JavaScript based application running in the Web browser.

Our main contributions are the following:

- We propose an approach for grouping and aggregating HTTP traffic into abstract events which help understanding the structure and root cause of HTTP requests issued by individual workstations.
- We present Hviz, an interactive visualization tool based on the proposed approach to represent the event timeline of HTTP traffic and explore anomalies based on cross-computer correlation analysis.
- We evaluate the performance of our approach with synthetic and real-world HTTP traces.

As input data, Hviz supports HTTP and HTTPS traffic recorded by a proxy server (Cortesi and Hils, 2014) and HTTP network traces in tcpdump/libpcap format (Tcpdump/Libpcap, 2015). We make Hviz's interactive visualization of sample traces available at http://hviz.gugelmann.com.

In the remainder of this paper, we formulate the problem in Section 2 and introduce our design goals and core concepts in Section 3. In Section 4, we present our aggregation and visualization approach Hviz. We evaluate our approach in Section 5 and discuss evasion strategies and countermeasures in Section 6. We conclude with related work in Section 7 and a summary in Section 8.

Problem statement

When a security administrator receives intrusion reports, virus alerts, or hints about suspicious activities, he may want to investigate the network traffic of the corresponding workstations in order to better understand whether those reports relate to security breaches. With the prevalence of Web traffic in today's organization networks (Palo Alto Networks, November 2012), administrators are often forced to dig into the details of the HTTP protocol in order to assess the trustworthiness of network flows. However, Web pages may exhibit hundreds of embedded objects such as images, videos, or JavaScript code. This results in a large number of individual HTTP requests each time a user visits a new Web page.

As an example, during our tests, we found that users browsing on news sites cause on average more than 110 requests per visited page. Even more problematic than the mere number of requests is the fact that on average a single page visit resulted in requests to more than 20 different domains. These numbers, which are in line with prior work (Pries et al., 2012; Butkiewicz et al., 2011), clearly highlight that manually analyzing and reconstructing Web browsing activity from network traces is a complex task that can be very time-consuming.

Malicious actors can take advantage of this issue: Recent analyses of malware and botnet traffic have shown that the HTTP protocol is often used by such actors to issue command and control (C&C) traffic and exfiltrate stolen data and credentials (Dell SecureWorks, March 2014). Our aim is therefore to support an investigator at investigating the HTTP activity of a workstation when looking for malicious activities, such that.

1. the investigator can quickly understand which Web sites a user has visited and
2. recognize malicious activity. In particular, HTTP activity that is unrelated to user Web browsing such as malware C&C-traffic should stand out in the visualization despite the large amount of requests generated during Web browsing.

Design Goals and Concepts

We start this section by introducing our terminology. Then, we present the underlying design goals and describe the three core concepts behind Hviz.

Terminology

For simplicity we use the term HTTP to refer to both HTTP and HTTPS, unless otherwise specified. We borrow some of our terminology from ReSurf (Xie et al., 2013). In particular, a *user request* is an action taken by a user that triggers one or more HTTP requests, e.g., a click on a hyperlink or entering a URL in the address bar. The first HTTP request caused by a user request is referred to as the *head request*, the remaining requests are *embedded requests*. We refer to a request that is neither a head nor an embedded request as *other request*. These requests are typically generated by automated processes such as update services or malware. The sequence of *head requests* is the *click stream* (Kammenhuber et al., 2006). We organize HTTP requests of a workstation in the *request graph*, a directed graph with HTTP requests as nodes and edges pointing from the Referer node to the request node (see Section 4.1.1 for details).

Design goals

The aim of Hviz is to visualize HTTP activity of a workstation for analysis using input data recorded by a proxy