The final authenticated version is available online at https://doi.org/10.1007/978-3-319-94496-8_10

Idea: Visual Analytics for Web Security

Victor Le Pochat[0000−0003−2297−8328], Tom Van Goethem, and Wouter Joosen

imec-DistriNet, KU Leuven, 3001 Leuven, Belgium
firstname.lastname@cs.kuleuven.be

Abstract. The growing impact of issues in web security has led researchers to conduct large-scale measurements aimed at analyzing and understanding web-related ecosystems. Comprehensive solutions for data collection on a large set of websites have been developed, but analysis practices remain ad hoc, requiring additional efforts and slowing down investigations. A promising approach to data analysis is visual analytics, where interactive visualizations are used to speed up data exploration. However, this approach has not yet been applied to web security, and creating such a solution requires addressing domain-specific challenges. In this paper, we show how visual analytics can help in analyzing the data from web security studies. We present a case study of leveraging an interactive visualization tool to replicate a security study, and evaluate a prototype tool implementing visual analytics techniques designed for web security. We conclude that such a tool would provide a solution that allows researchers to more effectively study web security issues.

1 Introduction

Cyber attacks, data breaches and other forms of cybercrime are increasingly common on the Internet today, making an ever larger impact on our society and economy. To maintain the security of the web in the light of these incidents, the ecosystems of security practices and illicit operations warrant extensive analysis, in order to obtain an overview and gather valuable insights, which ultimately allows for creating better defenses. A variety of large-scale web security observations have been performed for that purpose [1, 2, 5, 19]. However, while comprehensive reusable solutions have been developed for data collection [4, 5], there are no such solutions for the subsequent analysis phase.

Open-source releases of data analysis code from recent web security studies [1, 2] show that current practices for data analysis remain ad hoc and largely underdeveloped. This leads to duplicated efforts, and as analysis tasks may be labor-intensive, they take up time that researchers could use instead to focus on the security issues themselves. However, researchers have no choice but to develop custom solutions, as no comprehensive solution for data analysis specific to web security studies exists in the literature up to date. Creating such a general, reusable and performant framework would allow researchers to gain better insights into their large-scale data and expedite their research, ultimately leading to them being able to investigate and respond to more phenomena at a faster pace.

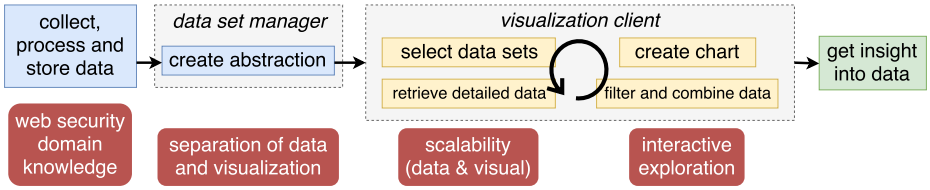

Fig. 1. The pipeline of a visual analytics approach to data analysis for web security, with below each stage the challenge it addresses.

Visual analytics is a promising approach to data analysis [10] which studies the integration of visualization and interaction into this process [16], using the former to leverage the increased data processing power of the human perception [10] and the latter to encourage data exploration. It has already been explored within cyber security in the domains of network security [14] and malware analysis [20]. However, it has not yet been applied for web security, despite its benefits to exploring vast data sets.

While the solutions from the other domains can serve as inspiration, they cannot be directly adopted for web security, as each domain has its own challenges that need to be addressed in visual analytics applications. In prior work [11], we presented an overview of four such challenges, and constructed a design of a visual analytics approach for web security, showcasing techniques that can address these domain-specific challenges. Figure 1 shows the pipeline of this design, alongside the challenges that each step seeks to solve.

In this paper, we explore the application of visual analytics to improve common analysis practices in web security studies. As an example of such an application, we develop a case study of using an interactive visualization tool to replicate a security study. Finally, we perform an initial validation of a prototype that implements our design, to evaluate whether its techniques are beneficial for analyzing the large web security data sets that are collected or publicly available.

2 Motivation

In order to gather correct and comprehensive insights from the large data sets that are collected for web security studies, it is important that the analysis process used can cope with the scale and diversity of that data. We discuss how visual analytics applications would be appropriate for this analysis, taking into account the specific characteristics of web security data.

Nowadays, web security studies routinely measure data for a large section of the Internet: Amann et al. [1] covered 193 million domains in their study of the HTTPS ecosystem, Englehardt and Narayanan [5] mapped online tracking through 90 million requests originating from one million websites, and Durumeric et al. [4] set up Censys for access to regular snapshots of the IPv4 address space. Through visualization, these large amounts of data can be represented within a single view, e.g. using aggregation. The visual representation makes it easier to

discover global patterns and detect outliers, which are often interesting data points from a security perspective. Interactive operations can then allow zooming into the interesting parts of the data to study them in more detail and determine whether they have some special properties. Alternative or domain-specific representations of the data can provide additional insights: displaying server location data on a map may reveal geographical distributions, while plotting the IPv4 address space on a Hilbert curve [9] uncovers patterns in adjacent subnets.

The studies usually entail collecting different kinds of data and searching relations among them and with other data sets. Amann et al. [1] determined the correlation between the application of several security mechanisms related to TLS, while Vissers et al. [19] determined the distribution of sites with cloud-based security across the Alexa top 1 million websites. Multiple data sets can be explored simultaneously by placing their visual representations on a dashboard. By providing interactive combination of data sets, it is not necessary to consider possible correlations upfront: instead, hypotheses based on the patterns and insights found while exploring the data can immediately be tested by linking relevant data sets. Moreover, other data sets, including publicly available ones, could be imported to further augment the data that was collected. Interactions for making selections and synchronizing them across data sets allow for changes made in a certain view to automatically affect the visible data in other charts.

These examples show how visual analytics methods can be used in web security studies to support common analysis tasks, in order to speed up and enhance insight gathering and scale up the breadth of the studies. This helps researchers to have a more complete overview of web security ecosystems.

3 Case study

As a case study of analyzing web security data using an interactive visualization tool, we replicate an experiment conducted by Englehardt and Narayanan as part of their large-scale study of online tracking on Alexa's top 1 million sites [5]. We used a custom web crawler to repeat their measurement of inclusions of third-party resources on those sites in April 2017, collecting 19.6 million inclusions.

The first step toward visually analyzing the data is making the inclusions data set easily accessible in the visualization tool. The data set is transformed to standardized data records and a context (a data type and description) is added, to remove the heterogeneity and ambiguity of data formats and sources. In order to provide interactive control of our crawler, we establish a link between it and the inclusions data set, which enables the dispatching of queries for additional data from within the interactive visual interface.

We can now use the interactive tool to start the crawling process. We load the top one million websites and their rank, whose data is sourced from a publicly available CSV file provided by Alexa[1], into a chart. The distribution of sites is shown in a bar chart, and by zooming into the desired range (through clicking or

[1] `https://s3.amazonaws.com/alexa-static/top-1m.csv.zip`

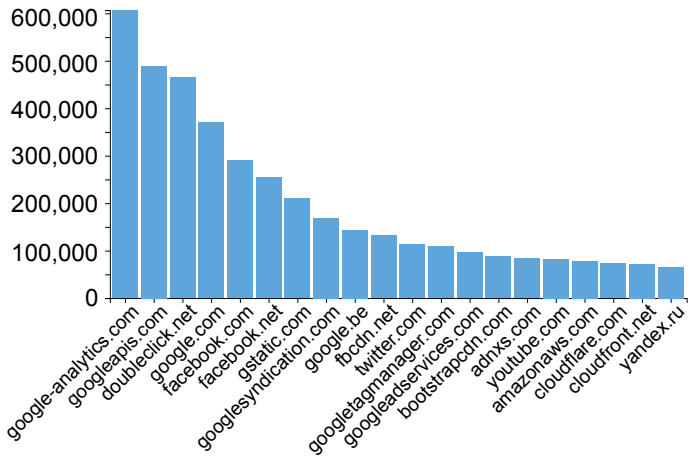

Fig. 2. A visualization of the third parties that are most included in Alexa's top 1 million sites.

dragging) we could select e.g. only the top 100 000 sites. However, we want to collect data for all sites so we do not change the view. In a separate chart, we load the data set for the inclusions per domain. No crawling has occurred yet, so the data set and therefore the chart are empty for now.

Based on the shared `domain` data type, the tool knows that the two data sets are compatible. This, together with the link between the inclusions data and the crawler, allows us to interactively select and queue the one million sites from the Alexa data set, for which the crawler will collect and store the requested data.

We analyze the data with our tool once the crawling operation has completed, but could check on a preliminary distribution while it is ongoing. We load the inclusions data set into a new chart. Based on the `domain` data type, the tool automatically chooses a bar chart that displays the (sorted) aggregate number of sites that include a certain third party. Figure 2 shows the generated chart, replicating the original chart [5, Fig. 2].

By comparing both charts, we can see how tracking practices have changed in the 15 months between the original crawl and ours. In general, inclusions have decreased for the most popular sites. Google domains still serve the most included resources, with Google Analytics as the top domain. The top 10 has not changed much (differences are due to the merged `googleapis.com` and a localized Google domain), but in the next 10 we see more movement, with CDNs pushing out trackers such as BlueKai and MathTag. The visual representation of the data makes it easier to detect these patterns and changes.

We can continue exploring the data to obtain further insights: we can request more detailed data, the domains can be filtered or used as a filter with another chart to study additional properties, and distributions or correlations can be checked through combination with another data set.

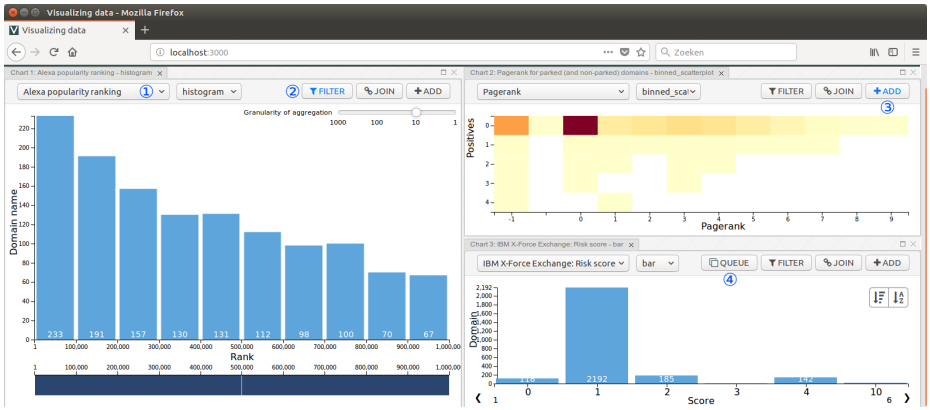

Fig. 3. The interface of the prototype visualization client, with a dashboard allowing to select data sets (1) and explore them simultaneously. The data set on the left is filtered on the items visible in the top right chart (2), where two data sets on the same set of items are combined (3). The bottom right chart shows data that has been interactively obtained using a crawler (4).

For their analyses, Englehardt and Narayanan created the OpenWPM platform [5], designed to simplify and automate data acquisition for web privacy studies. Our approach is complementary, as it provides interactive visual analysis of the obtained data, with both processes being linked in the visualization client. Moreover, the platform's crawls can be interactively launched and managed, which makes replicating studies straightforward (even periodically).

4 Design evaluation

We implemented a prototype visualization tool, shown in Figure 3, based on the design in our prior work [11] addressing the challenges we identified for bringing visual analytics to web security. We perform an initial evaluation of the prototype on three different aspects, which form proxies for evaluating utility and usability. More functionality makes a tool applicable to more use cases. For performance, a more responsive tool does not interrupt the train of thought. For productivity, requiring less effort to visualize data leads to more fluent exploration.

4.1 Functionality

We evaluate our tool's functionality using four criteria obtained from the surveys of open-source and commercial visual analytics systems by Harger and Crossno [8] and Zhang et al. [22] respectively: (1) data source support, (2) visualization and interaction techniques, (3) data analysis methods and (4) system architecture.

We hide the heterogeneity of data sources used in web security through a transformation into standardized records. This allows us to support displaying

individual data sets from any data source. Two data sets of the same source can be combined interactively, however, supporting the composition of multiple data sets across sources is not yet supported. This would require a more complex data retrieval setup since data can no longer be combined at the database level.

The charts we add follow best practices from information visualization [17], in order to ensure correct interpretation of the data without requiring visualization expertise. Web security data comprises multiple data types, and currently our charts can display numerical and geospatial data. Graph and temporal data are currently unsupported, but our modular approach to charts simplifies extending the tool with appropriate visualizations. As for interaction, we support filtering and zooming to study data both as an overview and in depth [15], as well as linking and brushing [21] to enable synchronization of selections across data sets.

We have not yet added any interactive data analysis, such as statistical measures or data mining algorithms. These analyses would be interactively applied in the client but executed on the server, as the calculations need to be performed before aggregating the data.

We implement our tool using the client-server model, which places the burden of retrieval and processing of the raw large-scale data on the server. This reduces the processing power needed on the client and allows it to be web-based and therefore accessible across devices and platforms.

4.2 Performance

We focus our performance evaluation on how well the process scales with data sets of increasing size, as web security studies often yield large amounts of data. To achieve better scalability, we integrate default aggregation into our design, and we only request non-aggregated records upon explicit selection. We evaluate two performance aspects: the time needed to answer a data request, as this affects the responsiveness of our tool and therefore the exploration process [12], and the size of the resulting data, which affects the processing speed and transfer time. We test on data sets of 0.1, 1 or 10 million randomly generated items with attributes of either 100, 1 000 or 10 000 possible values.

For both the aggregated and non-aggregated approach, the time needed to retrieve the whole data set scales linearly with the size of the data set. However, the request for aggregated data is answered around ten times faster, leading to better responsiveness for larger data. Regarding the size of the response, aggregated data scales with the number of bins, but non-aggregated data scales with the size of the data set. For our test set, the latter yields a document that is at least six orders of magnitude larger.

4.3 Productivity

While visual interfaces are known to speed up analysis of cyber security data [7], analysts may avoid the process of creating visualizations due to it being difficult and labor-intensive [6]. We reduce this effort through automation of two phases: setting up the transformation of data sets to standardized records and selecting

appropriate charts based on the data type. We evaluate the complexity of our visualization tool by repeating analyses using the original data from a study by Vissers et al. [18] on the parked domains ecosystem.

Data sets are transformed by executing code that describes data access and parsing. This code can be custom developed, which for a transformation to aggregated data requires 29 logical lines using the methodology of Nguyen et al. [13]; in total there are 28 such transformations. Automatically generating this code requires less configuration: for an SQL database 8 parameters are sufficient.

To quantify the effort of visualizing and exploring data, we estimate the number of actions and time needed using the Keystroke-Level Model [3]. Creating a new chart takes 4.0 seconds for four operations. Applying an operation to a chart (e.g. combining two data sets) takes 6.6 seconds for six operations. Combining these tasks into an analysis where two data sets are loaded, a selection is made in one chart and that selection is then applied to the other chart, takes 17 seconds.

4.4 Discussion and future work

Our evaluation shows that several design elements have a positive impact on the three evaluated aspects and therefore on utility and usability: abstracting over data sources expands functionality, aggregation improves responsiveness and automation reduces the visualization effort. Opportunities for further development lie in additional data processing and analysis functionality as well as further simplification of the visualization process.

In order to formally evaluate the utility and usability of our tool, we plan to validate it through a user study with web security researchers and analysts. This validation will allow us to more conclusively determine if our visual analytics approach is an adequate solution for enhancing their analysis workflow.

5 Conclusion

Through an overview of common analyses in web security studies and the development of a case study, we demonstrate how visual analytics can be advantageous for analyzing and extracting insights from the vast amounts of web security data generated and publicly available. However, domain-specific challenges need to be addressed in order to develop a useful and usable solution. Through an initial evaluation of a prototype tool, we show that techniques such as data abstraction, aggregation and automated visualization effectively tackle these challenges to enhance the exploration and interpretation of large web security data sets.

In the future, we plan to make our visualization tool available to the wider communities of researchers and analysts, as a platform for stimulating collaboration through shared data sets and analyses. In combination with easier (periodic) replication of previous studies, this opens up even more possibilities to analyze ecosystems and test hypotheses using the wealth of available data.

Acknowledgments This research is partially funded by the Research Fund KU Leuven.

References

1. Amann, J., Gasser, O., Scheitle, Q., Brent, L., Carle, G., Holz, R.: Mission accomplished?: HTTPS security after DigiNotar. In: Proc. IMC. pp. 325–340 (2017)
2. Cangialosi, F., Chung, T., Choffnes, D., Levin, D., Maggs, B.M., Mislove, A., Wilson, C.: Measurement and analysis of private key sharing in the HTTPS ecosystem. In: Proc. CCS. pp. 628–640 (2016)
3. Card, S.K., Moran, T.P., Newell, A.: The psychology of human-computer interaction. Lawrence Erlbaum Associates (1983)
4. Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., Halderman, J.A.: A search engine backed by internet-wide scanning. In: Proc. CCS. pp. 542–553 (2015)
5. Englehardt, S., Narayanan, A.: Online tracking: A 1-million-site measurement and analysis. In: Proc. CCS. pp. 1388–1401 (2016)
6. Fink, G.A., North, C.L., Endert, A., Rose, S.: Visualizing cyber security: Usable workspaces. In: Proc. VizSec. pp. 45–56 (2009)
7. Goodall, J.R.: Visualization is better! A comparative evaluation. In: Proc. VizSec. pp. 57–68. IEEE (2009)
8. Harger, J.R., Crossno, P.J.: Comparison of open-source visual analytics toolkits. In: Proc. VDA. SPIE (2012)
9. Irwin, B., Pilkington, N.: High level Internet scale traffic visualization using Hilbert curve mapping. In: Proc. VizSec. pp. 147–158. Springer (2008)
10. Keim, D.A.: Visual exploration of large data sets. Commun. ACM 44(8), 38–44 (2001)
11. Le Pochat, V., Van Goethem, T., Joosen, W.: Towards visual analytics for web security data. In: Proc. PAM (Posters) (2018), extended abstract. Available from `https://lirias.kuleuven.be/handle/123456789/618030`
12. Liu, Z., Heer, J.: The effects of interactive latency on exploratory visual analysis. IEEE Trans. Vis. Comput. Graphics 20(12), 2122–2131 (2014)
13. Nguyen, V., Deeds-Rubin, S., Tan, T., Boehm, B.: A SLOC counting standard. In: Proc. COCOMO. USC CSSE (2007), `http://csse.usc.edu/TECHRPTS/2007/usc-csse-2007-737/usc-csse-2007-737.pdf`
14. Shiravi, H., Shiravi, A., Ghorbani, A.A.: A survey of visualization systems for network security. IEEE Trans. Vis. Comput. Graphics 18(8), 1313–1329 (2012)
15. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proc. VL. pp. 336–343 (1996)
16. Thomas, J.J., Cook, K.A. (eds.): Illuminating the Path: The Research and Development Agenda for Visual Analytics. IEEE Computer Society Press (2005)
17. Tufte, E.R.: The visual display of quantitative information. Graphics Press (1983)
18. Vissers, T., Joosen, W., Nikiforakis, N.: Parking sensors: Analyzing and detecting parked domains. In: Proc. NDSS. Internet Society (2015)
19. Vissers, T., Van Goethem, T., Joosen, W., Nikiforakis, N.: Maneuvering around clouds: Bypassing cloud-based security providers. In: Proc. CCS. pp. 1530–1541 (2015)
20. Wagner, M., Fischer, F., Luh, R., Haberson, A., Rind, A., Keim, D.A., Aigner, W.: A survey of visualization systems for malware analysis. In: Proc. EuroVis - STARs. pp. 105–125. Eurographics Assoc. (2015)
21. Ward, M.O.: Linking and brushing. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, pp. 1623–1626. Springer (2009)
22. Zhang, L., Stoffel, A., Behrisch, M., Mittelstadt, S., Schreck, T., Pompl, R., Weber, S., Last, H., Keim, D.A.: Visual analytics for the big data era – a comparative review of state-of-the-art commercial systems. In: Proc. VAST. pp. 173–182 (2012)