CVExplorer: Multidimensional Visualization for Common Vulnerabilities and Exposures

Vung Pham
Dept. of Computer Science
Texas Tech University
Lubbock, TX, USA
vung.pham@ttu.edu

Tommy Dang
Dept. of Computer Science
Texas Tech University
Lubbock, TX, USA
tommy.dang@ttu.edu

***Abstract*—Cyber attacks cause great damage to our national security, ranging from individual internet user to biggest governmental/industrial organizations, such as Equifax (Data Breach 145.5 Million Accounts, reported in July 2017) or Uber (Data Breach 57 Million Records, reported in November 2017). The cyber assault has significantly increased in breadth and depth. This paper introduces *CVExplorer*, a novel interactive system for visualizing cybersecurity threats reported in the National Vulnerability Database. The proposed system aims to work as a reporting and alerting tool that can help enhance the security against cyber attacks can potentially reduce network vulnerabilities. The *CVExplorer* system containing multiple linked views allows users to visualize the relationships of various dimensions in the large number of vulnerability reports, such as types and levels of vulnerability, vendors, and products. The *CVExplorer* provides an intuitive interface and supports a range of interactive features, such as filtering and ordering by vulnerability severity ratings, allowing users to narrow down topics of interest quickly. To demonstrate the effectiveness of the proposed system, we demonstrate the *CVExplorer* on two case studies of Common Vulnerabilities and Exposures retrieved from the National Vulnerability Database.**

***Keywords*-Common Vulnerabilities and Exposures; Common Vulnerability Scoring System; High-Dimensional Visual Analytics; Parallel coordinates; Force directed layouts.**

I. Introduction

With virtually all computing networks and data storage under constant bombardment of cyber attacks and cyber espionage activities, the battlefield of national defense is no longer restricted to military facilities or security agencies. Computing network facilities and data storages in national, industry, academic research labs and offices are all possible targets of cyber attacks. Besides, the popularity of social networking sites and applications can quickly spread the vulnerability from sites seemingly irrelevant to national defense to locations within federal defense facilities. Thus, a network vulnerability analysis, remedy, and alerting tool that can help enhance the security against human-error-utilizing cyber attacks can potentially reduce network vulnerabilities. Even though human error is the most significant cybersecurity vulnerability (such as falling for phishing, unrestrained web browsing, and lousy password habits), the state-of-the-art vulnerability scanners are not designed to detect vulnerabilities introduced by humans interacting with the system [1]. The proposed system aims to fill in this gap.

In particular, we expand the features exposed by vulnerability scanners such as Nessus [2] and Shodan [3] (a kind of "dark" Google) and present vulnerability assessments to users via interactive visual interface instead of dealing with tediously technical outputs. Our analytics system means to provide better understandings of cybersecurity threats and will enable it to provide timely recommendations regarding potential risks via well-documented daily reports from the National Vulnerability Database (NVD), a widely used database containing millions of records about specific device vulnerabilities. The proposed method is implemented in JavaScript embedded in the standard web browsers and potentially extended as a browser plugin for alerting possible cybersecurity threats when users access a site/domain.

Our contributions in this paper thus are:

- We propose a new approach to analyze prominent features in Common Vulnerabilities and Exposures entries through coordinated multiple views. In contrast to existing techniques which mostly look into one dimension at a time, we inspect the relationships of these dimensions for interesting correlations.
- We develop an interactive prototype, named *CVExplorer*, which adopted the standard as well as customized visual representations to explore these relationships in big data. The *CVExplorer* supports a range of interactive features allowing users to narrow down events of interest quickly.
- We demonstrate the *CVExplorer* on two case studies of Common Vulnerabilities and Exposures in 2017 and of an Autonomous System Number.

The paper is structured as follows: We describe related work in the following section. Then we discuss the design motivations and considerations of *CVExplorer* in Section III. We introduce our *CVExplorer* interface and its components in Section IV. We illustrate the use of *CVExplorer* on two case studies in Section V and discuss its limitations and scalability for big data. Finally, we conclude the paper.

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

II. Related Work

In this section, rather than attempting to survey all cybersecurity visualization, we instead highlight the most related work. Current approaches to vulnerability assessments can be roughly classified into passive and active vulnerability assessments. Passive vulnerability assessment techniques aim to cross-reference system specific characteristics with databases of known vulnerabilities [4], such as the National Vulnerability Database. Techniques belong to this category include p0f [5], PRADS [6], and ShoVAT [7]. In contrast to passive vulnerability assessment, active vulnerability assessment techniques actively probe devices to identify vulnerabilities, including port scanning, checking for SQL injections and HTML injections, monitoring network traffic, and dropping malicious or exploitative payloads [8]. While passive vulnerability assessment supports historical vulnerability assessments on vulnerabilities throughout a services lifetime [7], active vulnerability assessment only provide a snapshot in time of known vulnerabilities [2]. An exemplary scanner of active vulnerability assessment tools is the Nessus Network Security Scanner that lists the various vulnerabilities present in the remote host.

In recent research, Watson et al. [9] proposed a visualization for vulnerability scan data by network zone using free and open-source tools. The proposed visualization uses a mean of all Nexpose severity scores for a given node to determine its overall severity score. However, this visualization technique is limited since it is attempting to capture large data in a simplified, visual representation. Scalability is another issue with a one to one mapping between devices and nodes for large networks with thousands of devices.

Shiravi et al. [10] have also identified that most three dimensional systems [11], [12] are harder for a security analyst to perceive and interact with (compared to conventional 2D systems) due to occlusions which require a substantial amount of interactions (such as rotating and zooming) from an already overworked security analyst. Consequently, our proposed system focuses on 2D standard and modified visualization techniques to tackle the design requirements for analyzing a large number of Common Vulnerabilities and Exposures entries described in the next section.

III. Design motivations and Decisions

In this section, we present the design considerations of multidimensional *CVExplorer* Visualization for Common Vulnerabilities and Exposures (CVE). We first start with some background knowledge on the Common Vulnerability Scoring System (CVSS) defined by the NVD.

A. Vulnerability Metrics

CVSS is a free and open industry standard for assessing the severity of computer system security vulnerabilities. CVSS attempts to assign severity scores to vulnerabilities, allowing responders to prioritize responses and resources according to the threat. NVD provides CVSS scores for almost all known vulnerabilities. The NVD supports both CVSS v2.0 and v3.0 standards which contains *base scores* (which represent the innate characteristics of each vulnerability), *temporal scores* (metrics that change over time due to events external to the vulnerability), and *environmental scores* (scores customized to reflect the impact of the vulnerability on an organization). We focus on the latest CVSS (v3.0) and use color-encodings in Table I consistently in this paper.

Severity	None	Low	Medium	High	Critical
Base Score	0.0	0.1-3.9	4.0-6.9	7.0-8.9	9.0-10.0

Table I
NVD Vulnerability Severity Ratings.

Common Vulnerabilities and Exposures (CVE) is a catalog of known security threats. The catalog is sponsored by the United States Department of Homeland Security (DHS), and threats are divided into two categories: vulnerabilities and exposures. The CVE entries available in NVD include a variety of fields [13] such as a the vulnerability scores (described above), the vulnerability types (such as *CWE-400: Uncontrolled Resource Consumption* or *CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer*), a list of vulnerable vendors (*Microsoft*, *Oracle*) and products (*Windows*, *apple_tv*, or *android*), external references to advisories, CVE published date, CVE last modified date, and entry descriptions. These are the important variables for our *CVExplorer* visualization.

B. Design motivations

Shodan has been acknowledged as one of the first search engines designed to crawl the Internet and to index discovered services and to provide advanced vulnerability assessment capabilities [7]. To free a cyber analyst from the tediously long vulnerability Shodan outputs, we propose *CVExplorer* visualization for analyzing a set of obtained CVEs. While other visualization approaches [9], [14] for analyzing individual CVE feature at a time are publicly available, our *CVExplorer* inspects the dynamic correlations between these dimensions to answer the following research questions:

- **R1:** Are there any relations within and between vendors, products, and vulnerability types at a given time point/interval?
- **R2:** For a given vendor, what are the targeted products/software and what are their levels of vulnerability change over time?
- **R3:** What are the popular vulnerability types and how did they evolve?
- **R4:** What are the popular topics associated with the different level of vulnerability severity over time?

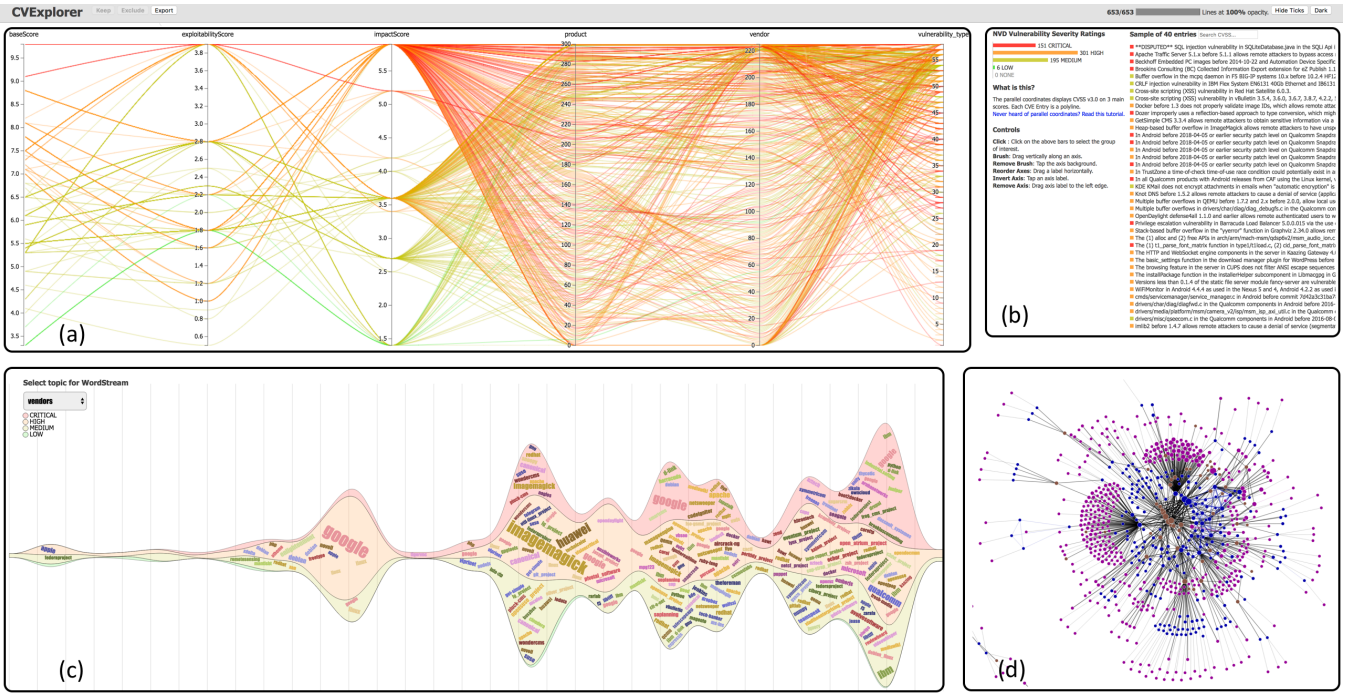

Figure 1. *CVExplorer* visualization for 653 CVEs: (a) Parallel coordinates of important features in CVEs (b) Summary of NVD Vulnerability Severity Ratings (c) *Timed Wordle* of popular vulnerable vendor over time (d) Network of vendors (in blue), products (in purple), and vulnerability types (in brown).

C. Design Decisions

Parallel coordinates are a standard way of visualizing high-dimensional geometry and analyzing multivariate data [15], [16]. Therefore parallel coordinates are adapted to display the correlations of prominent dimensions within the data. As force-directed layout uses repulsive forces between nodes and attractive forces between adjacent nodes and therefore it is handy to highlight network structures (such as clusters or outliers). We use force-directed layouts as the primary way to group related entities (vendors, products, and vulnerability types) and minimize link crossings which are the main limitation of parallel coordinates.

While very effective for visualizing network structures, the lack of temporal information is the main drawback of the force-directed graph. Therefore, we propose a hybrid visualization of *Streamgraph* [17] and *Wordle* to maximize the space usage for displaying the evolution of important topics (vendors, products, and vulnerability types) and hence communicate global criticality trends [18]. Within this graph, the time axis is aligned horizontally from left to right. This design is widely adopted when visualizing time series data [19]. The main drawback of this hybrid visualization: It is tricky to follow the progression of individual entities. In other words, visually identifying different instances of the same label is challenging due to changes in text orientation produced by the *Wordle* algorithm. We alleviate this problem via user interaction: a local stream of the interested entity will be highlighted on demand.

IV. *CVExplorer* Visualization

A. Parallel Coordinates

We adopted parallel coordinates depicted in Figure 1(a) to present the relationships of the following prominent dimensions in the CVE data: vulnerability scores (basic, temporal, and environmental scores), vendors, products, and vulnerability types. The last three dimensions/coordinates are ordered by how popular they are in the input data. Filters on each dimension can be applied by simply dragging on the axes. *CVExplorer* also supports other interactions such as reordering axes by *drag and drop*, dropping a dimension, or revert its scale. The right panel in Figure 1(b) also supports group or individual CVE selection.

B. Network

The force-directed layout depicted in Figure 1(d) is an efficient tool to reduce edge-crossings (a drawback of parallel coordinates) as the related entities can freely move closer to each other to form clusters. In this context, the entities (vendors, products, and problem types) are considered as related if they appear in the same CVE and therefore connected by a link. The link thickness indicates how often they are reported in the same CVEs. Node sizes are calculated based on the number of reported CVEs. We use a different color scheme to encode network nodes to differentiate them from the vulnerability severity color range: blue for vendors, purple for products, and brown for vulnerability types.

C. Timed Wordle

The *Timed Wordle* depicted in Figure 1(c) is implemented using the combined *Wordle* and *Streamgraph* algorithms. *Wordle* main strength is the ability to give quick emphasis on important terms using relatively larger font sizes. It is also known for its algorithm to optimize its space by organizing words efficiently. However, *Wordle* lacks the ability to show the evolution of the topic over time. On the other hand, *Streamgraph* is a popular method for visualizing topic evolution. Its strength is the ability to provide a comprehensive overview of the evolution of the underlying topics over time. However, *Streamgraph* is limited in terms of the number of layers and space for each layer. Hence, its layer normally does not contain or contains only a few terms of the topic that the layer is representing, making it difficult to trace and compare the evolutions of different terms in the topic across time. Our implemented *Timed Wordle* presents a strategy to visualize the evolution of topics (i.e., qualitative severity rankings of the CVEs) and their terms (i.e., vendors, problem types, products, and descriptions) by using *Streamgraph* to represent the topic overview across time and using *Wordle* to compactly and elegantly place terms to its corresponding layer and time step and to visually emphasize important terms with relatively larger font-sizes.

V. Use Cases

A. CVE Data

The NVD is the vulnerability management database managed by U.S government in order to support systematic and automatic reporting and managing of vulnerabilities. As of the time of this writing, NVD contains 110,766 CVEs with different qualitative severity rankings as *low* (6,181 CVEs), *medium* (52,218 CVEs), *high* (47,182 CVEs), and *critical* (5,185 CVEs). Figure 2 shows the overview of the severity distribution of these 110,766 CVEs over time. We can easily notice the emergence of *critical* type in 2014 and the gain of its popularity as *critical* rating has been introduced recently together with CVSS v3.0.

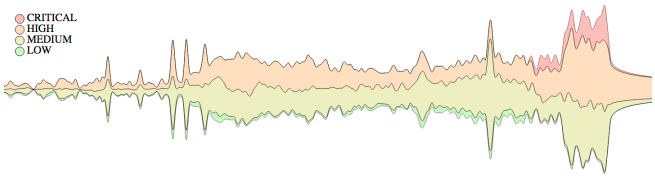

Figure 2. Overview of 110,766 CVEs reported from 1998 to 2018 obtained from NVD. Layers are severity classification: *low*, *medium*, *high*, and *critical*.

This section demonstrates our approach on two sample CVE subsets to show how this visualization solution could quickly highlight the overview of security problems with different qualitative severity rankings over time and also assists in an in-depth investigation if needed. The first

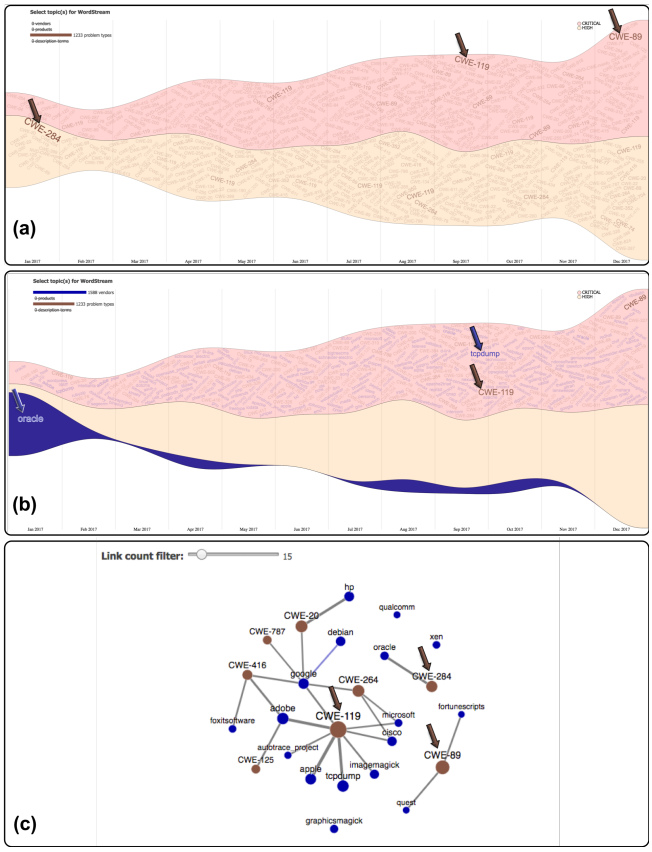

Figure 3. *CVExplorer* visualization for the NVD CVE-2017 dataset for *high* and *critical* qualitative severity rankings: (a) *CWE-284*, *CWE-119*, and *CWE-89* as the dominant security problem types in 2017; (b) *tcpdump* and *CWE-119* are the dominant vendor and problem type in September 2017 and Oracle has *seasonal* reporting schedule; (c) The network view shows the relationships among the main problem types and vendors.

one is to analyze security alerts in 2017 from the NVD Data feeds. The second case is the analysis of the security issues of an ASN (Autonomous System Number) with CVEs dataset collected from *Shohan* [3], [7]. While the first use-case communicates global criticality trends, the second demonstrates *CVExplorer* application to a specific ASN.

B. Use case 1: CVE-2017

The *CVE-2017* dataset collected from NVD CVE feeds contains 14,746 CVEs with qualitative severity rankings distribution: *low* (205 CVEs), *medium* (5,119 CVEs), *high* (7,509 CVEs), and *critical* (1,913 CVEs). This use-case shows how our interactive visual analytics system could help to quickly identify critical security alerts of their affected vendors, products, and problem types. Besides, users could also investigate the related references to examine these CVEs further as well as finding patch updates for these security alerts.

We focus on analyzing the CVEs at *critical* and *high* qualitative rankings by filtering on the severity axis of parallel

coordinates. Figure 3 displays several interesting views of the filtered data. At a glance to the *Timed Wordle* in the top panel (Figure 3(a)), we can easily identify *CWE-284* (Vulnerability in the Oracle Service Fulfillment Manager component of Oracle E-Business Suite), *CWE-119* (improper restriction of operations within the bounds of a memory buffer), and *CWE-89* (Movable Type plugin A-Member and A-Reserve vulnerable to SQL injection) as the dominant problems in 2017. The middle panel shows our investigation of an affected company (*Oracle*). It reveals the *seasonal* vulnerability reporting pattern from *Oracle*. After further reviewing the related references in the CVE *json viewer*, we found that these were *Oracle* scheduled patch updates. In particular, Oracle's critical patch updates schedules were on the Tuesday closest to the 17th day of January, April, July, and October.

Figure 3(c) displays the high correlations of among popular vendors and problem types by filtering via the provided slider: vendors and problem types are collocated in at least 15 *high* and *critical* CVEs. As depicted, *CWE-119* vulnerability is highly connected to *Google*, *Adobe*, *Apple*, *Microsoft*, and *Cisco* among the other vendors. In particular by inspecting 96 *critical* CVEs reported in September 2017 in the *json* viewer, we can confirm that they all belong to the *CWE-119* vulnerability type and mostly affect Apple's products.

C. Use case 2: CVEs from an ASN

The first use case shows high-level vulnerability trends (which is more suitable for someone in charge of producing executive reporting for an organization), this section provides another use case for system administrator or cyber analyst to investigate the security of an ASN over time. One sample ASN was chosen, and five pages of CVEs (100 *banners* per page) were collected from *Shodan* [3] for the ASN and 441 vulnerabilities found. Of which 439 were CVEs and the other two were of type MS17-010 (Microsoft Security Bulletin Number). The CVE qualitative severity ranking distribution was *high* (157 CVEs), *critical* (56 CVEs), *medium* (187 CVEs), and *low* (39 CVEs).

The parallel coordinates Figure 4(a) show the correlations among the prominent dimensions within the data. Notice that *critical* (red) CVEs have lower *Exploitability* metrics which reflect the ease and technical means by which the vulnerability can be exploited. The *Timed Wordle* in the middle panel highlights dominant problem type, vendor, and product in 2010: *CWE-119*, *microsoft*, and *iis* versus the recent correspondings: *CWE-20* (`mod_auth_digest` does not properly initialize or reset the value placeholder in authorization headers leading to information disclosure or denial of service), *apache*, and *http_server*. The network view in Figure 4(a) further confirms these correlations.

This use case shows that it is relatively easy to use our visualization system to identify critical security alerts and related references for patch updates. As of July 2018, we

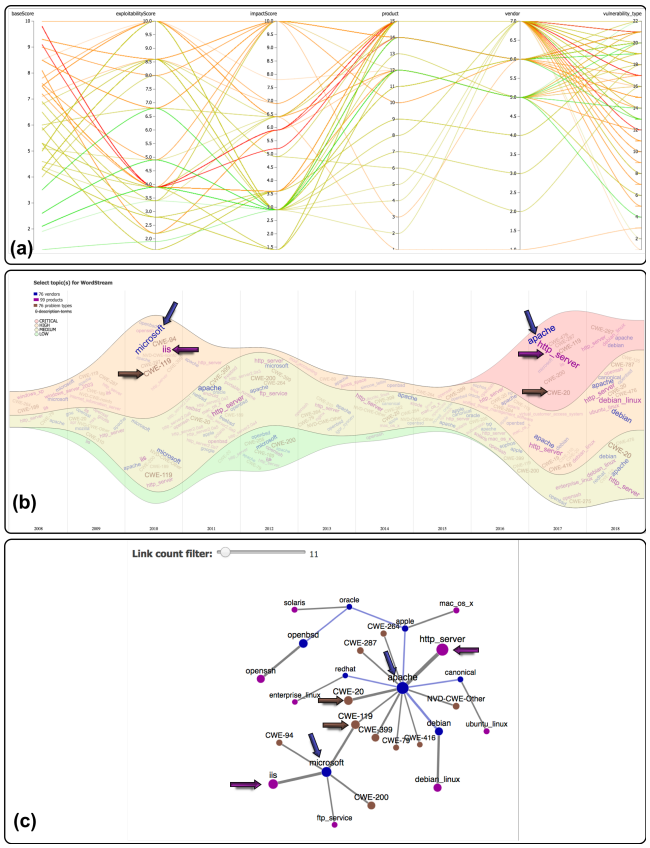

Figure 4. Visualizing features of CVEs from an ASN: (a) Parallel coordinates of prominent dimensions, (b) *Timed Wordle* highlights the recent critical problem type, vendor, and product (at the right arrows) versus those in 2010, (c) Network view confirms the stronger connections among problem types, vendors, and products in (b) via thicker links.

also selected several IPs from this ASN and scanned them with Shodan and discovered that many of these critical CVEs are still existing in the services from the specified ASN and this "security-sensitive" corporation seemed ignorant about these critical vulnerabilities.

D. Discussions and limitations

This section discusses the scalability of the proposed framework for big data: Can the prototype handle more than 110,766 CVEs (as of Angust 2018)?

The force-directed graph has known limitations ("hairball" issues), and one might argue that this representation is not the most effective way to show all relationships between vendors, products, and vulnerabilities. Nevertheless, it would be hard to argue that the networks in Figure 3(c) and Figure 4(c) have successfully captured the correlations of a focused set of entities (using the slider to filter the strength of these connections). While the parallel coordinates provide an overview of correlations between sequential dimensions, the force-directed graph displays a focused view: the relationships of popular vendors, products, and security

types (answering the research question **R1**).

Conventional parallel coordinates do not scale for big data (visual clutter and overplotting issues) since each CVE profile is rendered as a polyline. We tackle the scalability issue of parallel coordinates by adopting the following approaches: (1) Asynchronous rendering method using *d3.timer* [20] which is an efficient queue for managing a large number of concurrent renders for the polylines. (2) Edge-bundling method using density-based clustering for each dimension: this approach allows rendering the clustered lines using polygons, decreasing rendering time remarkably.

VI. Conclusion and Future Work

The vulnerability of our cyber systems constitutes a critical threat to national security. This paper proposes an interactive visual analytics system for analyzing vulnerability reports from the NVD that can help enhance the protection against human-error-utilizing cyber attacks. The system has three linked components: Parallel coordinates, forced directed network, and *Timed Wordle*. While parallel coordinates are a standard technique for visualizing high-dimensional data using polylines, the force-directed layouts provide a way to highlight related entities by positioning them near to each other. Network entities are brought closer to each other (forming clusters) by forces applied to nodes and connections between nodes. Finally, the *Timed Wordle* provides a supplement view on the evolution of vendors, products, as well as types and levels of vulnerability.

CVExplorer is implemented in D3.js [20]. The online application, source code, supplementary materials, more use cases, and a demo video are provided via our GitHub project repository, at https://idatavisualizationlab.github.io/CVSS/.

References

[1] M. Evans, L. A. Maglaras, Y. He, and H. Janicke, "Human behaviour as an aspect of cyber security assurance," *CoRR*, vol. abs/1601.03921, 2016. [Online]. Available: http://arxiv.org/abs/1601.03921

[2] L. Harrison, R. Spahn, M. Iannacone, E. Downing, and J. R. Goodall, "Nv: Nessus vulnerability visualization for the web," in *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*, ser. VizSec '12. New York, NY, USA: ACM, 2012, pp. 25–32. [Online]. Available: http://doi.acm.org/10.1145/2379690.2379694

[3] J. Matterly, "Shodan: The world's first search engine for internet-connected devices," 2014, http://www.shodanhq.com[Accessed date: June 5, 2018].

[4] S. Samtani, S. Yu, H. Zhu, M. Patton, and H. Chen, "Identifying scada vulnerabilities using passive and active vulnerability assessment techniques," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, Sept 2016, pp. 25–30.

[5] M. Zalewski, "p0f v3: Passive fingerprinter," 2012, http://lcamtuf.coredump.cx/p0f3/[Accessed date: July, 2018].

[6] E. Fjellskal, "Passive real-time asset detection system," 2009, http://gamelinux.github.io/prads/[Accessed date: July, 2018].

[7] B. Genge and C. Enăchescu, "Shovat: Shodan-based vulnerability assessment tool for internet-facing services," *Sec. and Commun. Netw.*, vol. 9, no. 15, pp. 2696–2714, Oct. 2016. [Online]. Available: https://doi.org/10.1002/sec.1262

[8] D. Kennedy, J. O'Gorman, D. Kearns, and M. Aharoni, *Metasploit: The Penetration Tester's Guide*, ser. No Starch Press Series. No Starch Press, 2011. [Online]. Available: https://books.google.com/books?id=TWKLBAAAQBAJ

[9] S. Watson and H. R. Lipford, "A proposed visualization for vulnerability scan data," in *SOUPS*, 2017.

[10] H. Shiravi, A. Shiravi, and A. A. Ghorbani, "A survey of visualization systems for network security," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 8, pp. 1313–1329, Aug. 2012. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2011.144

[11] T. Takada and H. Koike, "Tudumi: information visualization system for monitoring and auditing computer logs," pp. 570–576, 02 2002.

[12] A. Yelizarov and D. Gamayunov, "Visualization of complex attacks and state of attacked network," in *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*. IEEE, 2009, pp. 1–9.

[13] B. A. Cheikes, D. Waltermire, and K. Scarfone, "Common platform enumeration: Naming specification version 2.3," *NIST Interagency Report 7695, NIST-IR*, vol. 7695, 2011.

[14] S. Özkan, "Cve details: The ultimate security vulnerability datasource," 2011, https://www.cvedetails.com/index.php[Accessed date: July 10, 2018].

[15] X. Zhao and A. Kaufman, "Structure revealing techniques based on parallel coordinates plot," *Vis. Comput.*, vol. 28, no. 6-8, pp. 541–551, Jun. 2012. [Online]. Available: http://dx.doi.org/10.1007/s00371-012-0713-0

[16] A. Dasgupta and R. Kosara, "Pargnostics: Screen-space metrics for parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, pp. 1017–2626, 2010.

[17] L. Byron and M. Wattenberg, "Stacked graphs – Geometry & aesthetics," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1245–1252, 2008.

[18] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong, "TextFlow: Towards better understanding of evolving topics in text," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2412–2421, 2011.

[19] T. N. Dang, A. Anand, and L. Wilkinson, "TimeSeer: Scagnostics for high-dimensional time series," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 3, pp. 470–483, March 2013.

[20] M. Bostock, V. Ogievetsky, and J. Heer, "D3 data-driven documents," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, 2011.