Automatic Narrative Summarization for Visualizing Cyber Security Logs and Incident Reports

Robert Gove

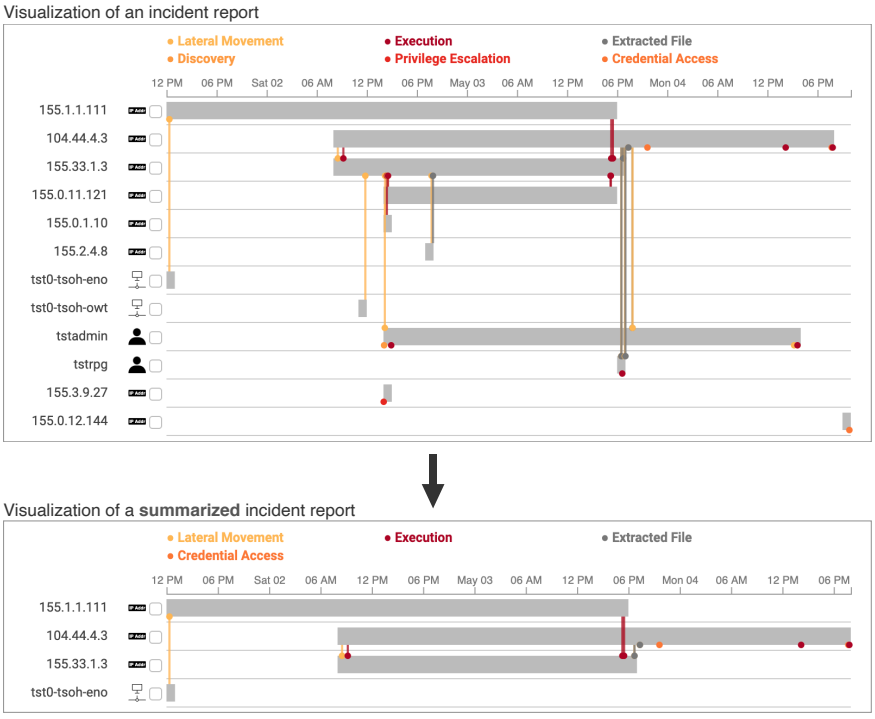

Fig. 1. Visualization of an incident report (top) and a summary of the incident report (bottom). (Screenshots do not show real data.) Incident reports that include dozens of entities and hundreds of relationships benefit from this compact visualization because a table shows one row for each relationship or event. By using a summary of the incident report, this visualization can be even more compact and focus analyst attention on the core sequence of events and the relationships between the main victims and attackers.

Abstract—Cyber security logs and incident reports describe a narrative, but in practice analysts view the data in tables where it can be difficult to follow the narrative. Narrative visualizations are useful, but common examples use a summarized narrative instead of the full story's narrative; it is unclear how to automatically generate these summaries. This paper presents (1) a narrative summarization algorithm to reduce the size and complexity of cyber security narratives with a user-customizable summarization level, and (2) a narrative visualization tailored for incident reports and network logs. An evaluation on real incident reports shows that the summarization algorithm reduces false positives and improves average precision by 41% while reducing average incident report size up to 79%. Together, the visualization and summarization algorithm generate compact representations of cyber narratives that earned praise from a SOC analyst. We further demonstrate that the summarization algorithm can apply to other types of dynamic graphs by automatically generating a summary of the Les Misérables character interaction graph. We find that the list of main characters in the automatically generated summary has substantial agreement with human-generated summaries. A version of this paper, data, and code is freely available at `https://osf.io/ekzbp/`.

Index Terms—Summarization, incident reports, dynamic graphs, cyber security

1 Introduction

Analysts often document and share information about cyber security incidents in the form of *incident reports*, which contain a narrative of the incident. Some cyber security incidents are relatively easy to understand because they are small in scope. However, in our work we see incident reports describing week-long events that contain several dozen entities and hundreds of relationships, which can make it difficult for analysts to quickly identify the key entities and events. Additionally, we are seeing the rise of systems to detect sets of malicious activity [4], moving beyond simple alerts toward automatically generated incident reports. This motivates the need to be able to generate easily consumable summaries of long or complicated logs, incident reports, and related types of cyber narratives.

If the incident reports were purely in natural language then we could use text summarization techniques [10] to reduce their size, thereby focusing analyst attention on key entities and relationships.

• *Robert Gove is with Two Six Technologies. E-mail: robert.gove@twosixtech.com.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

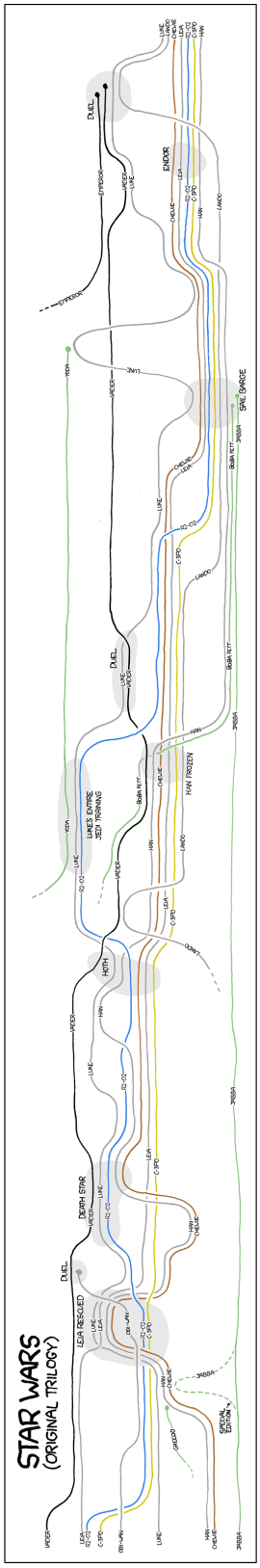

Fig. 2. A storyline visualization by Randall Munroe showing a narrative summary of the Star Wars original trilogy movies. In Munroe's words, "The horizontal axis is time. The vertical grouping of the lines indicates which characters are together at a given time." [23] This data is a summary of the movie trilogy narrative instead of the full narrative with all characters (see the discussion in Sect. 2).

However, incident reports often comprise structured data or extracts from system logs that are not amenable to traditional automatic text summarization techniques. This motivates the need to summarize these data to highlight the primary entities, events, and relationships. A narrative summary could help orient analysts to the relationships between the main victims and attackers and then quickly triage incident reports and remediate incidents, or a narrative summary could provide an easily consumable overview to brief organization leaders or the public.

Visualizations, such as storylines [23] or Gantt charts, can show relational data more compactly than tables. Time-oriented graphical visualizations similar to these could be effective for analyzing the temporal nature and inter-connectedness of log data and incident reports, which typically describe some sort of relational data. However, for large or complex data, analysts may prefer to begin analysis by visualizing an overview or summary of the data, which has proven successful in various contexts [27]. Indeed, we collaborate with a security operations center (SOC) analyst who told us a summary visualization is preferable over scrolling through a long table of data. However, even though prominent examples of storylines show a summarized narrative instead of the full set of characters and interactions (see Fig. 2), these summaries appear to be generated manually instead of automatically.

This paper introduces a novel algorithm to summarize cyber security logs and incident reports for the purpose of generating concise visualizations. The summary algorithm shares similarities with extractive text summarization by extracting the core sequence of events, the primary entities, and the relationships that connect them. Users can customize the amount of summarization to near-arbitrary levels. An evaluation on real incident reports finds that the summaries reduce false positives and improve average precision by 41% while reducing the average incident report size up to 79%. An accompanying visualization tool inspired by Gantt charts displays the resulting summaries in a more compact manner than tables and earned praise from a SOC analyst colleague. To demonstrate broader applicability on dynamic graphs, we use the summarization algorithm to automatically summarize the Les Misérables character interaction graph. We find substantial agreement between the characters in this automatic summary and the characters in human-generated summaries.

This paper's contributions are:

1. A new algorithm for summarizing temporal relational data, and an evaluation of the algorithm's performance.

2. A visualization tool designed to visualize cyber security logs and incident reports for SOC analysts.

2 Background and Related Work

Narrative, as defined by the Oxford English Dictionary, is "An account of a series of events, facts, etc., given in order and with the establishing of connections between them." We note differences between narrative and the concept of dramatic structure, which is often described in many Western cultures as having a beginning, middle, and end, for example by Aristotle in his work *Poetics*. Whereas narrative is specifically concerned with the series of events and connections between them, dramatic structure relates to plot, theme, dialogue, music, conflict, and other elements [11]. From this perspective, narrative accurately describes the communication goals of incident reports, whereas dramatic structure is not a core part of reporting and remediating a cyber security incident.

There is a body of visualization research that focuses on how humans tell a story that is augmented with data and visualization, such as news reports that include (interactive) data visualizations. This style of story visualization involves an authoring and editing process, and is not merely a presentation of narrative (i.e. a series of facts given in order, as defined above). Segel and Heer describe the design space of visual storytelling in news media [26]. Their design space contains three divisions of features, which they call genre, visual narrative, and narrative structure. They further describe three structures of these news media stories: Martini glass, interactive slideshow, and drill-down story. Chen *et al.* [6] proposed a framework for bridging visual analytics

and storytelling, and the researchers proposed recommendations for designing tools to synthesize stories from data analysis. Hullman and Diakopoulos [14] discuss how a visualization author's design choices can influence the audience's interpretation of narrative visualizations.

Automatic text simplification and automatic text summarization are two related processes within natural language processing to reduce the size of text. The goal of simplification is to reduce linguistic complexity and produce simpler prose [28], whereas the goal of summarization is to reduce the size of a document but retain the important information [10]. There are two general approaches to summarization: extraction and abstraction [10]. Extractive summarization extracts content from the original document and joins the content to create a summary. In contrast, abstractive summarization creates a summary by generating novel sentences about the document. The narrative summarization algorithm proposed in this paper extracts the important entities from the data and retains their relevant relationships, which makes the algorithm closer in concept to extractive summarization than abstractive summarization.

Within visualization, several areas have developed a variety of simplification and summarization techniques. One example of extractive summarization is the creation of proxy graphs, which are smaller versions of large graphs derived through sampling, filtering, or structural skeletons [25]. Communities can be used to generate summaries that might be analogous to abstractive summarization [38]. Motif simplification uses aggregation to replace common patterns of nodes and links in node-link graph visualizations with glyphs, resulting in a more concise visualization [8]. Event sequence simplification produces simplified event sequence visualizations by filtering, merging, and replacing events to create an aggregated display [21]. These types of simplifications and summaries are useful for identifying high level patterns, but it is unclear how they can be applied to summarize a narrative represented by relational data, or how they can account for temporal occurrences and relationships.

DataShot is a tool that extracts interesting facts from tabular data and automatically generates an infographic from the facts [36]. These facts can provide helpful summary statistics, but the infographics are not designed to present narrative. More specifically relating to incident reports, SumRe is visualization tool designed to summarize drug safety incident reports [15]. It was designed for the purpose of comparing incident reports about individual patients and finding similar incident reports to collect evidence of a potential safety issue. This differs from our use case, where our incident reports have many events about and relationships between several different entities instead of a single entity. Furthermore, SOC analysts are generally not comparing incident reports in order to identify similarities, but instead are trying to determine the level of severity and steps for remediation.

Gantt charts are a type of project schedule visualization developed by Henry Gantt [22]. Gantt charts show a project schedule and the duration of its tasks, or events, and the interdependencies between them. The project schedule itself is graphical, where the vertices are tasks and dependencies are edges. (Strictly speaking, this dependency graph must form a directed acyclic graph, which is used by the Gantt chart to topologically sort the graph to determine the positioning of tasks in the visualization. This means Gantt charts must be modified to support undirected or cyclical graphs.) Project schedules define additional data such as the start date and duration of each task. A task in a schedule is said to be *critical* if a delay in its finish date causes the project's finish date to also be delayed because of interdependencies between tasks. The *critical path* is the longest sequence of tasks in the schedule comprising only critical tasks. Studying the critical path helps reduce project risk by identifying tasks that are most likely to delay the project. In some sense the critical path is therefore an extractive summary of the schedule's most important tasks, but this type of summary relies on the notion of delaying a project and so it does not immediately apply to summarizing narratives.

Storylines are a type of visualization for analyzing entity timelines and relationships between entities. The design of storylines are based on movie narrative charts created by Randall Munroe [23]. Fig. 2 shows an example visualization illustrating the narrative of the Star Wars original trilogy movies. These visualizations were likely created by hand. Various researchers have extended the work on these types of visualizations, for example by developing and improving automatic layout algorithms [13, 17, 19, 29, 32, 35], developing authoring tools [33, 34], and extending the capabilities and performance of storylines [1, 24]. Many of the example narratives from the cited works show summaries of the original narratives instead of the full set of characters and their interactions. For example, the visualization of the Star Wars narrative (Fig. 2) excludes many entities (characters, ships, planets, etc.) found in the movies: Grand Moff Tarkin, Uncle Owen, Aunt Beru, Mon Mothma, the second Death Star (but not the first), and the planet Tatooine (but not Hoth or Endor), among others. These editorial decisions were likely an effort to help simplify the narrative and make it more feasible to display the narrative of all three movies together. However, none of the prior work addresses how to generate a summary of these temporal entity relationship graphs in order to create summarized storyline visualizations.

3 Representing Logs and Incident Reports as Graphs

Incident reports are a type of document that describe potentially harmful events that could affect the confidentiality, integrity, or availability of a system. Incident reports often include semi-structured or structured data from system or network logs that document what was affected, when it was affected, and where and how the incident originated. Because incident reports contain these tables of data, we need to transform the data into a format suitable for input into a visualization and summarization tool.

Structured Threat Information Expression (STIX) [2] is a standardized specification for communicating information about cyber security threats. When stored in Extensible Markup Language (XML) or JavaScript Object Notation (JSON), STIX is a widely supported machine-readable format for storing, transmitting, and reading cyber security threat information. Because many tools and vendors support importing and exporting data using the STIX specification, the work in this paper is engineered to read JSON-formatted STIX data using version 2.1 of the STIX specification.

For visualization and summarization purposes, the data must include:

- Entities, which are the subjects and objects of narratives, such as IP addresses, hosts, and user accounts.
- Relationships that describe associations between entities, such as a user logging into a host. Relationships are typically directed, where the source is acting on the target. The relationship type can be used to describe the relationship, for example to indicate a tactic, technique, or procedure (TTP) from the MITRE ATT&CK Framework [31]. The framework is a matrix with 14 columns (one column for each tactic). The columns are ordered so that tactics that occur in later stages of an attack are farther to the right, thus giving an order of severity.
- Timestamps are applied to entities and relationships to denote when entities and relationships were observed in the data.

This information forms a temporal entity relationship graph, a type of dynamic or temporal graph. The entities are the vertices and the relationships are the edges. The timestamps and relationship types provide additional data for visualization.

4 Incident Report Visualization

We designed and developed an incident report visualization tool to give SOC analysts a succinct view of the information contained in an incident report. The system is built to read incident reports stored in v2.1 of the STIX specification in the JSON format. Users can choose an incident report to load from a database, or users can copy the data and paste it into the visualization tool.

4.1 Design Goals

We designed the visualization with several design goals in mind, which we believed important to analyzing incident reports based on our experience working with SOC analysts:

Succinctness. Incident reports often contain relevant rows from network log database tables, e.g. DNS, Zeek, or Kerberos logs. For incident reports describing complex or long-running behavior, these tables can span several pages; they provide great detail, but are not succinct. Our goal is to represent this data more succinctly.

Consistency. Provide a consistent representation of the incident, regardless what log type the data came from or which person or automated tool generated the incident report.

Activity progression. Show the progression of activity in the incident report to clearly identify the time periods each entity was involved in the incident report.

Patterns. Show patterns that may be difficult to identify when the data is displayed in a table, such as connected components of entities related to each other. Other cyber security visualization tools have shown the importance of this [12].

Learnability. Use familiar visual designs to help users learn it.

4.2 Visualization Design

When a user selects an incident report, the system processes the incident report as a graph, where entities are vertices and relationships are edges as described above. Next the system identifies all connected components, identifies all timestamps associated with each entity, and checks which (if any) relationship types map to the MITRE ATT&CK framework.

At the top of the visualization is a timescale (Fig. 3A) that indicates the first and last observed timestamps in the incident report.

Each entity is assigned a row in the visualization (Fig. 3B). The row includes the entity name (such as the value of the IP address), an icon representing the type of entity, a checkbox to select the entity, and a horizontal gray bar (Fig. 3C) that indicates the duration of time when activity was recorded for that entity. This design was chosen to meet the goal of *familiarity*, so that the visual design would be similar to Gantt charts and bar charts.

Entities are ordered first by connected component, which are ordered by the earliest timestamp for any entity in the connected component. Within connected components entities are ordered by entity type, where IP addresses occur first, then hosts, then user accounts, then all other entities. Within entity type, entities are ordered so that earlier entities with longer durations are at the top. This type of consistent vertical ordering aids understanding [1].

If there is a relationship that connects two entities, the system draws a vertical line between the two entity rows (Fig. 3D). The system places the vertical line to align with the timescale for the timestamp when the relationship was logged. If the relationship represents a TTP from the MITRE ATT&CK framework then it is colored using a sequential color scale from yellow to orange to dark red [5] based on how advanced the phase is in the attack lifecycle (see the relationships in Fig. 1); otherwise the color is a dark gray. A legend at the top of the visualization (Fig. 3E) maps colors to relationship types. This design was chosen to be *familiar* to users who use Gantt charts, while also aiding *learnability* by invoking the Gestalt principle of connectedness to indicate relationships. Showing the relationships with the entity durations is meant to address the *activity progression* goal while also illustrating any temporal or relational *patterns*.

This visual design can provide a more *succinct* visual representation of a story than a table because each entity is a row, whereas typically a table has one row for each relationship or timestamp. This means that incident reports with fewer entities than relationships or timestamps will be longer than this incident report visualization.

Finally, because all incident reports are visualized this way, it provides a *consistency* to all incident reports, regardless of who or what generated them.

4.3 Changes Due to User Feedback

We showed the visualization to our SOC analyst several times during the development process, each time gathering feedback and changing the visualization accordingly.

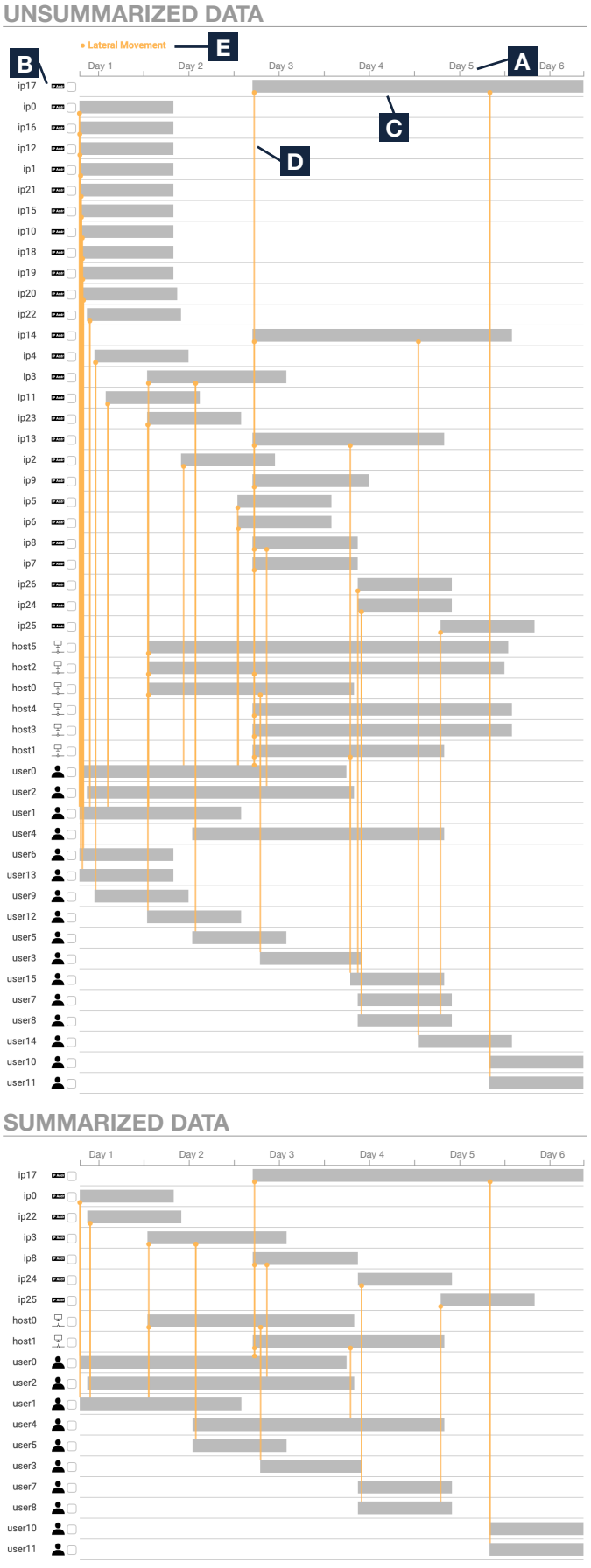

Fig. 3. An anonymized incident report of a red team event. The top visualization shows the unsummarized report, and the bottom shows a summary generated with threshold $t = 0.6$.

Initially, the system vertically ordered entities using a type of topological sort, so there was a relational and temporal order to the entities. This had the benefit of reducing the number of connections that cross entities. However, when getting feedback from a SOC analyst he said that he preferred to see the IPs at the top, and then the hosts. This is because he is typically more familiar with the network aspect of the data (i.e. IPs and hosts) and the user account aspect.

An earlier version of the visualization used a categorical color scale for the relationship colors because relationship types appear to be nominal, but the SOC analyst suggested using a sequential color scale because some TTPs are more concerning than others. This led us to implement the sequential scale described above.

4.4 Alternative Designs

Fundamentally, the algorithm's goal is to summarize the narrative in dynamic, or temporal, graph data. The summary can then be visualized in a variety of ways, using whatever method is suitable for the user and their tasks. In other domains this visualization method could use techniques like animation or small multiple to communicate the temporal elements of the data. However, we pursued a different method that was believed to be more suitable for our users and their tasks. In particular, in our experience in this domain it is uncommon for SOC analysts to use interactive visualizations due to the limited time analysts have to analyze incident reports, and information about incident reports is often shared via ticketing systems and presentations. Therefore we pursued a visualization design where each entity is easily identifiable by name and type, the precise times of entities and relationships is prominent, and the visualization does not rely on interactions or animations in order to reduce the learning curve and make it easy to share the visualization in a variety of media.

Therefore we primarily considered two alternative types of visualizations to illustrate incident reports:

Storylines could show entity interactions over time, but they probably would not work well with the large number of relationships we see in incident reports. This would likely cause large numbers of line crossings, and therefore a visualization with reduced readability.

Gantt charts are designed to show directed acyclic graphs, but the data in network logs and incident reports is not necessarily directed nor acyclic. As discussed in Sect. 2, this makes it impossible to use the standard Gantt chart sorting algorithm to order entities. Additionally, Gantt charts have at most one relationship between a pair of tasks, which always occurs between the end of one task and the beginning of the other. But with incident report data these relationships can occur at any point in the duration of an entity, and not necessarily at the beginning or end of its duration.

5 Summarizing Incident Report Narrative Graphs

Our data analysis goal is to summarize the narrative in order to reveal the most important entities and sequences of events. Explicitly, the goal is not to identify malicious activity within an incident report. By the vary nature of an incident report, the activity is already believed to be malicious or otherwise important. Instead our goal is to summarize the narrative in the incident report to help situate analysts and to provide an overview of the events and the entities involved. In our experience, cyber security analysts often have little or no time to explore data, so we believe it can be helpful to provide a summary of the incident report narrative to help avoid spending unnecessary time analyzing incident reports that may not turn out to be important. We discussed this with our SOC analyst who confirmed that larger incident reports are difficult to understand because of the amount of complex data they include.

We introduce a narrative summarization algorithm for temporal entity relationship graphs, such as incident reports and story narratives. The primary steps are:

1. Specify a summarization threshold $t \in [0,1]$.
2. Calculate a set of scores for each entity e using several criteria, and another set of scores for each connected component C using another set of criteria.
3. Average the scores for each entity and for each component.
4. Remove entities whose average score or whose component's average score is below the summarization threshold.
5. Remove all relationships where either the source or the target entity was removed.

The criteria come from our thought that entities with large numbers of relationships, that have TTPs from later stages of the attack lifecycle, that frequently occur in the incident report, that have longer durations, and that occur more recently in incident reports are likely to be more important to analysts. Indeed, our conversation with the SOC analyst confirmed this.

5.1 Entity Scores

We create a set of scores $S(e)$ for each entity e in the set of entities E using the criteria described below. Each criteria yields an associated score for an entity, unless the score is undefined for the entity.

As a preprocessing step, we first identify all connected components, i.e. all groups of entities (vertices) where a set of relationships (edges) connect each pair of entities in the group. Additionally, for each connected component we identify the entity with the earliest timestamp, the entity with the latest timestamp. Then we conduct a depth-first search to find all relationship paths from the earliest entity to the latest entity, traversing children of each entity (vertex) from highest degree to lowest degree. We consider this the *core sequence of events* (if a component has only one entity then that entity is trivially the core sequence of events for that component).

The first criteria is for entities in the core sequence of events that belong to components with more than one entity. This criteria adds a score of 1 to the entity's set of scores $S(e)$; no score is added for all other entities.

If an entity is not in the core sequence of events and it has a confidence score provided by an analyst or a severity scoring algorithm, then this number is converted to a $[0,1]$ scale where larger numbers indicate higher importance. This number is added to the set $S(e)$. No score is added for all other entities.

Next, for a given connected component C, we induce subgraphs from the set of entities that are not part of the core sequence of events. Each connected component C' in the induced subgraphs can be thought of as a "branch" from the core sequence of events. We calculate a variety of scores for the entities in C'. First, we calculate $\frac{\text{latest timestamp in } C' - \text{ earliest timestamp in } C}{\text{duration of } C}$ and add it to $S(e)$ for all e in C'. Second, we calculate $\frac{\text{latest timestamp in } C' - \text{ earliest timestamp in } C'}{\text{duration of the core sequence of events}}$ and add it to $S(e)$ for all e in C'. Third, we identify all relationship types that are a tactic from the MITRE ATT&CK matrix, convert all tactics to a normalized score by dividing their (zero-indexed) column number n by 13, and add $n/13$ to $S(e)$ for all e in C'. (Because the 14 tactic columns in the MITRE ATT&CK matrix are ordered from least severe to most severe, tactics with higher column numbers are more important to SOC analysts.) Taken together, these criteria generate lower scores for entities in branches that occur earlier in the incident report, are shorter, and include TTPs from earlier stages in the attack lifecycle.

5.2 Component Scores

We also create a set of scores $S(C)$ for each connected component C using a set of criteria described below. Each criteria results in an associated score for a component, unless the score is undefined for the component.

The first criteria relates to the relative duration of the component. We calculate this as $\frac{\text{the duration of } C}{\text{the longest duration of all components}}$ and add it to the set $S(C)$.

The second criteria relates to the relative number of entities in the component. We calculate this as $\frac{\text{the number of entities in } C}{\text{the number of entities in the largest component}}$ and add it to the set $S(C)$.

The third criteria relates to the relative number of relationships in the component. We calculate this as

$\frac{\text{the number of relationships in } C}{\text{the largest number of relationships in all components}}$ and add it to the set $S(C)$.

The fourth criteria relates to the relative number of timestamps in the component. We calculate this as $\frac{\text{the number of timestamps in } C}{\text{the largest number of timestamps in all components}}$ and add it to the set $S(C)$.

The fifth criteria relates to the TTPs in the component. We calculate this by identifying all relationship types in C that are a tactic from the MITRE ATT&CK matrix and finding the largest zero-indexed column number n for all the tactics. Then we calculate $\frac{\text{the largest column number in } C}{\text{the largest column number in all components}}$ and add it to the set $S(C)$. (This takes advantage of the fact that the 14 tactic columns in the MITRE ATT&CK matrix are ordered from least severe to most severe, tactics with higher column numbers are more important to SOC analysts.)

Together, these criteria measure the relative amount of activity occurring in the entities in each component.

5.3 Removing Entities and Relationships

To generate the summary, the algorithm removes an entity e if

$$t > \sum_{s \in S(e)} \frac{s}{|S(e)|} \tag{1}$$

or

$$t > \sum_{s \in S(C)} \frac{s}{|S(C)|} \tag{2}$$

where $t \in [0, 1]$ is the summarization threshold.

However, there are a few cases where entities are not removed regardless of scores or the summarization threshold:

- If the entity's mean score $\sum_{s \in S(e)} \frac{s}{|S(e)|}$ is the highest of all entities in its component, and if the score of its component C has the highest mean score $\sum_{s \in S(C)} \frac{s}{|S(C)|}$ of all components then that entity is never removed. This is to ensure that the summary always has at least one entity.
- If the user specifies that entities selected in the visualization, or entities connected to selected entities, should always be shown, then those entities are never removed. Sect. 4.2 describes how to select entities in the visualization.

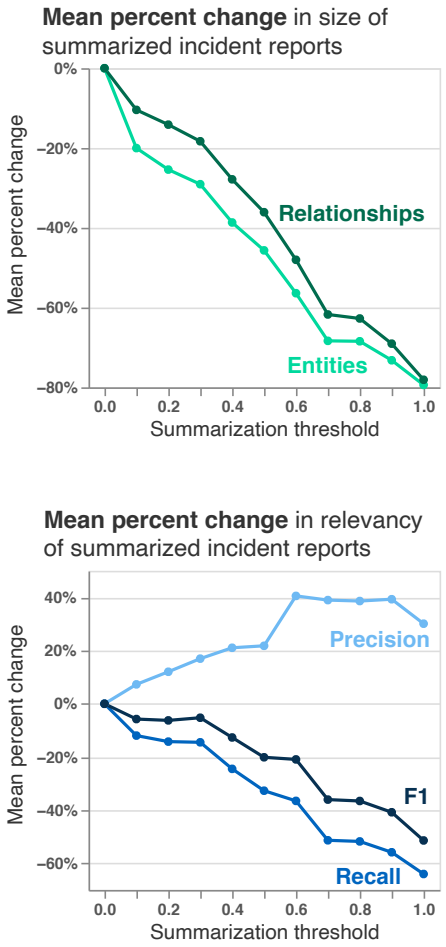

Fig. 4. Mean percent change in the number of entities and relationships (top) and precision, recall, and F1 (bottom) on 15 real incident reports as the summarization threshold varies from 0 to 1 (0 is no summarization, and 1 is maximum summarization).

6 Summarization Evaluation

This evaluation conducts a multi-dimensional analysis of the narrative summarization algorithm on 15 automatically generated alerts from two red team events conducted on an enterprise network. The alerts are designed to mimic incident reports and present a thorough narrative of detected malicious activity. Most alerts describe activity occurring during a single day, but some alerts describe activity spanning many days. Because these were generated during red team events we also have ground truth of victim and attacker entities (IPs, hosts, users, domains, etc.). Fig. 3 shows an anonymized screenshot of one of these incident reports and a summary of it at the 0.6 threshold.

6.1 Size and Complexity

We can think of the number of entities in an incident report as its size. The number of relationships can be a measure of incident report complexity, since relationships add substantial detail and interconnectedness to incident reports.

Fig. 4 (top) shows the mean percent change in number of entities and relationships in the 15 incident reports as we vary the summarization threshold t. These change fairly smoothly and linearly, indicating that the summarization algorithm tends to provide a continuum in its ability to summarize narratives. At 1, the maximum summarization threshold, we see that the mean numbers of both entities and relationships have decreased by about 80%. This is a large reduction in the size and complexity of the incident report.

Many of these incident reports contain a few dozen entities and relationships, which means that we could expect the summarization algorithm to reduce an incident report with 40 entities and 80 relationships to about 8 entities and 16 relationships. This is a substantial reduction that could help analysts more easily understand a summary of the incident.

We can also use binary search to automatically find a summary closest to a desired size. For example, when briefing incidents to others, we might wish to have the largest summary that will fit in the presentation template. Using binary search we can test summarization thresholds until we find a summary closest to the desired size.

6.2 Precision, Recall, and F1 Scores

We also evaluate the quality of the summaries produced by the summarization algorithm. We can compare the entities in the incident reports and the incident report summaries to the ground truth victim and attacker entities from the red team exercises. With these ground truth entity lists we can calculate the precision, recall, and F1 score of the incident reports and the incident report summaries. Precision is the fraction of incident report entities that are true victims or attackers. Recall is the fraction of all victim and attacker entities contained in the incident report. And F1 is $2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$, or equivalently the harmonic mean of precision and recall.

Fig. 4 (bottom) shows the mean percent change in precision, recall, and F1 in the 15 incident reports as we vary the summarization threshold from 0 to 1. As we increase the amount of summarization, mean precision increases and peaks at a 41% increase over the un-summarized incident reports. This indicates that on average the summarization algorithm is removing false positives from the incident reports and

improving their quality. Recall steadily decreases as we increase the amount of summarization. This indicates that the summarization algorithm is removing true positives in addition to false positives. That is not necessarily bad, because we want a summarization algorithm that will generate a concise summary even if some true positives are removed. And this is counterbalanced by the increase in true positive rate (precision).

After analyzing these results, we chose 0.6 as the default summarization threshold. For the incident reports generated on this enterprise network, 0.6 maximizes average precision without reducing recall or F1 to the degree seen at higher summarization thresholds. It also reduces the number of entities by 56% and the number of relationships by 48% on average. This produces a good balance between the quality of the summary and the size of the summary. Users of the visualization tool can choose a custom summarization threshold if they wish to see a bigger or smaller summary.

One limitation is that these incident reports were generated by automated algorithms during red team exercises on a single network. Because of the nature of this data, these results may not generalize to other incident report generation processes on other networks.

6.3 Runtime

The incident reports from our network determine our scalability requirements for the summarization algorithm and the incident report visualization. The largest of these incident reports have more than 100 entities and about 600 relationships, which are extracted from tables with 100–150 rows of network log snippets contained in the incident reports. This determines the performance requirements for our use case. As described below, the summarization algorithm runs in linear time so we expect it to quickly summarize larger data, but the scalability of the visualization has not been tested beyond the incident reports on our network.

The preprocessing steps and summarization algorithm include many steps: find all connected components, determining the core sequences of events with depth-first searches, finding the earliest and latest timestamps, etc. All steps are linear time, and therefore the overall summarization algorithm runs in $O(|E|+|R|)$, where $|E|$ is the number of entities and $|R|$ is the number of relationships.

In practice, the largest incident report we tested had more than 100 entities and about 600 relationships. A commodity laptop ran the summarization algorithm instantaneously on this incident report.

6.4 User Feedback

We made the incident report visualization and summarization tool available to the SOC analyst, who used it on his own without supervision, direction, or a list of tasks. During this time he used it to analyze several incident reports (less than 10). A few weeks later we requested he give us feedback on the incident report visualization and the summarization algorithm.

His comments were overall very positive. He liked the visualization design, saying "I feel like I can look at this and get an understanding of the key parts faster" compared to looking at the tables of data contained in typical incident reports. Regarding the summarizations, he commented "you're going to save me a bunch of time" compared to analyzing unsummarized incident reports.

He also had several suggestions for improvements. As described earlier in this paper, he suggested using a sequential color scale, which is what is shown in all the screenshots in this paper, and also ordering the entities by entity type. He suggested showing information about the underlying data source (e.g. Zeek [30]), but in our environment this data is not included in the STIX data.

He also suggested modifying the time scale at the top of the visualization to have three rows: The top row for the year, the next row for the month, and the bottom row for the day and time. He found the current timescale difficult to read and determine the day and month to contextualize the data he is seeing. He also suggested using a 24-hour time format in UTC time. The current timescale is the default provided by D3.js [3]. We intend to implement this feedback in the next version of the visualization. These comments may be useful to the broader community of visualization designers who are considering using timescales in their own visualizations.

Fig. 5. A summary of the character interaction network from Les Misérables [16] visualized in the incident report visualization. The summarization threshold was set to 1 (maximum summarization). This summary reduces the number of characters from 80 to 19 and includes essentially all major characters.

7 Generalization to Other Domains

To demonstrate that the summarization algorithm can be applied to other domains, we use the algorithm to summarize the character interaction graph of Les Misérables [16]. Although the summarization algorithm includes some criteria that are specific to cyber security, such as criteria related to TTPs, the summarization algorithm simply ignores any criteria that do not apply to a given data set. This characteristic lends the summarization algorithm to applications in other domains.

For Les Misérables, character interactions are recorded per chapter, and interactions are converted to relationships where chapter numbers are converted to relative timestamps. There are no confidence levels associated with the entities, and all the relationship types are `interacts-with`, so no confidence nor TTP information is used by the summarization algorithm.

Fig. 5 shows a visualization of the summarized narrative of the character interaction graph. The summarization threshold is set to 1, the maximum summarization. This reduces the number of characters from 80 to 19. The summary includes essentially all the major characters, although there are a few apparently minor characters included as well.

To assess the quality of this summary we can compare the characters in this automatically generated summary to the characters in human-generated summaries. We identified the 13 characters directly mentioned in the CliffsNotes book summary [20], the 15 characters directly mentioned in the SparkNotes [9] plot overview, and the 12 "major" characters listed in the Wikipedia article [37] on Les Misérables. We also note that both CliffsNotes and SparkNotes reference the "Friends of the ABC," which is a group of revolutionary students in the book. If we consider this to be a reference to all the members of the group, then the CliffsNotes book includes 20 characters and the SparkNotes plot overview includes 22 characters.

In comparison to the human-generated summaries, our automatic summary is a similar size; it has 19 characters, whereas the human-generated summaries range from 12 to 22 characters. Table 1 shows the characters included in each summary.

To measure the similarity of these summaries, we calculate the inter-rater reliability of these character lists using Cohen's kappa coefficient [7] between each pair of summaries, shown in Table 2. The maximum possible is 1, which represents perfect agreement. Kappa ranges 0.60 to 0.87 between the human-generated summaries, and from 0.54 to 0.71 between the automatic summary and the human-generated summaries. We can interpret these results using the guidelines from Landis and Koch [18]: the human-generated summaries have substan-

Table 1. The 24 characters contained in each of the six summaries of Les Misérables. The other 66 characters were not included in any summary and are omitted from this table. Both SparkNotes and CliffNotes reference the character group Friends of the ABC, and the rightmost two columns consider this to be a reference to all the characters in that group and therefore includes all of those characters.

	SparkNotes	CliffsNotes	Wikipedia	**Automatic**	SparkNotes with ABC	CliffsNotes with ABC
Azelma	x				x	
Felix Tholomyes	x			**x**	x	
Gavroche	x		x	**x**	x	
Grantaire			x		x	x
Fauchelevent		x				x
Colonel Pontmercy	x	x			x	x
M Gillenormand	x	x			x	x
Javert	x	x	x	**x**	x	x
Cosette	x	x	x	**x**	x	x
Enjolras	x	x	x	**x**	x	x
Eponine	x	x	x	**x**	x	x
Fantine	x	x	x	**x**	x	x
M Thenardier	x	x	x	**x**	x	x
Marius Pontmercy	x	x	x	**x**	x	x
Mme Thenardier	x	x	x	**x**	x	x
Myriel	x	x	x	**x**	x	x
Valjean	x	x	x	**x**	x	x
Bahorel				**x**	x	x
Bossuet (Lesgle)				**x**	x	x
Combeferre				**x**	x	x
Courfeyrac				**x**	x	x
Jean Prouvaire				**x**	x	x
Joly				**x**	x	x
M Mabeuf				**x**		
Mlle Baptistine				**x**		
Napoleon				**x**		

tial (0.6–0.8) to near perfect (0.8–0.99) agreement with each other, and the automatically generated summary has moderate (0.4–0.6) agreement with CliffNotes (excluding the Friends of the ABC characters) and substantial agreement (0.6–0.8) agreement with the other human-generated summaries.

We interpret these results as a positive outcome. The agreement between the automatically generated summary and the human-generated summaries is not too much lower than the agreement between the human-generated summaries. This difference can be explained because it is expected that human-generated summaries derived from all the detail in the entire book would differ from an automatically generated summary that was created using only a character interaction graph. Additional information and scoring criteria could improve the agreement between the automatically generated summary and the human-generated summaries. For example, we could replace the `interacts-with` relationship type with more descriptive relationship types such as `talks-to` or `attacks`. These new relationship types would have an implicit order of importance analogous to the tactic ordering in the MITRE ATT&CK framework.

From this analysis we believe this summarization algorithm has potential to generate narrative summaries suitable for storyline visualizations, such as that seen in Munroe's original movie narrative chart in Fig. 2.

8 Conclusion

This paper presented a new incident report visualization tool. Because incident reports can include lengthy cyber security logs, we also developed and presented an algorithm to summarize the narrative in cyber security logs and incident reports. The evaluation shows that the summaries can increase the incident report quality by improving precision and increase usefulness by reducing the incident report size and complexity. A SOC analyst was very excited about the tool and viewed it as a way to save time.

Table 2. Cohen's kappa inter-rater reliability between the characters contained in six summaries of Les Misérables.

	SparkNotes	CliffsNotes	Wikipedia	**Automatic**	SparkNotes with ABC	CliffsNotes with ABC
SparkNotes		0.83	0.78	**0.63**	0.76	0.60
CliffsNotes	0.83		0.76	**0.54**	0.61	0.74
Wikipedia	0.78	0.76		**0.64**	0.64	0.62
Automatic	**0.63**	**0.54**	**0.64**		**0.71**	**0.63**
SparkNotes with ABC	0.76	0.61	0.64	**0.71**		0.87
CliffsNotes with ABC	0.60	0.74	0.62	**0.63**	0.87	

We also show the summarization algorithm's ability to generalize to other domains by using it to summarize a dynamic graph of the narrative from Les Misérables. This reduced the size of the graph by 76% while obtaining substantial agreement between its characters and the characters in human-generated summaries. Therefore we believe this algorithm has the potential to summarize narratives for use in other narrative visualizations like storylines.

Future work should examine how to extend the summarization algorithm to also summarize events and relationships. The version presented here focuses on reducing the number of less important entities in the narrative, and only removes a relationship if it is connected to an entity that was removed. This reduces the size and complexity of the narrative graph, but it may be possible to identify less important relationships and remove them to further reduce the complexity and number of events in the summarized narrative.

Acknowledgments

Thanks to Chae Clark, Nathan Danneman, Tony Wong, and the other colleagues for feedback on this work. This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). Distribution Statement "A" (Approved for Public Release, Distribution Unlimited). The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] D. Arendt and M. Pirrung. The "y" of it matters, even for storyline visualization. In *Conference on Visual Analytics Science and Technology*, pp. 81–91. IEEE, 2017.

[2] S. Barnum. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *Mitre Corporation*, 11:1–22, 2012.

[3] M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *IEEE TVCG*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185

[4] B. Bowman, C. Laprade, Y. Ji, and H. H. Huang. Detecting lateral movement in enterprise computer networks with unsupervised graph {AI}. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses*, pp. 257–268, 2020.

[5] C. Brewer. Colorbrewer. `https://colorbrewer2.org/`, 2006. [Online; accessed 03-June-2021].

[6] S. Chen, J. Li, G. Andrienko, N. Andrienko, Y. Wang, P. H. Nguyen, and C. Turkay. Supporting story synthesis: Bridging the gap between visual analytics and storytelling. *IEEE TVCG*, 26(7):2499–2516, 2018.

[7] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[8] C. Dunne and B. Shneiderman. Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In *Proceedings of CHI*, pp. 3247–3256. ACM, 2013.

[9] S. Editors. Les misérables: Study guide. https://www.sparknotes.com/lit/lesmis/, 2005. [Online; accessed 18-June-2021].

[10] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679, 2021.

[11] E. J. Fink. *Dramatic Story Structure: A Primer for Screenwriters*. Taylor & Francis, Milton Park, 2014.

[12] R. Gove and L. Deason. Visualizing automatically detected periodic network activity. In *Symposium on Visualization for Cyber Security*, pp. 1–8. IEEE, 2018.

[13] M. Gronemann, M. Jünger, F. Liers, and F. Mambelli. Crossing minimization in storyline visualization. In *International Symposium on Graph Drawing and Network Visualization*, pp. 367–381. Springer, 2016.

[14] J. Hullman and N. Diakopoulos. Visualization rhetoric: Framing effects in narrative visualization. *IEEE TVCG*, 17(12):2231–2240, 2011.

[15] T. Kakar, X. Qin, T. La, S. K. Sahoo, S. De, E. A. Rundensteiner, and L. Harrison. Sumre: Design and evaluation of a gist-based summary visualization for incident reports triage. *Computer Graphics Forum*, 40(3):263–274, 2021.

[16] D. E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. ACM Press, 1st ed., 1994.

[17] I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, and D. Strash. On minimizing crossings in storyline visualizations. In *International Symposium on Graph Drawing*, pp. 192–198. Springer, 2015.

[18] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *biometrics*, pp. 159–174, 1977.

[19] S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu. Storyflow: Tracking the evolution of stories. *IEEE TVCG*, 19(12):2436–2445, 2013.

[20] A. L. Marsland and G. Klin. Cliffsnotes on les misérables. https://www.cliffsnotes.com/literature/l/les-miserables/book-summary, 2006. [Online; accessed 06-18-2021].

[21] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal event sequence simplification. *IEEE TVCG*, 19(12):2227–2236, 2013.

[22] P. W. Morris. *The Management of Projects*. Thomas Telford, London, 1994.

[23] R. Munroe. Movie narrative charts. https://xkcd.com/657/, 2009. [Online; accessed 07-June-2021].

[24] K. Padia, K. H. Bandara, and C. G. Healey. A system for generating storyline visualizations using hierarchical task network planning. *Computers & Graphics*, 78:64–75, 2019.

[25] D. Rafiei and S. Curial. Effectively Visualizing Large Networks Through Sampling. In *Visualization*, pp. 375–382, 2005.

[26] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE TVCG*, 16(6):1139–1148, 2010.

[27] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the Symposium on Visual Languages*, pp. 336–343. IEEE, 1996.

[28] A. Siddharthan. A survey of research on text simplification. *International Journal of Applied Linguistics*, 165(2):259–298, 2014.

[29] S. Silvia, R. Etemadpour, J. Abbas, S. Huskey, and C. Weaver. Visualizing variation in classical text with force directed storylines. In *Workshop on Visualization for the Digital Humanities*. ACM, 2016.

[30] R. Sommer. Bro: An open source network intrusion detection system. *Security, E-learning, E-Services, 17. DFN-Arbeitstagung über Kommunikationsnetze*, 2003.

[31] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas. Mitre ATT&CK: Design and philosophy. Technical report, The MITRE Corporation, McLean, VA, July 2018.

[32] Y. Tanahashi and K.-L. Ma. Design considerations for optimizing storyline visualizations. *IEEE TVCG*, 18(12):2679–2688, 2012.

[33] T. Tang, R. Li, X. Wu, S. Liu, J. Knittel, S. Koch, L. Yu, P. Ren, T. Ertl, and Y. Wu. Plotthread: Creating expressive storyline visualizations using reinforcement learning. *IEEE TVCG*, 27(2):294–303, 2020.

[34] T. Tang, S. Rubab, J. Lai, W. Cui, L. Yu, and Y. Wu. istoryline: Effective convergence to hand-drawn storylines. *IEEE TVCG*, 25(1):769–778, 2018.

[35] T. C. van Dijk, M. Fink, N. Fischer, F. Lipp, P. Markfelder, A. Ravsky, S. Suri, and A. Wolff. Block crossings in storyline visualizations. *J. Graph Algorithms Appl.*, 21(5):873–913, 2017.

[36] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang. Datashot: Automatic generation of fact sheets from tabular data. *IEEE TVCG*, 26(1):895–905, 2019.

[37] Wikipedia contributors. Les misérables — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Les_Mis%C3%A9rables&oldid=1026995562, 2021. [Online; accessed 18-June-2021].

[38] Y. Wu, W. Wu, S. Yang, Y. Yan, and H. Qu. Interactive visual summary of major communities in a large network. In *2015 Pacific Visualization Symposium*, pp. 47–54. IEEE, 2015.