KRAKEN: A Knowledge-Based Recommender system for Analysts, to Kick Exploration up a Notch

Romain Brisse, Simon Boche, Frédéric Majorczyk, Jean-François Lalande

KRAKEN: A Knowledge-Based Recommender system for Analysts, to Kick Exploration up a Notch*

Romain Brisse[1,2][0000−0002−7132−1966], Simon Boche[1], Frédéric Majorczyk[2,3], and Jean-Francois Lalande[2][0000−0003−4984−2199]

[1] Malizen, Rennes, France
[2] CentraleSupélec, Inria, Univ. Rennes 1, CNRS, IRISA, Rennes, France
[3] Direction Générale de l'Armement

Abstract. During a computer security investigation, a security analyst has to explore the logs available to understand what happened in the compromised system. For such tasks, visual analysis tools have been developed to help with log exploration. They provide visualisations of aggregated logs, and help navigate data efficiently. However, even using visualisation tools, the task can still be difficult and tiresome. The amount and the numerous dimensions of the logs to analyse, the potential stealthiness and complexity of the attack may end with the analyst missing some parts of an attack. We offer to help the analyst finding the logs where her expertise is needed rapidly and efficiently. We design a recommender system called KRAKEN that links knowledge coming from advanced attack descriptions into a visual analysis tool to suggest exploration paths. KRAKEN confronts real world adversary knowledge with the investigated logs to dynamically provide relevant parts of the dataset to explore. To evaluate KRAKEN we conducted a user study with seven security analysts. Using our system, they investigated a dataset from the DARPA containing different Advanced Persistent Threat attacks. The results and comments of the security analysts show the usability and usefulness of the recommender system.

Keywords: Attacks and Defences · Intrusion detection and prevention system · Digital Forensics

1 Introduction

IT systems are the target of an ever-growing number of attacks. Their complexity ranges from simple attacks like brute-force or DDoS, to complex APT. To defend those systems, IT organizations implement CSOC (Computer Security Operation Center) [37] where security analysts try to understand what happens in the

* We thank the participants to our evaluation, and all the members of Malizen for their help and support. This work was supported by a CIFRE-Defense grant from Agence Innovation Defense (AID).

monitored systems and to react accordingly. Among popular inputs there are security alerts raised from IDS (Intrusion Detection Systems) [11,28,26]. Their goal is to detect suspicious activities that are symptoms of an intrusion. Then, a security analyst is in charge of investigating if alerts are related to a real attack.

To complement IDSs, visualisation tools [13,31,12,8,32,3] have been developped to identify attacks in the data. Among those tools, some have focused on log visualisation [15,16]. Because of the complexity of monitored systems, the quantity of events logged and their complexity, the necessary time to investigate is too long [9]. Visualisation systems also require extensive field knowledge to be used efficiently [4]; as a consequence, users have to learn their usage.

Recommender systems have been proposed as a complementary approach to visualisation tools to address these issues. They are mostly designed to help the user choose a better representation of the data [22,33,34,35,14]. These approaches are not specific to security investigations and require advanced knowledge and practice in terms of visualisation. Visualisation recommendation [23] has also been studied for security purposes. In [23], the authors focus on designing a visualisation tool that can better guide analysts during their work. However, these solutions have their shortcomings such as the lack of reliable data to make recommendations, the heterogeneity and quantity of data to explore. They also suffer from specific issues, such as the cold start problem [25], where the recommender system has not gathered enough data to make relevant decisions yet.

In this paper we present a recommender system that helps the analyst in her visual investigation by suggesting exploration options, either to test a hypothesis on the incident, or to analyse another part of the logs that has not been explored yet. The proposed recommender system does not require any input from past usage of the visual tools it is based on. It uses a knowledge base of adversary tactics and techniques extracted from real-world observations: the MITRE ATT&CK matrix[4]. Finally, we present an evaluation of our recommender system based on a campaign of investigations conducted with seven security experts.

The rest of the paper is organised as follows. Related work is discussed in Section 2. Context about visualisation and investigations in security is explained in Section 3. Section 5 details the dataset used for the evaluation, the user interviews and then discusses results. Finally, in Section 6, we conclude the article and discuss future work.

2 Related work

To help analysts during security investigations, many visualisation tools have been proposed by the research community to analyse various event data such as network logs [32,8], DNS logs [29], system logs [15,16] or file system metadata [3]. These methods allow a faster and easier investigation giving the analyst the possibility to query and visualise large amount of complex data. They however require significant expertise in both security and visualisation techniques. Recommender systems are starting to be used used to tackle that issue.

[4] `https://attack.mitre.org/`

Historically, recommender systems [18,27] have been used by commercial websites to present a user with a set of relevant options such as new books to read or new TV shows to watch. Recommender systems can base their recommendations on similarities between user profiles, item metadata, or even domain knowledge [1,5]. However, some kind data, such as user profiles of item values often cannot be used in the security field because it cannot be allowed to be retained due to its sensitivity. The recommender system has to rank the different options extracted from the data, generally by computing a score for each option and by presenting a subset of the options to the end user: recommendation candidates. Their goal is to help users make better and faster decisions by presenting them with relevant options.

Outside cybersecurity, previous work combines recommendations with visualisation [34,35,10,33,14]. When a recommender system is used with a visualisation system, the recommendations are mainly used to offer to the analyst the more useful representations. This can be seen as an extension of work about automatic representation [21,22]. The recommendations can be computed using statistical and perceptual measures [34,35] or using machine learning [14]. As a consequence multiple visualisation options are offered to a user, which has to decide which one is the best suitable to her needs. We believe that the required level of expertise is too high to be usable in real use cases. Therefore, this paper introduces a recommender system that helps to make the best use of a security visualisation tool instead of recommending the best possible representations of security data.

Only few works are related to the visualisation of security data enhanced by a recommender system [23,36]. Li et al. focus on security risk analysis and offer defensive measures recommendations [20]. Zhong et al. [36] present a recommender system aiming to help tier 1 analysts in CSOC with data triage, based on the experience of senior analysts. These two contributions need reliable data and have to face the cold start problem. The closest work to ours is NAVSEC [23], a recommender system integrated with a 3D visualisation tool [24] for network data. During the investigation, NAVSEC will regularly offer to the security analyst a set of interactions with the 3D visualisation tool so as to discover a possible intrusion. The best interactions are selected by a nearest-neighbor approach based on a database of previous investigations conducted by an expert security analyst. NAVSEC is a collaborative recommender system; it does not consider the user's need or query in the recommendations and does not benefit from the accumulated knowledge on attacks. The nearest-neighbor approach also relies on data from past investigations, which makes this recommender system suffer from the cold start problem [25] as well.

3 Visualising and investigating

In this section we discuss the nature of logs and summarise the principles of an investigation. Then, we briefly describe ZeroKit, a visualisation tool that

an analyst can use to explore logs. ZeroKit is the technological base of our recommender system presented in Section 4.

3.1 Log investigation

Log files contain a list of events generated by a monitoring tool such as an IDS or any program that records events, such as a SMTP or a web server. Each event is composed of multiple different fields. A field has a data type and a value. To efficiently describe log data, we use data types as defined by Elastic Common Schema (ECS)[5]. Typically, in security investigations we use from a dozen up to 50 data types, such as *ip* or *action*.

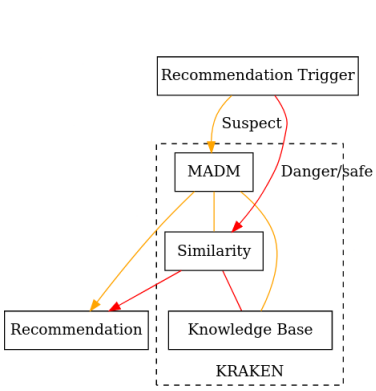
Investigating consists in finding potential threats, risky behaviours, or security flaws by analysing the provided logs. During those investigations, analysts explore multiple log sources; some have common data types: pivot types. Pivot types are common data types that allow the user to navigate between log files when they study interesting values. Navigating in the data is more convenient when a visualisation tool offers an understandable frontend. In the following section, we present the tool used in this paper to perform the investigations.

3.2 ZeroKit

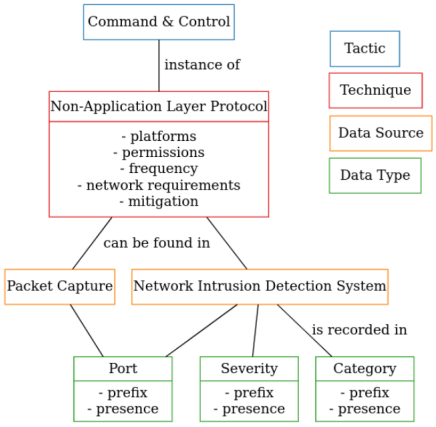
ZeroKit is a collaborative log analysis and incident response tool, aiming to put data visualisation at an analyst's disposal. An analyst can explore logs through interactive and reactive data visualisation. The three main actions she can do in ZeroKit are visualising the distribution of values of a data type, filter by a value or filter using time. Each action operated by the analyst refreshes the visualisation panes. Thus, a sequence of actions constitutes a path of exploration, and the user can navigate along this path by doing new actions or going back by undoing some. Additionally, to mark an item of interest, an analyst can flag an item such as an IP address and add context by choosing a severity: *safe*, *suspect* or *danger*. It shows the discovery of an attack artifact, or on the contrary, the absence of findings. The decision about severities will be the starting point of the recommendation, and is discussed later.

The following example illustrates some steps of an investigation of the VAST 2012 dataset[6] using ZeroKit. It is composed of the logs from an IDS monitoring the network of a small enterprise and the logs from the border router. Analysing the logs from the router, the analyst observes many connections to the TCP port 6667. That port is related to IRC, which is not a commonly used protocol in an enterprise environment. Those connections are thus suspicious and the analyst flags this value as suspect. She needs now to confirm that the activity related to this destination port is linked to an attack or not. At that point of the investigation, there are still 43 thousand events related to that destination port and the whole process can take hours.

[5] `https://www.elastic.co/guid`
[6] `http://www.vacommunity.org/VAST+Challenge+2012`



(a) Recommendation process



(b) Extract from the knowledge base

Fig. 1: KRAKEN

3.3 Recommendation

Our goal is to enhance ZeroKit by offering recommendations about the next step the analyst should take to confirm or deny her hypotheses. Offering a potentially relevant path for exploration can save a significant amount of time to analysts during an investigation. Using security knowledge as the base for making recommendations, KRAKEN can rapidly help the analyst confirming or denying that she found an attack by using real expertise, without the issue of collecting the data first. It also provides assistance in the investigation without requiring non-security expertise such as complex visualisation processes.

4 Recommender system

We built KRAKEN as a knowledge-based recommender system [5]. As knowledge-based recommender systems take the user's query and domain knowledge as inputs, they do not suffer from cold start. Similarly to other recommender systems [27], KRAKEN tries to ensure three properties:

- **R1:** Enhance user efficiency and rapidity in their tasks.
- **R2:** Offering recommendations at a relevant time, without any disturbance.
- **R3:** Making sure to avoid overlooking important information.

4.1 Recommendation process

Figure 1a shows the recommendation process of KRAKEN. First, when an analyst flags a value as safe, suspect or danger, KRAKEN triggers the recommendation process. Recommendations use a Similarity scoring process to compute

recommendations. Recommendation are also computed using Multi-Attribute Decision-Making (MADM) in one case, as discussed in Section 4.4. Both those methods use the knowledge base as input. Finally, the three best candidates are displayed to the analyst.

4.2 Recommendation triggers

In general, guessing the true intent of the user in a visual tool is complex. Too many actions are possible and depending on the context the same action can have a different intent. In our work, we focus on the action of flagging a value with a severity, because it is the closest action related to a security decision we can find from an user. It allows us to make hypotheses about his opinion as an expert. We trigger a recommendation when the user puts a flag that represents a specific intent **(R2)**:

1. A suspect flag means that the analyst needs more information before deciding whether the value is linked to malicious activity or not.
2. A danger flag means that a threat artifact has been found and, as such, ends this part of the investigation. The analyst wants to direct his attention somewhere else.
3. A safe flag means that the threat has not been found yet, or that there is none. The analyst wants to take a look at the situation from another angle.

4.3 Knowledge base

ATT&CK's goal consists in recording detailed, real and observed adversarial techniques and to categorise them in tactics. It is used to characterise security threats, and as a common and shared vocabulary with all cybersecurity actors. Each specified adversarial technique includes many usable attributes, and notably sources of logs where they can be observed. We mapped these sources to the data types defined by ECS[7]. The resulting knowledge base allows us to gather data types useful to observe attacks within logs.

To illustrate the use of the ATT&CK matrix for spawning recommendations, we show in Figure 1b an extract of the knowledge base that corresponds to the use case of Section 3.2. That extract shows that the port data type is associated with the "Packet Capture" source, in turn associated with the technique "non-application layer protocol". From there, we can select all the data sources linked to the technique ("Network Intrusion Detection System" in the example) and all data types linked to them: "Severity" and "Category". The analyst was analysing suspect traffic coming from IRC, and upon flagging the port as suspect, she is given the recommendation containing the "Severity" and "Category" data types. By following these recommendations the analyst finds alerts raised by the NIDS about attempted information leaks and corporate policy violations.

[7] `https://www.elastic.co/guid`

Algorithm 1 Our recommendation algorithm

procedure RECOMMENDATION(dt_0) ▷ With dt_0: flagged data type
  $Ds_0 \leftarrow AllDataSourcesLinkedTo(dt_0)$
  $T_{all} \leftarrow AllTechniquesLinkedTo(Ds_0)$
  if severity is *suspect* **then**
    for all techniques in T_{all} **do**
      perform MADM scoring.
      $T_{best} \leftarrow scored\_technique$
    $T_{best} \leftarrow OrderByScore(T_{all})$ ▷ The best scored techniques
    $Ds_0 \leftarrow Ds_0 + AllDataSourcesLinkedToBestTechniques(T_{best})$
  $Dt_{all} \leftarrow AllDataTypesLinkedTo(Ds_0)$
  $Dt_{filtered} \leftarrow FilterIrrelevantDataTypes(Dt_{all})$
  for all data types in $Dt_{filtered}$ **do**
    perform Similarity scoring.
    $Dt_{best} \leftarrow scored\_datatype$
  $Dt_{best} \leftarrow OrderByScore(Dt_{best})$ ▷ The scored data types
  return three best candidates from Dt_{best}

4.4 Decision-making

To compute the recommendations, we implemented a decision-making algorithm that uses as input the knowledge base presented in the previous section and the severity of the flag. In the case of a safe or danger flag, we only used a Similarity scoring method, whereas for a suspect flag we implemented MADM, on top of the similarity score. We did so because we needed to score techniques from the knowledge base, and their attributes are far more complex and less comparable than those of data types.

Algorithm 1 shows how recommendations are generated. The decision-making process is developed in further sections hereafter. All functions that appear in this algorithm represent queries to the knowledge base. Data types categorised as irrelevant in function *FilterIrrelevantDataTypes()* are those that are not present in the investigation, those who only have one value through the dataset and the flagged data type itself.

MADM Due to their complex attributes, technique objects from ATT&CK[8] are difficult to rank. We choose an additive Analytical Hierarchy Process (AHP) [2,17] to do so: the Simple Additive Weighting (SAW).

Simple Additive Weighting is a decision-making process that relies on partial orders determined by our security knowledge, to compute a score. We use the attributes associated to a technique as a list of criteria (*i.e.* platforms, permissions, network requirements, frequency, mitigation and data sources). The process is divided into two phases: the creation of a consistent Pairwise Comparison Matrix (PCM) [19] and the computation of candidate scores, which is executed each

[8] `https://attack.mitre.org/`

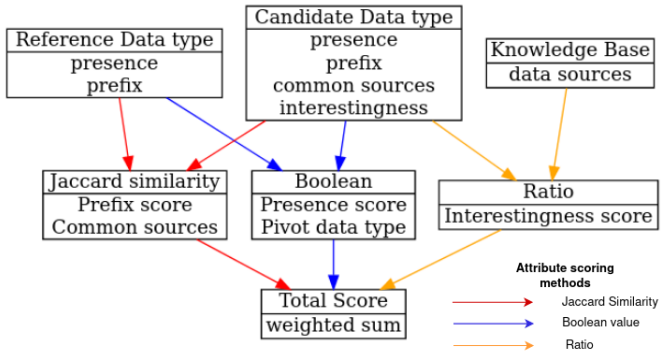


Fig. 2: A flowchart of the scoring process using similarity

time a recommendation is needed. After verifying that it is indeed consistent, this matrix is used to compute an overall weight for each criterion. From there we can score candidate techniques. Each of these steps are described in more detail in the appendices.

Similarity Figure 2 shows how a data type's attributes are scored using our Similarity scoring process. The reference data type is the data type of the value that was flagged by the analyst. All attributes are scored separately. A final score is computed using a weighted sum of all previously obtained sub-scores. For example, the prefix score is computed using the Jaccard similarity between the prefixes set present in the candidate date type and the prefixes set present in the reference data type. Prefixes are all the values that can precise a data type, such a "source" or "event". Common sources refer to the ATT&CK data sources and are scored in the same manner as prefixes. The presence and pivot attributes check if the data type is already present in the investigation and if it can be used as a pivot between already available log files in the investigation. The interestingness is a ratio of data sources where this data type can be recorded over all possible data sources. It is essentially an inverse rarity. Once we have all subscores, we evaluated their relative importance and determined weights, which we apply to them to compute the final score. If the final score between two data types approaches zero, the data types are similar, and dissimilar if the score approaches one.

5 Evaluation

We conducted an evaluation to gather feedback on how well KRAKEN met the requirements **R1**, **R2** and **R3**. In security it is difficult to find enough experts to get a strong statistical result. Consequently, the evaluation is mostly qualitative.

APT	Attack step	Flags	Discovery threshold	Investigation ratio
APT 1	Firefox ad	2	1	10%
	Firefox extension	2	1	10%
	SSH	2	2	20%
	Wget	2	1	20%
APT 2	Pine	3	2	20%
	Tcexec Malware	1	1	20%

Table 1: APTs present in the TC3 dataset

5.1 Datasets

For the evaluation we use a subset of the TC3 (Transparent Computing exercise 3) dataset[9]. TC3 has been released by the DARPA as part of their "Transparent Computing" program. The subset of TC3 that we used was captured in identical conditions, but at a much smaller scale in order to limit the number of threats to find during the evaluation. This subset contains 19.5 million system call events, from one machine, targeted by the APTs. The data types that can be found in this subset are grouped in different object types: file, memory, network, unnamed pipes, and sinks. The subset contains two APT described in Table 1. The two Firefox exploits aim to gain access to the machine, while SSH is used for network discovery. Wget is used to exfiltrate data. Pine is an old text-based email client here used to provide a backdoor into the machine and spread a malware: tcexec.

5.2 Experimental setup

After a short presentation of our work on recommender systems, we asked the participants about their experience in cybersecurity. We also asked if they had some previous experience in CSOC or with a SIEM, in order to classify them in three categories: low, medium and high experienced analysts. Then, we did a rapid presentation of the subset of TC3 used for the investigation. Next, we demonstrated the key features of ZeroKit. After their investigations, we collected their feedback through a qualitative interview.

Qualitative interview The discussion was informal, yet we guided the participants to obtain answers to specific questions, each trying to assess a different aspect of KRAKEN. They are enumerated thereafter:

Q1. **Usefulness (R1):** were the recommendations useful to your investigation?
Q2. **Efficiency (R2):** did KRAKEN help you gain efficiency in your search?
Q3. **Relevance (R2):** did KRAKEN offer you relevant recommendations?
Q4. **Tool future:** in the future would you use KRAKEN during investigations?
Q5. **Clarity (R3):** did you find the recommendations clear and easy to grasp?

[9] `https://github.com/darpa-i2o/Transparent-Computing/blob/master/README-E3.md`

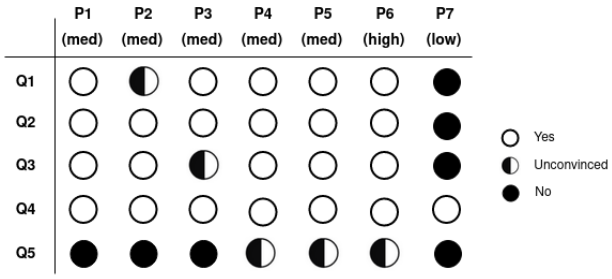


Fig. 3: Answers of evaluation participants to the questionnaire

Quantitative measures During all the investigations, we collected traces of user actions. The main variables we recorded and used to analyse the investigations are: number of flags, recommendations, followed recommendations and the proportion of threats discovered during the 25-minutes investigation. Each step has a discovery threshold, and the threat coverage is computed from the number of attack steps found by each participant. These measures help us quantify the usefulness of the recommendations. Table 1 shows the statistical importance we gave to each attack step in regards to the overall threat coverage.

5.3 User feedback

Figure 3 shows the answers of the participants to the questionnaire from Section 5.2. The white dot shows a positive answer, the black a negative one and the black and white a mitigated answer.

The majority of participants affirms that KRAKEN was useful to them and accelerated them during this investigation. They found the recommendations useful as they helped them progress in their investigations. In terms of efficiency, their feedback is consistent with the fact that they were all able to find parts or all of the APTs in only 25 minutes. The answers given by participant seven can be explained by his low experience as an analyst. During his investigation, he had little idea of how to proceed so he could not make use of the tool properly.

The results show that all the participants were enthusiast about the future of the tool. They all saw the benefits in terms of efficiency during an investigation that this research suggests **(R3)**. They agreed to say that this tool helped them get better coverage of the dataset and guided them in the right direction.

The sore spot of the evaluation was the clarity of recommendations. All users felt that they were not highlighted enough in the interface **(R3)**. However, once familiar with the investigation interface, they were all able to use KRAKEN properly and even said it did not cause unwanted distractions during their work.

5.4 Recommendation relevance

Table 2 shows measures about recommendations aggregated by flag severity: the total number of recommendations made during all investigations, the number of

Flag Severity	Recommendations	Distinct data types	Relevant	Followed
Safe	5	2	80%	60%
Suspect	13	5	100%	84.6%
Danger	8	3	50%	12.5%

Table 2: Recommendation relevance according to its associated severity

distinct data types concerned by those recommendations, the ratios of relevant recommendations and the ratios of recommendations followed by the analyst over the total number.

All recommendations triggered by a *suspect* flag were relevant **(R2)**. The results for the danger and safe severity are less categorical. The recommendations triggered by a safe flag seems to have mostly provided the participants with relevant recommendation according to our hypotheses. On the other hand, the recommendations triggered by a danger flag were less relevant.

Most suspect flag recommendations were followed, showing that not only we were able to provide relevant recommendations, we were also able to convey them to the analyst properly. Safe flag recommendations were also followed 60% of the time, meaning that we have mostly well interpreted the analyst's intent for it. However, the danger flag recommendations are only followed 12.5% of the time. We noticed that analysts would often flag as danger and then start back from that point to find other threats, possibly linked to the one already found.

5.5 Providing assistance to investigations

Figure 4 is a scatter plot of the overall threat coverage in function of the proportion of recommendations followed for each analyst. The analyst's experience is also represented by a colour.

Figure 4 shows that participant seven, who had little to no experience found few attacks in the dataset and did not use the recommendation, as discussed previously. However, the rightmost point shows that by selecting a majority of recommendations, the experienced analyst achieved very satisfying results **(R2)**.

Figure 4 also shows that, in the case of mid-level experts, the recommendations do not help the analysts discover more than 50% of the threats. While 50% of threat coverage is a good result in 25 minutes, even if our prototype offers relevant recommendations, interpreting them still requires expert skill.

6 Conclusion

During the last few years, new tools have been designed to help security analysts in their investigations using visualisations recommendation. However, analysing security incidents is still a tiresome task. Exploration recommendations using expert knowledge can significantly help analysts.

In this paper, we presented a recommender system aiming to help the security analyst in her investigation. KRAKEN suggests new paths to explore within log

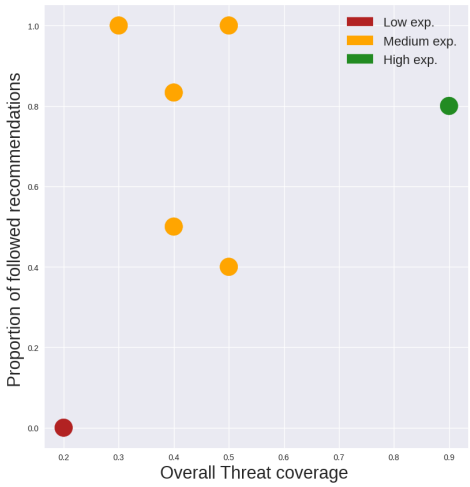


Fig. 4: Overall threat coverage discovered by each analyst correlated with the proportion of followed recommendations

data. It is composed of a knowledge base linking techniques, tactics, data sources and data types, two scoring processes and several recommendation triggers.

We evaluated KRAKEN with seven cybersecurity experts, whose experience as analysts were various. Our evaluation shows that recommendations are relevant most of the time, and when followed help security analysts during incident response. Participants to the evaluation also noted that the recommender system did not distract them during their investigations while providing insight.

Following the feedback from the evaluation, we implemented some features enhancing user experience such as a history of recommendations and a better explanation of the recommendations. This lays the groundwork for a future evaluation on a larger scale, with non security specialists. The results of such a study will help us reach a better understanding of all possible use cases for KRAKEN. Since we saw that the user intent associated with a severity is more complex than we thought, a larger pool of participants could better frame user intents.

In the near future we aim to work on some technical issues that we identified during the development and the evaluation of KRAKEN as well as larger issues. For example, the SAW decision-making model is known to overrate objects that have extreme values. Later on, we would also like to hybridise [7,6] other types of recommender systems with KRAKEN. For example, a collaborative filtering recommender system would allow to include user actions from past investigations. With that hybridisation, recommendations would be based on user habits, expertise and commonly relevant investigation paths as well as expert knowledge, and so could be more useful to analysts.

Appendix

MADM scoring method

Designing the Pairwise Comparison Matrix While PCM are an effective and widely used solution, they have to respect a rule of consistency, depending on the scale used: ratio scales, geometric scales and logarithmic scale. The simplicity of the ratio scales method presented by Saaty *et al.* [30] and how we use the final score in KRAKEN make the ratio scale a good fit for our decision making process. Two guidelines are proposed by Saaty *et al.* to build a PCM with a high level of consistency:

- Using an adapted scale depending on the number of criteria (presented in Table 3) to clearly differentiate answers.
- Keeping pairwise consistency: $a_{i_j} = 1/a_{j_i}$, is a necessary but not sufficient condition. Although Saaty *et al.* specify that "improving consistency does not mean getting an answer closer the real life solution", a balance is to be found between perfect mathematical consistency and reality for the scoring to be relevant.

	2	3	4	5	6	7
1-5	0	0.244	0.335	0.472	0.479	0.527
1-7	0	0.515	0.504	0.708	0.798	0.827
1-9	0	0.416	0.851	1.115	1.150	1.345

Table 3: A measure of inconsistency between the PCM's order and the scale used [30]

Computing normalised weights Using the PCM, we compute normalised weights to obtain values bounded between 0 and 1. The weight or the i^{th} criteria corresponds to the sum of the i^{th} row divided by the total sum of the matrix. They are obtained by the following formula:

$$w_i = (\sum_{i,j=0}^{n} PCM_{i_j})/T$$

with n the number of criteria and T the total sum of the PCM.

Convert data to numerical values Some of our criteria are not numerical values, like the set of platforms and the permissions required. Using our expertise, we ranked and weighed the possible values of each attribute and computed a ratio for each of them.

Check Consistency Rate Before using it, the consistency of the PCM must be checked. The process is the following, as explained in [2]:

1. Find all eigenvectors and eigenvalues for the matrix.

2. Find the maximum inconsistency by taking the maximum possible eigenvalue.
3. Calculate the consistency index:
$$CI = (\lambda_max - n)/(n-1)$$
where n is the matrix size.
4. Finally, compute the Consistency Rate (CR):
$$CR = CI/RI$$
with RI the Random Index for consistency, or, in other words, the average consistency obtained when filling the PCM at random.
If CR is inferior or equal to 0.1, then the matrix is considered consistent. This operation is only necessary once. From the moment a PCM is determined to be consistent, it can be used in the decision-making process.

We computed different PCMs using our own security expertise and checked their consistency. We found that trying to generally rank the criteria gave back large inconsistencies in our matrices. So, we focused on more specific security goals when deciding the importance of a criteria, such as detection difficulty and accessibility. Table 4 shows one of the resulting (and consistent) matrices we built.

.	Plat.	Perm.	Net. Reqs.	Freq.
Platforms	1	1	0.25	0.5
Permissions	1	1	0.33	0.5
Network Requirements	4	3	1	3
Frequency	2	2	0.33	1

Table 4: Designing a consistent PCM using our own knowledge

Compute scores Reaching this step, scores are computed every time a recommendation is requested, using the PCM that was previously determined to be consistent.

1. For each scored attribute of each candidate s, apply the following formula (for positive scores only) within the $n * m$ matrix composed of m candidates and n criteria:
$$s_{ij} = r_{ij}/r^*_j$$
with $i = 1, ..., m$, $j = 1, ..., n$, and r^*_j being the maximum value of r in column j.
2. The total score is the sum of each attribute's score, multiplied by its previously computed weight.

Relative and absolute scoring The scores use the maximum recorded value among the candidates for each criterion, and not the maximum possible value. This is to avoid cases where we would obtain bad scores for every candidate.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering **17**(6), 734–749 (2005). https://doi.org/10.1109/TKDE.2005.99
2. Afshari, A., Mojahed, M., Yusuff, R.M.: Simple additive weighting approach to personnel selection problem. International journal of innovation, management and technology **1**(5), 511 (2010)
3. Beran, M., Hrdina, F., Kouřil, D., Ošlejšek, R., Zákopčanová, K.: Exploratory analysis of file system metadata for rapid investigation of security incidents. 2020 IEEE Symposium on Visualization for Cyber Security (VizSec) pp. 11–20 (2020). https://doi.org/10.1109/VizSec51108.2020.00008
4. Bertin, J., Barbut, M.: Semiology of Graphics: Diagrams, Networks, Maps. Ed. de l'EHESS (2005)
5. Burke, R.: Knowledge-based recommender systems. In: Encyclopedia of library and information systems. vol. 69, pp. 175–186 (2000)
6. Burke, R.: Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction **12**(4), 331–370 (2002). https://doi.org/10.1023/A:1021240730564
7. Burke, R.: Hybrid web recommender systems. pp. 377–408. Springer (2007)
8. Cappers, B.C., van Wijk, J.J.: Snaps: Semantic network traffic analysis through projection and selection. In: 2015 IEEE Symposium on Visualization for Cyber Security (VizSec). pp. 1–8. IEEE (2015). https://doi.org/10.1109/VIZSEC.2015.7312768
9. Cremilleux, D., Bidan, C., Majorczyk, F., Prigent, N.: VEGAS: Visualizing, exploring and grouping alerts. In: NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium. pp. 1097–1100. IEEE (2016). https://doi.org/10.1109/NOMS.2016.7502968
10. Cui, Z., Badam, S.K., Yalçin, M.A., Elmqvist, N.: Datasite: Proactive visual data exploration with computation of insight-based recommendations. Information Visualization **18**(2), 251–267 (2019). https://doi.org/10.1177/1473871618806555
11. Denning, D.E.: An intrusion-detection model. IEEE Transactions on Software Engineering **SE-13**, 222–232 (1987). https://doi.org/10.1109/TSE.1987.232894
12. Fischer, F., Keim, D.A.: NStreamAware: real-time visual analytics for data streams to enhance situational awareness. In: Proceedings of the Eleventh Workshop on Visualization for Cyber Security. pp. 65–72. ACM (2014). https://doi.org/10.1145/2671491.2671495
13. Foresti, S., Agutter, J.: Visalert: From idea to product. pp. 159–174. Springer (2008)
14. Hu, K., Bakker, M.A., Li, S., Kraska, T., Hidalgo, C.: VizML: A Machine Learning Approach to Visualization Recommendation. In: Proceedings of the 2019 Conference on Human Factors in Computing Systems (CHI). ACM (2019). https://doi.org/10.1145/3290605.3300358
15. Humphries, C., Prigent, N., Bidan, C., Majorczyk, F.: Elvis: Extensible log visualization. In: Proceedings of the Tenth Workshop on Visualization for Cyber Security. p. 9–16. VizSec 2013, Association for Computing Machinery (2013). https://doi.org/10.1145/2517957.2517959

16. Humphries, C., Prigent, N., Bidan, C., Majorczyk, F.: CORGI: Combination, organization and reconstruction through graphical interactions. In: 2014 IEEE Symposium on Visualization for Cyber Security (VizSec). pp. 57–64. IEEE (2014). https://doi.org/10.1145/2671491.2671494
17. Ishizaka, A., Balkenborg, D., Kaplan, T.: Influence of aggregation and measurement scale on ranking a compromise alternative in ahp. Journal of the Operational Research Society **62**(4), 700–710 (2011). https://doi.org/10.1057/jors.2010.23
18. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender systems: an introduction (2010)
19. Kou, G., Ergu, D., Lin, C., Chen, Y.: Pairwise comparison matrix in multiple criteria decision making. Technological and economic development of economy **22**(5), 738–765 (2016). https://doi.org/10.3846/20294913.2016.1210694
20. Li, T., Convertino, G., Tayi, R.K., Kazerooni, S.: What data should i protect?: recommender and planning support for data security analysts. In: Proceedings of the 24th International Conference on Intelligent User Interfaces. pp. 286–297. ACM (2019). https://doi.org/10.1145/3301275.3302294
21. Mackinlay, J.: Automating the design of graphical presentations of relational information. ACM Transactions On Graphics (Tog) **5**(2), 110–141 (1986). https://doi.org/10.1145/22949.22950
22. Mackinlay, J., Hanrahan, P., Stolte, C.: Show me: Automatic presentation for visual analysis. IEEE Transactions on Visualization and Computer Graphics **13**(6), 1137–1144 (2007). https://doi.org/10.1109/TVCG.2007.70594
23. Nunnally, T., Abdullah, K., Uluagac, A.S., Copeland, J.A., Beyah, R.: *NAVSEC*: a recommender system for 3d network security visualizations. In: Proceedings of the Tenth Workshop on Visualization for Cyber Security - 2013 IEEE Symposium on Visualization for Cyber Security (VizSec). ACM Press (2013). https://doi.org/10.1145/2517957.2517963
24. Nunnally, T., Chi, P., Abdullah, K., Uluagac, A.S., Copeland, J.A., Beyah, R.: P3d: A parallel 3d coordinate visualization for advanced network scans. In: 2013 IEEE International Conference on Communications (ICC). pp. 2052–2057 (2013). https://doi.org/10.1109/ICC.2013.6654828
25. Park, S.T., Chu, W.: Pairwise preference regression for cold-start recommendation. In: Proceedings of the third ACM conference on Recommender systems - RecSys '09. p. 21. ACM (2009). https://doi.org/10.1145/1639714.1639720
26. Paxson, V., Campbell, S., Lee, J., et al.: Bro intrusion detection system. Tech. rep., Lawrence Berkeley National Laboratory (2006)
27. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Recommender systems handbook, pp. 1–35. Springer (2011)
28. Roesch, M., et al.: Snort: Lightweight intrusion detection for networks. In: Lisa. vol. 99, pp. 229–238 (1999)
29. Romero-Gomez, R., Nadji, Y., Antonakakis, M.: Towards designing effective visualizations for DNS-based network threat analysis. In: 2017 IEEE Symposium on Visualization for Cyber Security (VizSec). pp. 1–8. IEEE (2017). https://doi.org/10.1109/VIZSEC.2017.8062201
30. Saaty, T.L.: A scaling method for priorities in hierarchical structures. Journal of mathematical psychology **15**(3), 234–281 (1977). https://doi.org/10.1016/0022-2496(77)90033-5
31. Theron, R., Magán-Carrión, R., Camacho, J., Fernndez, G.M.: Network-wide intrusion detection supported by multivariate analysis and interactive visualization. In: 2017 IEEE Symposium on Visualization for Cyber Security (VizSec). pp. 1–8. IEEE (2017). https://doi.org/10.1109/VIZSEC.2017.8062198

32. Ulmer, A., Sessler, D., Kohlhammer, J.: Netcapvis: Web-based progressive visual analytics for network packet captures. In: 2019 IEEE Symposium on Visualization for Cyber Security (VizSec). pp. 1–10 (2019). https://doi.org/10.1109/VizSec48167.2019.9161633
33. Vartak, M., Parameswaran, A., Polyzotis, N., Madden, S.R.: Seedb: automatically generating query visualizations. Proceedings of the VLDB Endowment **7**, 1581–1584 (2014). https://doi.org/10.14778/2733004.2733035
34. Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J.: Voyager: Exploratory analysis via faceted browsing of visualization recommendations. IEEE Transactions on Visualization and Computer Graphics **22**(1), 649–658 (2016). https://doi.org/10.1109/TVCG.2015.2467191
35. Wongsuphasawat, K., Qu, Z., Moritz, D., Chang, R., Ouk, F., Anand, A., Mackinlay, J., Howe, B., Heer, J.: Voyager 2: Augmenting visual analysis with partial view specifications. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. p. 2648–2659. Association for Computing Machinery (2017). https://doi.org/10.1145/3025453.3025768
36. Zhong, C., Lin, T., Liu, P., Yen, J., Chen, K.: A cyber security data triage operation retrieval system. Computers & Security **76**, 12–31 (2018). https://doi.org/10.1016/j.cose.2018.02.011
37. Zimmerman, C.: The strategies of a world-class cybersecurity operations center. The MITRE Corporation (2014)