Bandit on the Hunt: Dynamic Crawling for Cyber Threat Intelligence

Philipp Kuehn[1], Dilara Nadermahmoodi[1], Markus Bayer[1], and Christian Reuter[1]

Science and Technology for Peace and Security (PEASEC),
Technical University of Darmstadt, Germany

Abstract. Public information contains valuable Cyber Threat Intelligence (CTI) that is used to prevent future attacks. While standards exist for sharing this information, much appears in non-standardized news articles or blogs. Monitoring online sources for threats is time-consuming and source selection is uncertain. Current research focuses on extracting Indicators of Compromise from known sources, rarely addressing new source identification. This paper proposes a CTI-focused crawler using multi-armed bandit (MAB) and various crawling strategies. It employs SBERT to identify relevant documents while dynamically adapting its crawling path. Our system ThreatCrawl achieves a harvest rate exceeding 25% and expands its seed by over 300% while maintaining topical focus. Additionally, the crawler identifies previously unknown but highly relevant overview pages, datasets, and domains.

Keywords: Focused crawling · Security · Classification · Multi-armed bandit.

1 Introduction

In Cyber Threat Intelligence (CTI) information is used to learn from current threats and prevent similar attacks against infrastructures. This is done by sharing actionable information such as Indicators of Compromise (IOCs) through various channels. It borrows its intelligence generation procedure from the intelligence cycle used by intelligence services [31]. CTI has established standards to publish, import, and export information to databases and platforms [26]. But CTI is often shared in unstructured formats like blog posts or threat reports [12]. Manually scanning online posts for IOCs is time-consuming for personnel. Hinchy [11] surveyed 468 full-time security analysts, finding over half spend most time on manual tasks, believe automation is possible, and may change jobs without modern tools. Kaufhold et al. [13] confirmed these results with participants stating they lack capacity to monitor all media and need more automation.

IOC extractors and threat detection methods are being developed [19,18], along with vulnerability severity predictors [9,15,16]. However, this research assumes the right information sources. The Internet is dynamic, with sites changing focus, ceasing publication, or emerging anew. Despite a decade of research, CTI cycle methods still fail to support practitioners in basic information collection [31,22]. Manual source identification conflicts with the existing information overload facing security personnel.

Maintaining current CTI-relevant websites requires unavailable manual work. This creates blind spots for active attack campaigns. Time spent searching for CTI results in less effective staff [13]. This reduces infrastructure security as less time is spent on actual protection. A precise approach to finding relevant web pages quickly is necessary. Focused crawlers "selectively seek out pages that are relevant to a pre-defined set of topics" [5]. This work focuses on identifying CTI-related information published online. We aim to answer the research question *"How can CTI related information be identified and crawled from the web?"* ***(RQ)***

This work combines techniques for crawling, classifying, and ranking content in our ThreatCrawl pipeline. It gathers specific CTI domain information from the surface web. Our proposal uses Sentence-BERT (SBERT) embeddings to decide which sources to follow **(C1)**. Documents are classified by information type and ranked by domain suitability **(C2)**.

First, we provide an overview of related work in §2. §3 presents the theoretical concept of ThreatCrawl, followed by initialization description in §4, while §5 evaluates the system. §6 discusses the evaluation, limitations and future work, and §7 concludes this work.

2 Related Work

Traditional document embeddings like Term Frequency-Inverse Document Frequency (TF-IDF) are surpassed by context-aware, deep-learning embeddings [7,23]. Reimers and Gurevych [27] build on Bidirectional Encoder Representations from Transformers (BERT) with SBERT for generating sentence and document embeddings.

Several focused crawlers use TF-IDF as embedding method [34,20,25]. Zhang et al. [35] propose finding datasets lacking metadata using a multi-armed bandit (MAB) focused crawler [25]. Koloveas et al. [14] propose an integrated Machine Learning (ML)-based crawler for managing CTI information using ACHE and Gensim. Sanagavarapu et al. [30] propose a cybersecurity-specific search engine.

While some CTI research focuses on Twitter/X for easy access [33,28], others use known sites [19]. Both methods rely on known sources without expanding view. Since Twitter/X's leadership change, crawling it became more difficult. Dekel et al. [6] use MAB to prioritize investigated attacks in CTI datasets.

Current work combining web crawling, classification, and ranking for CTI in a single pipeline is widely discussed. Tawil and Alqaraleh [32] describe a crawler using SBERT embeddings for document labeling. This approach separates crawling and classifying processes, crawling everything before classification. Koloveas et al. [14] present a different two-step approach, reducing focused crawling benefits with a harvest rate of 9.5 %. Computer Emergency Response Teams (CERTs) and Security Operations Centers (SOCs) personnel already monitor a small set of domains defining their infrastructure scope. An open question remains developing a general one-step focused crawling approach combining crawling, classification, and ranking of CTI-relevant content based on known URLs.

3 Methodology

The goal of the present work is the identification of new, previously unknown web pages, that are related to the user's input URLs. We combine the different concepts proposed in [25], integrate new technology, and tailor them to the requirement of security personnel. This work builds on top of [17].

3.1 Notation

We denote $\mathbb{P}$ as the set of all web pages. Given two pages $p, p' \in \mathbb{P}$, $p \rightarrow p'$ indicate, that p links to p' and $p \sim_\theta p'$ indicates that p is contextually similar to p' with regard to a relevance threshold θ. *theta* is omitted if it is clear from context. We extend $\sim$ to sets, *i.e.*, $p \sim_\theta P \subseteq \mathbb{P} \iff \exists p' \in P.p \sim_\theta p'$. Similarly, $P, P' \subseteq \mathbb{P}$, $P \sim \theta P' \iff \forall p \in P.p \sim \theta P'$. The function $crawl_a(p) \rightarrow \mathcal{P}(\mathbb{P})$ denotes a crawl of page $p \in \mathbb{P}$ based on a crawl action a, which returns a set of pages.

3.2 Problem Definition

One of the key aspects of this system is the relevance of information. Recent studies show a shift of relevance in the CTI domain[1] from very detailed information like IOCs to broader information like threat or malware reports (*cf.* Table 1). Therefore, we calculate the relevance of pages based on their similarity to the used seed $s \in S \subseteq \mathbb{P}$, rather than using binary information like the existence of IOC information [19]. Our approach might miss dataset pages that present IOC information, *e.g.*, through simple lists, but this is already covered by [25]. Def. 1 defines the problem, we aim to solve.

Definition 1 (CTI-focused crawler). *Given a set of seed pages $S \subseteq \mathbb{P}$ defining the scope of ones CTI infrastructure security information, we want to identify $P \subseteq \mathbb{P}$, such that $P \sim S$.*

We propose an architecture we call ThreatCrawl. It uses SBERT embeddings, and a one-step approach, *i.e.*, classification during the crawling process to adjust the crawling direction on the fly. It implements similar retrieval methods as [25] combined with a UCB1-MAB. The retrieval methods are (i) forward (F), (ii) backlink (B), and (iii) keyword search (K), *i.e.*, $crawl_a$ with $a \in [F, B, K]$. Given a page $p \in \mathbb{P}$. *Forward search* crawls all links provided on p, *i.e.*, $crawl_F(p) = \{p' \in \mathbb{P} \mid p \rightarrow p'\}$. *Backlink search* crawls all pages, which

[1] Based on an annual survey conducted by the SANS Institute[2] in which they survey security professionals from various organizations.

	2020	2021	2022
#1	IOCs	Textual information about targeted vulnerabilities	Textual information of used malware
#2	TTPs	Textual information of used malware	Textual information about targeted vulnerabilities
#3	Adversary analysis	IOCs	Broad information about attacker trends

Table 1. Top 3 most useful Cyber Threat Intelligence (CTI) types according to the SANS CTI surveys from the years 2020 to 2022 [29,2,3].

link to p, *i.e.*, $crawl_B(p) = \{p' \in \mathbb{P} \mid p' \rightarrow p\}$. *Keyword search* searches for pages containing keywords of p, *i.e.*, $crawl_K(p) = \{p' \in \mathbb{P} \mid p'$ contains keyword of $p\}$. Those methods are proven to provide a broad coverage during focus crawling [24,4,8], while still providing the possibility to dive deep in a source through forward searches.

4 ThreatCrawl

This section provides a general overview of the used system to answer our research question (*cf.* §4.1), followed by insights into the search actions (*cf.* §4.2), classification method (*cf.* §4.3), and MAB selection (*cf.* §4.4).

4.1 Overview

We propose ThreatCrawl (*cf.* Def. 1) to answer our research question *"How can CTI related information be identified and crawled from the web"*. Its central operation is bound on the user-provided set of seed URLs $S \subseteq \mathbb{P}$ and the used crawling methods F, B, K. An overview of the whole system is presented in Fig. 1. S is crawled as ground truth for the classification and to provide the initial steps for crawling and prepare the MAB. This is done using a priority queue based on the relevance of a given page. For the seed pages $p \in S$ the priority is set to the highest possible value. Following this step, the MAB is initialized with a discovery phase to calibrate the arm selection our search methods. All pages that are discovered this way can be directly classified as relevant through their content and if relevant, pushed into the priority queue with their similarity to S directly translated as priority, *i.e.*, more similar pages are picked first. Afterward, the priority queue is processed one step at a time, popping the most relevant page and using the MAB to select the most rewarding arm until one of the stop conditions is reached (*cf.* §3). This ensures the highest possible outcome measured as *harvest rate*, *i.e.*, $\frac{|R|}{|P|}$, where $R, P \subseteq \mathbb{P}$ is the set of relevant pages and the set of all seen pages by the crawler, respectively.

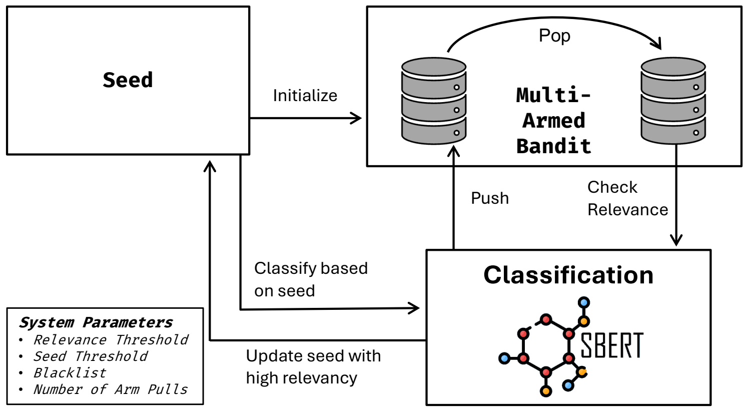

Fig. 1. A schematic overview of our proposed CTI source identification system ThreatCrawl leveraging a multi-armed bandit (MAB) and classification based on Sentence-BERT (SBERT) [27].

4.2 Search Actions

Forward Link Search Forward link search follows all hyperlinks on a page p, offering broad website coverage and uncovering in-depth content.

Backward Link Search Backward link search, commonly used in SEO tools, identifies pages linking back to p to discover related content. Since no open sources provide this data, we rely on a commercial API.

Keyword Search Keyword search extracts key terms from p to guide searches, using tools like *KeyBERT* [10], *YAKE*, or *RAKE* [1]. We use KeyBERT and combine the top three keywords with `OR` logic to ensure broader results. The keyword search itself is done with commercial API.

4.3 Relevance Classification

The content of a page p is embedded using SBERT [27]. Pages and their embeddings are used interchangeably. SBERT provides dense vector representations capturing semantic similarity between pages. Compared to TF-IDF or Word2Vec, SBERT excels at contextual nuances and generates sentence-level embeddings better suited for semantic search tasks. This allows contextual representation even where keyword-based approaches fail to capture deeper meanings. Once embedded, cosine similarity between vectors of pages p, q is calculated. Cosine similarity measures vector similarity independent of magnitude, ideal for comparing web page semantics. For pages p, q and page set Q we define $sim(p,q) = \cos\_\mathrm{sim}(p,q)$ and $sim(p,Q) = \max\left([\cos\_\mathrm{sim}(p,q) | \; q \in Q]\right)$. Two pages p, q are semantically related $p \sim q \iff sim(p,q) \geq \theta$, for relevance threshold θ. A page p and page set Q are semantically related $p \sim Q \iff sim(p,Q) \geq \theta$.

4.4 Multi-Armed Bandit

Multi-armed bandit algorithms are crucial for balancing exploration and exploitation in focused crawling. We provide a comparison of MAB algorithms in Table 2.

Table 2. Comparison of multi-armed bandit algorithms.

Algorithm	Best For	Strengths	Weaknesses
Epsilon-Greedy	Static or semi-static environments	Simple to implement; tunable exploration	Inefficient in complex, large environments
UCB1	Stable environments, long-term performance	Logarithmic regret, efficient exploration-exploitation	Slow to adapt to dynamic changes
EXP3	Adversarial or highly dynamic environments	Robust to non-stationary rewards, no explicit exploration	Over-explores in well-structured settings
EXP3-IX	Non-stationary but less adversarial environments	Better balance of exploration, robust	More complex to implement, requires tuning
Sliding-Window UCB	Non-stationary, fast-changing environments	Adapts to time-varying rewards	Needs careful window size tuning

Based on Table 2, UCB1 is the best fit for our setting based on its strong performance in stable environments[3], balancing exploration and exploitation with logarithmic regret. It efficiently uses stable domain knowledge to optimize long-term performance without complex tuning. Though slower to adapt to changes, its effectiveness in consistent environments makes it ideal for forward crawling tasks. Eq. (1) provides the reward function $r \in \mathbb{R}_{\geq 0}$ for a step $pull(p)$ for page p, the set of seeds S, a domain weight δ, and the number of discovered $|domains|$.

$$r(p) = \max(\delta \ |domains(pull(p))| \ + \sum_{p' \ in \ pull(p)} p' \sim S, 0) \tag{1}$$

5 Evaluation

The evaluation was performed on a machine with an Intel Core i7-8565U processor and 16GB of RAM. A relevance threshold of 0.6 was used to classify relevant pages. The seed threshold was set to 0.8, as a higher threshold of 0.9 resulted

[3] Stable in terms of information it seeks, rather than the environment of the web as a whole, which is, as discussed, a dynamic environment.

in nearly identical outcomes, which was undesirable for exploration. A blacklist excluded social media platforms like X and YouTube, thread aggregators, non-HTTP protocols, and file types such as images, documents, and program-specific formats. The number of steps was set to 500 and 2 000 to allow sufficient exploration actions. The seed consisted of 17 pages, covering security news and in-depth reports.

5.1 Evaluation

We evaluate our system based on the combinations of the different search actions, where **B** indicates backward search, **F** indicates forward search, and **K** indicates keyword search.

`TC_BFK` Balance broad coverage, deep relevance, and keyword-driven exploration.
`TC_BF` Follows forward links and analyzes backlinks to discover related pages, providing both broad and targeted exploration.
`TC_FK` Follows forward links and uses keyword searches to expand coverage, capturing more diverse content.
`TC_BK` Analyzes backlinks and performs keyword searches for targeted exploration and broader discovery.
`TC_F` Follows forward links from seed pages, offering extensive coverage but with potential inefficiencies.
`TC_B` Analyzes backlinks to uncover related pages not directly linked to the seed.
`TC_K` Search solely based on keywords of given pages.

Table 3 presents the results of the evaluation. For 500 steps we reached a maximum harvest rate of $\sim 23.86\%$ with `TC_F`, followed by `TC_FK` with $\sim 20.6\%$. The worst result achieved `TC_K` with a harvest rate of just 2.7%, which, overall, performed the worst of all combinations. When backlink search was present in the crawl, it performed the best, expect for `TC_BFK`. In terms of domains, `TC_BK` was able to spread the search the widest with 873 searched domains with 132 of them being relevant, despite crawling only 2 344 pages (including the seed). Those included multiple security overview and dataset pages[4], as well as multiple unknown security news pages.

During the 2 000 step test, we saw much more balanced harvest rates over the used methods (near to or over 20%, except `TC_B` and `TC_K`) with forward search gathering the most relevant pages on average. The top harvest rate was achieved by `TC_BK` with 25.14%. Keyword search alone still performed the worst but combining it with backlink search achieved the highest harvest rate and gathered the highest amount of relevant domains. In total, nearly a quarter of all crawled pages are of different domains (with `TC_BK`) showing the impact of the keyword search through search engines, and we identified in total 270 relevant domains (with `TC_BF`).

[4] E.g, `https://malpedia.caad.fkie.fraunhofer.de` and `https://github.com/CyberMonitor/APT_CyberCriminal_Campagin_Collections`.

Table 3. Result of the evaluation with a maximum of 500 and 2 000 steps. $|S|$, $|P|$, and $|P{+}|$ show the number of steps (if not the maximum), the number of total crawled pages, and number of relevant pages, of which are $|Seed|$ new seeds, HR the harvest rate [in %], and $\lceil\sim\rceil$ the maximum similarity of identified pages. $|Dom|$ and $|Dom{+}|$ indicate how many domains and relevant domains are identified, respectively. TM shows the top method during this run with $\overline{\text{TM}}$ as the average similarity of this method. Best results are highlighted with light background color, and the worst darker background color.

	Method	S	$\|P\|$	$\|P{+}\|$	$\|Seed\|$	HR	$\lceil\sim\rceil$	$\|Dom\|$	$\|Dom{+}\|$	TM	$\overline{\text{TM}}$
500	`TC_BFK`	128	6 199	387	24	6.24	0.90	751	96	F	0.11
	`TC_BF`	11	405	32	6	7.90	0.90	145	24	B	0.17
	`TC_FK`		9 110	1 877	30	20.60	1.00	384	23	F	0.25
	`TC_BK`	278	2 344	339	9	14.46	0.90	873	132	B	0.20
	`TC_F`		9 982	2 382	28	23.86	0.87	244	38	F	0.29
	`TC_B`	111	1 313	97	12	7.39	0.90	397	61	B	0.15
	`TC_K`	21	148	4	1	2.70	0.81	105	4	K	0.24
2 000	`TC_BFK`		21 663	4 889	46	22.57	0.92	1 484	189	F	0.38
	`TC_BF`		26 715	4 965	56	18.59	0.92	1 683	270	F	0.29
	`TC_FK`		18 270	4 119	39	22.55	1.00	809	49	F	0.27
	`TC_BK`		8 175	2 055	19	25.14	0.90	1 965	269	B	0.27
	`TC_F`		21 710	4 992	48	22.99	1.00	372	52	F	0.27
	`TC_B`	117	1 290	102	12	7.91	0.90	387	61	B	0.15
	`TC_K`	22	155	5	1	3.23	0.81	111	5	K	0.24

6 Discussion, Limitations & Future Work

ThreatCrawl effectively addresses the challenge of identifying and crawling CTI related sources from the web, which answers our research question "How can CTI related information be identified and crawled from the web" **(RQ)**. By utilizing a MAB approach and seed URLs, the system efficiently expands relevant web content, offering a targeted crawling strategy that is well-suited for CERT and SOC personnel. Unlike related crawlers [25,14], ThreatCrawl focuses on expanding a predefined set of pages, aligning with the current needs and priorities instead of keywords. In terms of efficiency, the system significantly outperforms related work. While [14] reported a harvest rate of $\sim$ 9.5%, our system achieves a much higher rate of $\sim$ 25.14%, indicating a more effective method for discovering relevant information. ThreatCrawl **(C1)** is able to identify key information sources relevant to the initial search domain and even expand the current seed by over 300% without relaxing its focus. It went over the course of nearly 2 000 different web domains and identified 270 relevant ones with starting just 17 seeds **(C2)**.

Limitations Despite these positive results, ThreatCrawl faces some key limitations. Firstly, it should be noted that the related page search functionality, as described in [35], is no longer available due to the removal of this feature by search engines[5]. This removal has resulted in the loss of a key capability for searching across a wide range of sources. Second, no comparisons with others [14] was conducted, preventing a broader benchmarking of performance besides the one stated in the references directly. Runtimes exceeding 2 000 steps were not evaluated, so we did not observe how the crawler behaved over time or whether it reaches saturation with certain search actions. Tipping points for the crawling parameters are not evaluated either, *e.g.*, using to lax thresholds, using to distinct seeds, or using to few seeds. Finally, embeddings are generated using pretrained SBERT models rather than larger models like LLaMA or GPT, which could offer improved semantic accuracy but at the cost of higher computational demands and additional privacy considerations.

Future Work Future work could include dynamic adjustment of thresholds based on real-time crawler performance, allowing better adaptability. User feedback from CERT and SOC personnel could also be integrated to guide the system's relevance assessments. Building on that, while using a MAB is definitely a top choice to decide, which actions are more promising in the long run and if there is no information of the page that is used for the current search action. But with crawling this information is just a `GET` away. For example, a very detailed page with very few links could yield better results with keyword or backlink search. Otherwise, if the page is very new, backlink would probably yield worse results than the others. Using graph-based approaches to map and analyze relationships between crawled pages could provide deeper insights into the structure of the CTI landscape. This could be combined with local-sensitive hashing to identify news aggregation platforms and pages that copy others, as well as, the first publisher of information. Other improvements could be a multistep-depth crawling, *i.e.*, if an irrelevant page is reached keep crawling for n steps just to be sure. Such an approach could be combined with URL classification [21] or adaptable domain blacklisting. While the focus of this paper is the CTI domain, the system should perform well on other domains too, since it is primarily based on the used seed. This aspect needs to be evaluated further.

7 Conclusion

CTI information is published in unstructured form on the web, which presents a time-consuming task for CERTs and SOCs to maintain an up-to-date list of web pages to visit for such information. Our proposed ThreatCrawl system addresses the challenge of identifying previously unknown and relevant CTI sources from the vast amount of unstructured public information available online. By using a MAB approach, it efficiently expands a seed of URLs, making it highly suitable for security personnel who need to automate this time-consuming task, even with a small amount of seed URLs ≤ 20. With a harvest rate of over 25%,

[5] Query a search engine with *"related:page_url"* returned related pages, excluding those of the given domain.

ThreatCrawl outperforms prior work, uncovering previously unknown information sources and datasets. However, limitations like short evaluation runtimes and reliance on SBERT leave opportunities for further enhancement. Future work should focus on optimizing search actions, adjusting thresholds dynamically, and leveraging larger models for better accuracy and adaptability.

Acknowledgement. This work has been co-funded by the LOEWE initiative (Hesse, Germany) within the emergenCITY centre[6] and by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts in their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

1. Amur, Z.H., Hooi, Y.K., Soomro, G.M., Bhanbhro, H., Karyem, S., Sohu, N.: Unlocking the Potential of Keyword Extraction: The Need for Access to High-Quality Datasets. Applied Sciences **13**, 7228 (2023). https://doi.org/10.3390/app13127228
2. Brown, R., Lee, R.M.: SANS Cyber Threat Intelligence Survey 2021. Tech. rep., SANS (2021)
3. Brown, R., Stirparo, P.: SANS Cyber Threat Intelligence Survey 2022. Tech. rep., SANS (2022)
4. Chakrabarti, S., Gibson, D.A., McCurley, K.S.: Surfing the Web backwards. Computer Networks **31**, 1679–1693 (1999). https://doi.org/10.1016/S1389-1286(99)00042-0
5. Chakrabarti, S., van den Berg, M., Dom, B.: Focused crawling: A new approach to topic-specific Web resource discovery. Computer Networks **31**, 1623–1640 (1999). https://doi.org/10.1016/S1389-1286(99)00052-3
6. Dekel, L., Leybovich, I., Zilberman, P., Puzis, R.: MABAT: A Multi-Armed Bandit Approach for Threat-Hunting. IEEE Transactions on Information Forensics and Security **18**, 477–490 (2023). https://doi.org/10.1109/TIFS.2022.3215010
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NORD'19 **1**, 4171–4186 (2018). https://doi.org/10.18653/v1/N19-1423
8. Diligenti, M., Coetzee, F., Lawrence, S., Giles, C.L., Gori, M.: Focused Crawling Using Context Graphs. In: Proceedings of the 26th International Conference on Very Large Data Bases. pp. 527–534. Morgan Kaufmann Publishers Inc. (2000)
9. Elbaz, C., Rilling, L., Morin, C.: Fighting N-day vulnerabilities with automated CVSS vector prediction at disclosure. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. pp. 1–10. ACM (2020). https://doi.org/10.1145/3407023.3407038
10. Grootendorst, M.: KeyBERT. Zenodo (2021)
11. Hinchy, E.: Voice of the SOC Analyst. Tech. rep., Tines (2022)
12. Husari, G., Al-Shaer, E., Ahmed, M., Chu, B., Niu, X.: TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources. In: Proceedings of the 33rd Annual Computer Security Applications Conference.

[6] Funding code: [LOEWE/1/12/519/03/05.001(0016)/72]

pp. 103–115. Association for Computing Machinery (2017). https://doi.org/10.1145/3134600.3134646
13. Kaufhold, M.A., Riebe, T., Bayer, M., Reuter, C.: 'We Do Not Have the Capacity to Monitor All Media': A Design Case Study on Cyber Situational Awareness in Computer Emergency Response Teams. In: CHI'24. pp. 1–16. Association for Computing Machinery (2024). https://doi.org/10.1145/3613904.3642368
14. Koloveas, P., Chantzios, T., Alevizopoulou, S., Skiadopoulos, S., Tryfonopoulos, C.: inTIME: A Machine Learning-Based Framework for Gathering and Leveraging Web Data to Cyber-Threat Intelligence. Electronicsweek **10**, 818 (2021). https://doi.org/10.3390/electronics10070818
15. Kuehn, P., Bayer, M., Wendelborn, M., Reuter, C.: OVANA: An Approach to Analyze and Improve the Information Quality of Vulnerability Databases. In: ARES '21: Proceedings of the 16th International Conference on Availability, Reliability and Security. p. 11. ACM (2021). https://doi.org/10.1145/3465481.3465744
16. Kuehn, P., Relke, D.N., Reuter, C.: Common vulnerability scoring system prediction based on open source intelligence information sources. Computers & Security (2023). https://doi.org/10.1016/j.cose.2023.103286
17. Kuehn, P., Schmidt, M., Bayer, M., Reuter, C.: ThreatCrawl: A BERT-based Focused Crawler for the Cybersecurity Domain (2023). https://doi.org/10.48550/arXiv.2304.11960
18. Le Sceller, Q., Karbab, E.B., Debbabi, M., Iqbal, F.: SONAR: Automatic Detection of Cyber Security Events over the Twitter Stream. In: ARES'17. pp. 1–11. ACM (2017). https://doi.org/10.1145/3098954.3098992
19. Liao, X., Yuan, K., Wang, X., Li, Z., Xing, L., Beyah, R.: Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In: CSS'16. vol. 24-28-Octo, pp. 755–766. ACM Press (2016). https://doi.org/10.1145/2976749.2978315
20. Liu, W., He, Y., Wu, J., Du, Y., Liu, X., Xi, T., Gan, Z., Jiang, P., Huang, X.: A focused crawler based on semantic disambiguation vector space model. Complex & Intelligent Systems **9**, 345–366 (2023). https://doi.org/10.1007/s40747-022-00707-8
21. Mahdaouy, A.E., Lamsiyah, S., Idrissi, M.J., Alami, H., Yartaoui, Z., Berrada, I.: DomURLs_BERT: Pre-trained BERT-based Model for Malicious Domains and URLs Detection and Classification (2024). https://doi.org/10.48550/arXiv.2409.09143
22. Oosthoek, K., Doerr, C.: Cyber Threat Intelligence: A Product Without a Process? International Journal of Intelligence and CounterIntelligence **34**, 300–315 (2021). https://doi.org/10.1080/08850607.2020.1780062
23. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep Contextualized Word Representations. In: NAACL'18. pp. 2227–2237. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/N18-1202
24. Pham, K., Santos, A., Freire, J.: Learning to Discover Domain-Specific Web Content. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. pp. 432–440. Association for Computing Machinery (2018). https://doi.org/10.1145/3159652.3159724
25. Pham, K., Santos, A., Freire, J.: Bootstrapping Domain-Specific Content Discovery on the Web. In: The World Wide Web Conference. pp. 1476–1486. Association for Computing Machinery (2019). https://doi.org/10.1145/3308558.3313709

26. Preuveneers, D., Joosen, W.: Privacy-Preserving Polyglot Sharing and Analysis of Confidential Cyber Threat Intelligence. In: Proceedings of the 17th International Conference on Availability, Reliability and Security. pp. 1–11. Association for Computing Machinery (2022). https://doi.org/10.1145/3538969.3538982
27. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1410
28. Riebe, T., Wirth, T., Bayer, M., Kuehn, P., Kaufhold, M.A., Knauthe, V., Guthe, S., Reuter, C.: CySecAlert: An Alert Generation System for Cyber Security Events Using Open Source Intelligence Data. In: Information and Communications Security (ICICS). pp. 429–446 (2021). https://doi.org/10.1007/978-3-030-86890-1_24
29. Robert M Lee: Cyber Threat Intelligence Survey. https://www.sans.org/white-papers/39395/ (2020)
30. Sanagavarapu, L.M., Mathur, N., Agrawal, S., Reddy, Y.R.: SIREN - Security Information Retrieval and Extraction eNgine. In: Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A. (eds.) Advances in Information Retrieval. pp. 811–814. Springer International Publishing (2018). https://doi.org/10.1007/978-3-319-76941-7_81
31. Sauerwein, C., Fischer, D., Rubsamen, M., Rosenberger, G., Stelzer, D., Breu, R.: From Threat Data to Actionable Intelligence: An Exploratory Analysis of the Intelligence Cycle Implementation in Cyber Threat Intelligence Sharing Platforms. In: Proceedings of the 16th International Conference on Availability, Reliability and Security. pp. 1–9. Association for Computing Machinery (2021). https://doi.org/10.1145/3465481.3470048
32. Tawil, Y., Alqaraleh, S.: BERT Based Topic-Specific Crawler. In: 2021 Innovations in Intelligent Systems and Applications Conference (ASYU). pp. 1–5 (2021). https://doi.org/10.1109/ASYU52992.2021.9599076
33. Tundis, A., Ruppert, S., Mühlhäuser, M.: A Feature-driven Method for Automating the Assessment of OSINT Cyber Threat Sources. Computers & Security **113**, 102576 (2022). https://doi.org/10.1016/j.cose.2021.102576
34. Wang, W., Chen, X., Zou, Y., Wang, H., Dai, Z.: A Focused Crawler Based on Naive Bayes Classifier. In: 2010 Third International Symposium on Intelligent Information Technology and Security Informatics. pp. 517–521 (2010). https://doi.org/10.1109/IITSI.2010.30
35. Zhang, H., Santos, A., Freire, J.: DSDD: Domain-Specific Dataset Discovery on the Web. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 2527–2536. Association for Computing Machinery (2021). https://doi.org/10.1145/3459637.3482427