IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Exploring Dark Web Crawlers: A systematic literature review of dark web crawlers and their implementation

JESPER BERGMAN[1], OLIVER B. POPOV[2] (Member, IEEE)
[1]Department of Computer and Systems Sciences, (e-mail: jesperbe@dsv.su.se)
[2]Department of Computer and Systems Sciences, (e-mail: popov@dsv.su.se))

Corresponding author: Jesper Bergman (e-mail: jesperbe@dsv.su.se)

The work is partially supported by the NordForsk Grant No. 80512 for the project "Police Detectives on the TOR-network".

ABSTRACT Strong encryption algorithms and reliable anonymity routing have made cybercrime investigation more challenging. Hence, one option for law enforcement agencies (LEAs) is to search through unencrypted content on the Internet or anonymous communication networks (ACNs). The capability of automatically harvesting web content from web servers enables LEAs to collect and preserve data prone to serve as potential leads, clues, or evidence in an investigation. Although scientific studies have explored the field of web crawling soon after the inception of the web, few research studies have thoroughly scrutinised web crawling on the "dark web" or via ACNs such as I2P, IPFS, Freenet, and Tor. The current paper presents a systematic literature review (SLR) that examines the prevalence and characteristics of dark web crawlers. From a selection of 58 peer-reviewed articles mentioning crawling and the dark web, 34 remained after excluding irrelevant articles. The literature review showed that most dark web crawlers were programmed in Python, using either Selenium or Scrapy as the web scraping library. The knowledge gathered from the systematic literature review was used to develop a Tor-based web crawling model into an already existing software toolset customised for ACN-based investigations. Finally, the performance of the model was examined through a set of experiments. The results indicate that the developed crawler was successful in scraping web content from both clear and dark web pages, and scraping dark marketplaces on the Tor network. The scientific contribution of this paper entails novel knowledge concerning ACN-based web crawlers. Furthermore, it presents a model for crawling and scraping clear and dark websites for the purpose of digital investigations. The conclusions include practical implications of dark web content retrieval and archival, such as investigation clues and evidence, and the related future research topics.

INDEX TERMS cybercrime, digital forensics, systematic literature review, dark web crawling, Tor

I. INTRODUCTION

The high level of confidentiality and limited traceability have made cybercrime on the Internet, particularly on the so-called dark networks, considerably more challenging to investigate. The tight protection of data travelling through the dark networks has rendered law enforcement with tedious and complex tasks that require extraordinary resources in terms of time, labour, knowledge, and competence.

There several different software available that execute code which connects a computer to one of the circa half-dozen dark network available today. Despite being different, they all have in common that they use state-of-the-art encryption algorithms and network traffic routing protocols that do not leave unnecessary traces. Since the trails of the traffic routed through the network are minimal, and decryption of data is not efficient nor realistically feasible, evidence must be collected elsewhere.

The largest of the dark networks, or more formally anonymous communication network (ACN), is Tor. Tor constitutes a network of servers, some of which are web servers that comprise the so-called "dark web". More correctly, they comprise one of the dark webs; other dark networks such as Lokinet, Freenet, IPFS, and I2P also include a number of servers that comprise dark webs specific to their respective network. Tor has, however, emerged as the most commonly used and essential ACN for citizens in non-democratic or semi-non-democratic countries, as well as for whistle-blowers and journalists in need of end-to-end anonymity [97].

Nevertheless, the anonymity provided by Tor is equivocal; the well-founded privacy and encryption scheme of the Onion Routing protocol is not discriminant against its users. Whistle-blowers and criminals alike benefit from the same liberating encryption algorithms and anonymous traffic routing. The unethical use of anonymity by various cybercriminals include hosting of malicious servers, illicit and illegal content, which create an arduous digital policing arena for law enforcement. To a large extent, although not exclusively, the criminal activity using or being dependent on Tor is concentrated to Tor websites textual and graphical content such as dark marketplaces, child abuse websites, hacking web fora, and akin illicit or illegal website content.

The Tor network is designed to encrypt its traffic in different layers with different keys for each layer between each server in the network, using up-to-date standardised encryption algorithms. It consists of more than eight thousand servers, or *relays*, that encrypt and route data through the Internet cables around the world. For each connection that is made through the Tor network, a minimum of three relays is required to build a circuit for anonymous Onion Routing. The first relay encrypts the data with one key, the next encrypts it with another key, and the third encrypts it with yet another key. The result is an onion like layer structure of encrypted data and encrypted encrypted data. Anonymity is upheld by the principles of the routing protocol that requires multiple relays to create a circuit; no single relay knows the complete chain of transmission.

As the possibilities of network traffic analysis and decryption are limited on ACNs, collection of web content is a profitable and fruitful alternative technique. Manual web monitoring, web intelligence gathering, and undercover operations have proven to be successful means of identifying suspects [20].

Web crawling, i.e. automated collection of web content, is an effective technique for gathering data that is unencrypted on the Tor web that excludes a lot of manual work. The web scraping technique is widely used on the clear web for commercial and utilitarian purposes. News aggregation services, price comparison services, and digital preservation units amongst national libraries worldwide use web crawling to gather, store, and archive data that is of value to the future.

Historical snapshots, or copies, of web pages or entire websites, have been included as evidence in multiple large criminal investigations in recent years. Historical screenshots of the Silk Road 2.0 occurred in the court case against the suspected operator of the notorious dark marketplace [14]. Another example is the court complaint against the suspected operator of AlphaBay, in which screenshots were also enclosed to build the case [80]. Furthermore, the evidence presented against a suspect connected to the Swedish language dark marketplace Flugsvamp 2.0 consisted of historical copies of the website scraped by the Swedish Police [34]. A noticeable amount of servers on the Tor network are reportedly volatile and disappear from the network after some time [8]. Consequently, snapshots and historical copies of websites potentially comprise unique and essential data.

Once acquired and preserved, web content can be used as traces, clues, or evidence in investigations. Moreover, it can also be further explored and dissected by examiners, investigators, and computer programs empowered with data analysis such as statistical processing, machine learning, and artificial intelligence.

Due to the volatile nature of websites, website crawling and acquisition requires a rigid and reliable programming logic to maintain and uphold the forensic scientific principles of data integrity in order for it to be admissible in a court of law. There are a number of different web crawler software available today. Some of them are customised for forensic acquisition of web content, and others are optimised for performance and breadth and coverage of large quantities of web pages.

Current and previous research pertaining to web crawling has mainly focused on the clear web, and not anonymous communication networks.

The survey and literature review research of dark web crawlers is scarce to date. By further exploring this topic and identifying the characteristics of dark web crawlers, taking into account the aforementioned properties of anonymous communication networks and their discrepancy from the regular Internet and the clear web, a general understanding of the landscape of dark web crawlers can be presented to provide knowledge regarding practical dark web crawling construction and usage. Possibly, this knowledge will assist researchers and practitioners in using and developing tools for

crawling websites on ACNs.

In addition to the knowledge contribution, an implementation of a Tor based crawler is presented and evaluated as a use-case in the second part of this study. The design of the crawler was based on the result from the literature review and extends an already existing dark web investigation toolkit.

A. DISPOSITION

The structure and the organisation of the paper consist of seven chapters and a bibliography. This paper presents two research contributions: one systematic literature review and one experiment-based web crawler implementation. Section wise, the first chapter is the introductory chapter that briefly describes the nature of anonymous communication networks, website content and web crawling, and how it can be used in digital investigations. Chapter two extends the scientific foundation from which the research problem and the research questions spring. Chapter three contains the systematic literature review. Chapter four includes the design science-based development and evaluation of a dark web crawler, based on the results from the literature review. Chapter five presents the results from the crawler implementation in chapter four, and chapter six discusses the all of the results. The final chapter includes conclusions and suggested topics for future research based on the findings of the current paper. The last sections of the paper contain the bibliography and acknowledgements.

II. RELATED WORK

This section covers the previous research in the area of dark web crawling and cybercrime investigations to give a better understanding of the research problem and elaborate on the motivation of the subject of the current paper.

A. THE WORLD WIDE WEB

Before the world wide web was established in the early 1990s, there were other protocols that other types of "webs". Gopher was one of the preceding protocols of HTTP. Gopher was text based with a main focus on network files sharing. Due to the visual hyper text markup language (HTML) in combination with the HTTP protocol and rumours of licensing of Gopher software, the world wide web became the conquering information sharing technique in 1994-1995 [25].

HTTP utilises the Internet Protocol (IP) and the Transport Control Protocol (TCP) to transmit data such as HTML pages, images or video. The procedure of retrieving a an HTML page, or a web page, via HTTP is today the same as in the year 1990: a client (web browser) sends a GET request for a web page to a web server and gets a response in return. If the page is available the HTTP code 200 is sent, if the page was not found, a 404 code is sent. There are multiple other response codes specified in the HTTP standard [38], although 200 or not 200 is the essential response for most web crawlers.

The HTTP standard has been revised since 1989, from version 0.9, to versions 1.1, and 2.0, up to the latest version which currently is HTTP 3 [38, 39]. However, all HTTP versions support backward compatibility, which means that all requests and responses are the same as in HTTP version 1.0 [37, 38, 39].

By automating the GET requests to a website and following URLs found on it and sending GET requests for them, and storing the responses, is in simple terms what is known as web crawling. Web crawlers can be based on web browsers, or modified versions of web browsers, that automatically send HTTP GET requests. They can also be of smaller size in the form of software programmed to communicate using HTTP. As of today, there are many HTTP communication software libraries available for different programming languages, such as Haskell[1], Lisp[2], Go[3] simplifying the process of creating HTTP-based clients and servers.

B. WEBSITE ACQUISITION TOOLS

There are a number of open and closed source tools available for forensic acquisition of websites - i.e. websites that have been saved, or *scraped*, according to the principles of forensic science. A few of these website acquisition tools, albeit not all of them, support web crawling, and some support dark web crawling.

OSIRT is a web browser tailored for non-technologically savvy investigators that supports dark websites (Tor) as well as clear websites [68]. Reportedly OSIRT is widely used by law enforcement in the United Kingdom and supports video capture and video recording of website acquisition, as well as screenshots and audit log files to uphold the chain of custody in the investigation process [96].

Hunchly is a proprietary web browser add-on built for both clear- and dark (Tor) website acquisition tailored towards law enforcement [36].

FAW is a proprietary website forensic acquisition tool, shaped as a web browser, that is used by Police departments around the world to acquire web content from websites, social networks, and dark (Tor) websites. FAW also supports website crawling [58].

C. WEB CRAWLERS

The world wide web was invented in 1989-1990 [95], and one of the first scientific articles relating to web crawling was published in 1996. By using a web crawler called Inktomi, researchers could successfully collect circa 2.6 million web pages as of November 1995 [98].

The HTTP has not changed in the way web pages are requested or served, and the technique for web crawling is the same today as it was in 1995. There are ample web crawlers available today. The three most common crawlers include: Nutch[4], Crawler4j[5], and Mercator [33] according to Kumar, Bhatia, and Rattan [51].

[1] https://hackage.haskell.org/package/HTTP
[2] https://github.com/fukamachi/fast-http
[3] https://pkg.go.dev/net/http
[4] https://nutch.apache.org/
[5] https://github.com/yasserg/crawler4j

Performance wise, a research study by Yang and Thiengburanathum [101] showed that Heritrix[6] was one of the most scalable web crawlers available. In terms of robustness, the author states that the Python based crawler library Scrapy was one of the most prominent ones.

Clear web crawlers such as the above mentioned could potentially be used when a Tor socket or Tor proxy is put in front of it; thus an effective, i.e. functional, Tor crawler could be any clear web crawler used in combination with a local Tor socket connection. Although, there might be a risk of DNS request leaks and other privacy dire straits in case the configuration to the Tor network is not set up properly [93, 94].

D. THE TOR NETWORK AND THE TOR WEB

The regular web, or the clear web, uses, as mentioned, TCP and IP to transmit HTTP requests. Anonymous communication networks, or dark networks, use TCP and IP to transmit their own anonymous protocols, like Tor's the onion routing (OR) protocol, or I2P's similar garlic routing (GR) protocol. On top of the OR or the GR protocols, ACNs convey the HTTP.

The Tor network offers anonymous communication over IP using the Onion Routing (OR) protocol. While IP addresses are used to establish connection between all relays (nodes) in the Tor network, there are no IP addresses to Onion Services, previously known as Hidden Services, which comprise the "dark web" - websites running on servers on the Tor network. Onion Services are accessible only if their URL is known, for example `http://juhanurmihxlp77nkq76byazcldy2hlmovfu2epvl5ankdibsot4csyd.onion` is publicly known to be the search engine Ahmia's Onion Service [67]. There is no way of scanning a closed space of IP addresses to find an Onion Service on the Tor network, as could be done on the regular Internet, neither is it an effective approach to try to pseudo-randomly guess Onion Service URLs, although it is theoretically possible [22].

Tor websites, also known as "onionsites", do not differ from clear web pages; they look the same, they are constructed in the same way with text, images, HTML, CSS, JavaScript and they are also transferred using HTTP. The content of onionsites, however, tend to differ from the clear website due to their nature of being located on an anonymous communication network. Onionsites usually prioritise confidentiality, privacy, and anonymity over usability and performance, and therefore JavaScript is seldom implemented on these sites due to the risk of revealing the Tor user's real identity by using it [79, 61].

E. DARK WEB CRAWLERS

Aforementioned clear web crawlers such as Nutch or Mercator cannot be used without modifications to proxy a connection to the Tor network. For this reason, contributions have been made in academia to develop time efficient and powerful dark web crawlers for acquiring and analysing web content from Tor and other ACNs.

Hayes, Cappa, and Cardon [32] proposed a Tor web crawler that was capable of scraping vendor accounts from a dark marketplace and then plotting a link graph based of the vendor accounts data to investigate possible criminal activity [31].

Research has also been focused towards automating the process of cybercrime web content collection on both the clear and the dark web. Zulkarnine et al. [107] extended an already existing child exploitation identification crawler developed by Bouchard, Joffres, and Frank [9] to crawl both Tor an non-Tor websites simultaneously with the objective to identify extremist and terrorism content. In summary, the crawler managed to retrieve 260 GB of data from roughly 54.000 Tor web pages [108].

Multiple research studies have approached the same problem in a similar manner, namely by developing dark web crawlers for harvesting and computationally analysing web content, these include [77], [40], [5] [13], and [42], [87].

Despite the vast number of research articles in the area, there are remaining challenges for dark web crawlers. Dark marketplaces are often protected by CAPTACHAs and authentication mechanisms that obstructs crawlers and need to be bypassed in an effective manner. Moreover, some sites implement "crawler traps" to hinder web crawling robots from harvesting the content from the server, such as infinite loops of web pages that do not exist, or automatically pesudo-randomised pages and links that the crawler endlessly follows and downloads, exhausting it with nonesense [18] [19].

F. RESEARCH MOTIVATION

To date, a thorough literature review of dark web crawlers, like there are for clear web crawlers, is missing. Anonymous communication networks are designed and operate in a different manner compared to the clear web and the regular Internet. Therefore, crawlers need to be programmed and configured accordingly to successfully complete their crawling tasks. As pointed out in previous sections, there are dark web crawlers available that have been developed by both private and public actors, however, no scientific study has systematically reviewed them. One of the objectives of this research was thus to present a rigorous assessment of existing dark web crawlers developed or used in scientific literature. The second objective was to implement the dark web crawler most frequently used in academic research to fit it into an existing toolset lacking a comprehensive and verified crawler and evaluate its performance.

III. SYSTEMATIC LITERATURE REVIEW

In the first of the two segments of this article, a systematic literature review (SLR) regarding dark web crawlers is presented. The SLR was conducted using the commonly applied guidelines by Kitchenham [47], which include three major phases: (1) planning, (2) conducting, and (3) reporting the literature review. The phases (1) planning and (3) reporting

[6]https://github.com/internetarchive/heritrix3

are to a large extent implicit in the article itself, however they will be further clarified in this section.

The first step, (1) planning, includes the preparation of the SLR: sketching out the background of the research, the research question, study selection and study quality assessment criteria, as well as data extraction and dissemination strategy.

Phase two (2), conducting a literature review, on the other hand is more extensive and presented in further detail as follows. Phase two includes the activities: (1) study selection, (2) study quality assessment, (3) data extraction, (4) data synthesis. Each activity is presented in the upcoming four sections.

The third phase, (3) reporting, includes specifying the dissemination mechanism as well as the formatting and evaluating the report. These were considered to be implicit in the nature of this report as a peer reviewed research article that is evaluated and publicly disseminated.

A. RESEARCH QUESTIONS

All of these activities were carried out as instructed, and the remaining concrete outcome was the research questions that were specific to the SLR (i.e. this segment of the article), which is not equal to the research questions of the entire article:

1) Which crawlers and/or scrapers have been used in scientific literature to collect data from the Tor network?
2) How do crawlers and/or scrapers used to collect data from the Tor network route the traffic?
3) Which programming languages and libraries have been the most common for programming crawlers and/or scrapers on the Tor network?

B. SEARCH STRATEGY

The documents to be selected for review are referred to as "studies" by Kitchenham [47] In this work, the research database that was utilised for retrieving studies was Scopus[7]. The reason for choosing only Scopus as the main resource was its complete coverage of scientific research articles in combination with its powerful API for search and fetching articles and their META data.

Scopus includes articles dating back to the late 18th centry and indexes from the databases ACM digital library, ScienceDirect, SpringerLink, and IEEEXplore [88]. Due to the young age of the Tor network and the research of the "dark web", there was no risk of missing any historical articles that have yet not been added to the database. The search did therefore not have any publication date preference.

A Python script was developed to fetch articles from Scopus. The script can be found on https://gitea.dsv.su.se/jebe8883/SLR

C. STUDY SELECTION STRATEGY

A total number of 59 articles were retrieved from the database using the keywords `TITLE-ABS-KEY ((dark AND web AND crawler) OR (dark AND web AND scraper) OR (tor AND crawler) OR (tor AND scraper)) AND LANGUAGE(english)` where TITLE-ABS-KEY, i.e. title, abstract, and keywords define the META data in which the search for the specified terms were done. "LANG" specifies the language English; non-English articles were excluded from the search results.

Once the articles have been found and fetched by the script, they were written to a file on disk together their META data, such as title, abstract, keywords, authors, and DOI, as can be seen in the example below. This made the processing of data more manageable than working with the web based service. In addition, the search and selection process remains more transparent when publishing the source code of the script that performed it.

```
Title: Implementing UTM based on PfSense
platform
Abstract: Today, as Network environments
become more complex and cyber and Network
threats increase...
Authors: Asghari V.
Publication: Conference Proceedings of...
Publication Type: Conference Proceeding
Article Type: Conference Paper
Scopus ID: SCOPUS_ID:84971439968
DOI: 10.1109/KBEI.2015.7436210
URL: https://api.elsevier.com/content/
abstract/scopus_id/84971439968
Keywords: Keyword1, keyword2
Time: 2022-06-01 14:22:13.125470
```

1) Inclusion and Exclusion Criteria

According to Kitchenham [46], different criteria for including certain articles and excluding others are crucial for initially identifying studies that relate to the research question. Naturally, a number of articles were excluded already in the database search which only included English language articles relating to the search terms specified in the previous section. In this section further inclusion and exclusion of article is explained.

In this systematic literature review, the focus was on content crawling and scraping on the Tor network, although articles that do not explicitly concern Tor were decided to be included, given that they relate to, or mention, the impact the article might have on the Tor network. The reason for this inclusion scheme was limiting the risk of missing relevant or semi-relevant articles from the selection process.

- Inclusion criteria:
  - Articles that concerned crawling, scraping, intelligence gathering, or monitoring of servers on the Tor network.
  - Articles in which a crawler or scraper was used to retrieve data from the Tor network.
- Exclusion criteria:

[7]https://scopus.com

- Articles that did not mention the Tor network in regards to crawling or scraping.
- Articles that did not concern any sort of content retrieval from remote servers (on the Tor network).
- Articles that were not peer reviewed research articles, i.e. journal articles, conference proceedings, workshop proceedings.

The low number of articles from the search query enabled for a manual assessment to be made. As a first inclusion or exclusion assessment, the abstracts for all 59 articles were manually inspected and excluded or included based on the previously specified criteria. By inspecting the titles of all articles, it appeared that [21] and [19] had the exact same title. The former was a conference proceeding article comprised of nine pages and the latter was a journal article of fourteen pages. The conference article was filtered out since it was considered to be a briefer version of the journal article. The same action was taken for the conference and journal articles [40] respectively [41], where the latter journal article which was favoured. Similarly, there was a similar pair of the conference article [75] with the same title and DOI as the journal article [76]. The conference article was excluded in favour of the journal article.

The total number of articles that remained after removing conference and journal duplicates was 56.

D. STUDY QUALITY ASSESSMENT

After the initial sieving process, the guidelines by Kitchenham [46] suggest a quality assessment is done to filter out any possibly indecent studies based on a set of quality assurance check points.

The remaining 56 articles were quality-checked for: bias, inconsistencies, and validity, in addition to the established inclusion and exclusion criteria.

1) Excluded Articles

Out of the 56 remaining articles, 15 articles were excluded based on the exclusion criteria after reading their abstracts. These excluded articles are recorded in Table 1.

2) Included Articles

After subtracting the excluded articles, 41 articles remained for the quality assessment. The included articles and summary of them, as well as an external link to their source code of software repository can be found in Table 2.

E. DATA EXTRACTION STRATEGY

The 41 documents that remained after the quality assessment were extracted for further analysis and assessment. In the META data from each of the articles fetched from Scopus, there was a link to the full article. These articles were manually downloaded and then data were extracted from them.

In Table 2, the included and relevant articles are presented. The ACN based web crawler or scraper used in each of the articles is presented in the table. This table also includes a link to the source code of the crawler/scraper used, given that it is open source software and that the code is publicly available.

However, during the data extraction phase, there were seven articles that were not relevant that were included; these were articles that concerned crawling, but not on the dark web as in [55] and [4], or using a different definition of the dark web that does not mean anonymous communication network, such as I2P, Freenet, Lokinet, as in [102], [53] and [29]. Additionally, studies that used the Tor network and crawlers separately, as in [73] were excluded. A summary of the articles excluded during the data extraction can be found in Table 3.

F. DATA SYNTHESIS STRATEGY

Synthesising the data collected and analysed in the SLR is the final step in the process, according to Kitchenham and Charters [48]. The data synthesis strategy applied in this SLR was a descriptive content focused one, where the interest lies in web crawlers used in the selected studies.

35 relevant studies were collated and summarised in the synthesis stage. It can be concluded that a minority of them used or promoted the source code of the crawlers publicly; six out of 35 were open source crawlers.

The most common way of collecting dark web content in the scrutinised articles was to build a custom crawler specifically for that purpose. The most common programming language used was Python, and the most common crawler libraries used were Selenium and Scrapy. The pre-existing crawler that was used in more than one study was Apache Nutch, which was used in two studies: [40] and [43]. It should be noted that in both aforementioned studies, Nutch was customised to fit the study design and not used out-of-the-box. In addition, the two studies using Nutch as a crawler were written by the same five authors, with two more authors in [43] than in [40].

- The most commonly used crawler was Apache Nutch. It was used in two out of 34 relevant studies.
- The most common programming language was Python, which was used in 16 out of 34 articles, see pie chart in Figure 1.
- The most common crawler library mentioned in the selected articles was Selenium. Used in six out of 34 articles.
- The next most common crawler library mentioned was Scrapy, which was used in four out of 34 articles.

IV. IMPLEMENTATION OF A TOR CRAWLER

The second part of this article is complementing the systematic literature review with an implementation based on the results from the systematic literature review in the previous section.

In a previous research article, a toolset called D3, developed for annotating, highlighting, collecting, and analysing .onion sites was presented [7]. However, the web crawler in that toolset was a primitive one that only served as a proof-of-concept component. To extend that toolset, an adequate dark

Title	Reference	Reason for Exclusion
InTIME: A machine learning-based framework for gathering and leveraging web data to cyber-threat intelligence	[50]	No crawler used.
Link Harvesting on the Dark Web	[16]	No crawler used.
A Survey of the Dark Web and Dark Market Research	[105]	No specific crawler.
Exploring hackers assets: Topics of interest as indicators of compromise	[82]	No crawler used.
Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence	[66]	No relevant crawler.
A credit card fake detection system using image cryptography	[85]	Irrelevant; different topic.
DSC 2018 - 2018 IEEE Conference on Dependable and Secure Computing	[64]	Only abstract; no article
12th International Conference on Security, Privacy, and Anonymity in Computation, Communication, and Storage, SpaCCS 2019	[62]	Only abstract; no article
AHFE International Conference on Human Factors in Cybersecurity, 2018	[63]	Only abstract; no article
Analyzing the DarkNetMarkets subreddit for evolutions of tools and trends using LDA topic modeling	[76]	No crawler used.
Fingerprinting web browser for tracing anonymous web attackers	[56]	No crawler used.
Halo shapes, initial shear field, and cosmic web	[86]	Irrelevant; different topic.
A web-enabled software for real-time biogas fermentation monitoring - Assessment of dark fermentations for correlations between medium conductivity and biohydrogen evolution	[44]	Irrelevant; different topic.
Onion routing circuit construction via latency graphs	[11]	No crawler used.
Financial Cryptography and Data Security - FC 2011 Workshops, RLCPS and WECSR 2011, Revised Selected Papers	[65]	Only abstract; no article

Table 1. Articles excluded from the initial exclusion phase in the systematic literature review.

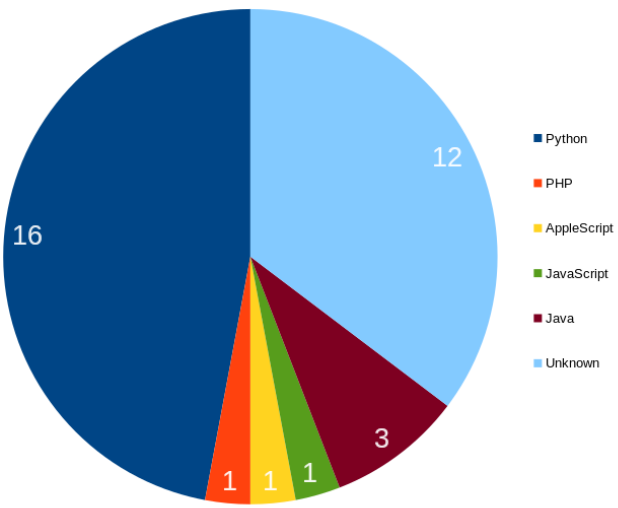


Figure 1. Programming languages used in articles found in the systematic literature review.

web crawler needed to be integrated to it. In this chapter, the development, integration, and testing of such crawler is presented.

The methodology for producing an artefact that will solve the problem of non-efficient non-uniform and non-flexible crawling of cybercrime .onion sites, was experiments driven design science research method (DSRM), as proposed by [72]. The DSRM consists of seven activities invented to guide its applier through the process of creating an artefact in a rigid, yet flexible, and scientific manner. The five activities are: (1) Problem Explication, (2) Requirements Definition, (3) Design and Development, (4) Artefact Demonstration, (5) Artefact Evaluation [72, p. 77].

In this research study, the problem explication was given in chapter 1. This section encloses elaborations on activities two to five. The artefact in this research was a computer program; or an *instantiation* as it is formally called, according to Perjons and Johannesson [72, p. 29].

A. REQUIREMENTS DEFINITION

To succeed with the research objectives and integrate a crawler to the already existing toolset "D3", a set of requirements were elicited to outline the design of the crawler. The requirements were created by the authors and supported by previous research, as seen in detail in Table 4.

B. DESIGN AND DEVELOPMENT OF ARTEFACT

It was concluded in the systematic literature review in the previous segment that the most commonly used means of crawling Tor websites was using writing a program in Python with help of the library Selenium, therefore those components were used to develop the complementary D3 crawler: the Digital Detectives Comprehensive Tor Toolset (DIDECT2S) crawler.

Selenium originally is a web browser automation testing and debugging tool, therefore it typically launches a web browser to complete its tasks. This makes a Selenium-based crawler more powerful, since it mimics human behaviour and allows human keyboard and mouse interaction. On the other hand, it also means the crawling process is slower and more computationally exhausting. Nevertheless, Selenium was chosen as the library for the crawler to use due to its flexibility and human interaction capabilities that suffice

Article	Crawler Used	Source Code	Summary
[106]	Self-developed	N/A	A framework, "DW-GAN", for breaking CAPTCHAs, a crawler is used. Programming language not specified.
[84]	Self-developed	N/A	A crawler used to find any differences before vs. after the Covid-19 pandemic on the dark web. Programming language: Python.
[19]	Self-developed	N/A	Detection of crawler traps using crawling and distance measures. Programming language not specified.
[17]	SpyDark	N/A	A Tor crawler that feeds the web content to an NLP model to classify it. Programming language not specified.
[60]	Black Widow	N/A	An efficient Scrapy based Tor crawler using Apache Solr and MongoDB. Program language: Python.
[27]	Self-developed	N/A	A Tor crawler collecting threat intelligence data from three different dark marketplaces. Programming language not specified.
[100]	Self-developed	N/A	Scrapy based Tor crawler for monitoring dark websites and analysing them based on a TF/IDF calculations. Program language: Python.
[15]	TorBot	https://github.com/DedSecInside/TorBot	Comparing the "Hidden Wiki" by crawling it in 2020 and 2021. Program language: Python.
[54]	Self-developed	N/A	Crawled dark websites to classify web pages using ML algorithms. Program language: Python.
[91]	CrawlBot	N/A	A self-developed crawler for identifying child abuse material on the dark web using AI. Program language: Python.
[3]	Self-developed	N/A	Used a crawler to make graphs and graph calculations on Tor websites. Program language: Python.
[6]	Self-developed	N/A	Used a crawler to find Tor sites that were identical. Program language: Python.
[104]	Self-developed	N/A	Used a Selenium based crawler to classify Tor websites. Program language: Python.
[24]	Self-developed	N/A	A crawler was used to retrieve Tor web data and classify it after reducing its feature space. Program language: Python.
[52]	Self-developed	N/A	A Selenium based crawler was used to harvest data specific to South Korean dark websites. Programming language not specified.
[103]	Unclear	N/A	A Hadoop based framework for collecting and analysing Tor web content. Programming language not specified.
[90]	Self-developed	N/A	A Selenium based crawler to identify IoT attack trends. Program language: Python.
[49]	ACHE	https://github.com/ViDA-NYU/ache	Clear and dark web crawler used to identify IoT threat intelligence. Programming language: Java.
[57]	(1) Self-developed (2) Branwen et al. [10]	N/A	Crawled and analysed data to investigate relation between drug overdoses and drug ads using Scrapy.Program language: Python.
[70]	Self-developed	N/A	A Selenium based crawler used to identify characteristics of Tor websites over time. Program language: Python.
[89]	The Dark Crawler	N/A	A crawler used for sentiment analysis of data for identifying extremist content. Programming language not specified.
[99]	Self-developed	Upon request	Crawling and scraping dark marketplaces with self-developed Selenium based crawlers. Program language: Python.
[69]	Self-developed	N/A	A conceptual system for crawling suspicious and malicious .onion sites. Programming language not specified.
[32]	Self-developed	N/A	A crawler developed for investigating dark marketplaces. Program language: AppleScript.
[41]	Self-developed (based on Nutch)	N/A	A clear and dark web crawler framework for classifying content using ML algorithms. Java.
[81]	(1) [35] (2) [59]	N/A	A crawler used to classify content based on fuzzy kNN. Program language: PHP and Python.
[12]	(1) bUbiNG (2) Self-developed	https://github.com/LAW-Unimi/BUbiNG	A suite for Tor crawling and text mining, customised using Scrapy spiders. Program language: Python.
[92]	Self-developed	N/A	By using a Selenium based crawler through Tor, discrimination of Tor exit nodes is examined. Programming language not specified.
[28]	OnionCrawler	N/A	Automatic thematic labelling of Tor web content crawled using self-developed OnionCrawler. Programming language not specified.
[45]	Self-developed	N/A	Analysis of illicit products using AI algorithms based on crawled Tor web content using Nightmare.js. Programing language: JavaScript
[107]	Dark Crawler	N/A	Analysis of extremist content fetched using a Tor crawler based on [9]. Programming language not specified.
[2]	DATACRYPTO	N/A	Analysis of monthly revenue per drug on Silk Road 1.0. Used Self-developed crawler. Programming language not specified.
[43]	Nutch	https://github.com/apache/nutch	A customised Nutch crawler automatically classifying explosives content from Tor. Programming language: Java.
[26]	Self-developed	N/A	Crawling Tor web to identify extremist content. Programming language not specified.

Table 2. Summary of the 34 articles included in the systematic literature review. N/A means source code was not available.

Title	Reference	Reason for Exclusion
Big Data, Method and the Ethics of Location: A Case Study of a Hookup App for Men Who Have Sex with Men	[55]	No ACN, like I2P, Freenet, Lokinet, or I2P.
Inference in OSNs via lightweight partial crawls	[4]	No crawler.
Discovering Topics from Dark Websites	[102]	No ACN, like I2P, Freenet, Lokinet, or I2P.
Dark Web—Onion Hidden Service Discovery and Crawling for Profiling Morphing, Unstructured Crime and Vulnerabilities Prediction	[83]	Theoretical work; no crawler used.
The investigation of the possibility of automated collection of information in the hidden segment of the Internet	[53]	No ACN, like I2P, Freenet, Lokinet, or I2P.
Discovering abnormal behaviors via HTTP header fields measurement	[29]	No I2P, Freenet, Lokinet, or I2P.
Understanding website behavior based on user agent	[73]	Tor network and crawlers addressed separately.

Table 3. Articles excluded in the data extraction activity. Reason for exclusion in the right-hand column.

ID	Requirement	Comments
RQ1	Support for crawling .onion services.	Authors themselves, [74]
RQ2	Option to scraping JavaScript	Most onion sites do not allow JS [18], although it should be optional.
RQ3	Support for graphic content retrieval	Reference: Graphic content is crucial in CSAM investigations [23]
RQ4	Support credentials authentication (e.g. cookies, username/password).	Reference: [74]
RQ5	Support for solving CAPTCHA token and authentication gates.	Reference: [74]
RQ6	Support for automatic screenshots for completeness and forensic soundness.	Reference: Authors themselves.
RQ7	Support for parallel processing.	To reduce download time. Reference: [74]
RQ8	Support extensive logging to maintain the chain of custody.	Authors themselves.
RQ9	Support for local or in-house hosting; non-cloud based.	Reference: Authors themselves.
RQ10	Be of open source code.	Reference: Authors themselves.
RQ11	Support delay in crawling	To avoid crawler traps [18].

Table 4. Requirements elicited by the author for the building a dark web crawler to be integrated with the already existing toolset.

the needs of more law enforcement tasks that rarely need extensive crawls of massive amounts of .onion URLs. Since a Tor crawler requires a connection to the Tor network and to adhere to the Onion Routing protocol, it was not possible to use Selenium out of the box. Therefore, a custom-made Selenium driver, version 0.6.2, for the Tor browser[1] was used.

To enable comparison between clear web and dark web crawling and in favour of evaluating the crawler, a clear web crawler not using a Tor connection and the Tor browser was created as well. The clear web uses the same code, except the web driver and network connection. For crawling clear web pages a Selenium driver[8] version 4.4.0, using Gecko driver version 0.31.0 for Linux 64-bit[9], and a non-modified Internet connection from Stockholm University was used. The Selenium web driver headers were not modified for the clear web, nor the dark web, crawler. In simple terms, the logic of the crawler was the following:

1) Initiate log file
2) Go to URL home page
3) Get robots.txt (if relevant)
4) Take screenshot
5) Save home page source
6) Find link elements on home page
7) For each link that is not external domain or disallowed by robots.txt
    a) Request link web page
    b) Save page source
    c) Save images on page
    d) Take screenshot
8) Close all file handlers and log files and quit program

It should be noted that this crawler was designed for a laboratory environment and thus it was configured to obey robots.txt and not crawl external URLs in order to avoid fetching unwanted content. Furthermore, the crawler was designed to be a general crawler, rather than a focused crawler. Therefore it might not work as intended on all websites, depending on how the website in question is constructed.

C. LIMITATIONS OF THE ARTEFACT

The developed crawler had a few limitations compared to other crawlers. Firstly, it was built for digital forensic purposes with forensic soundness and correctness primarily in mind.

Thus, the performance speed was not a priority. The Selenium-based web driver used as the engine for the developed crawlers limited the execution speed due to the fact thar it operatyes an actual browser the requires multiple system

[8]https://pypi.org/project/selenium/

[9]https://github.com/mozilla/geckodriver/releases/download/v0.31.0/geckodriver-v0.31.0-linux64.tar.gz

libraries and components to launch in order to function. In addition, crawling the Tor network is intrinsically slower than the regular Internet due to the onion routing, which limits the execution speed of Tor crawlers in general.

The crawler developed as part of this research was a focused crawler and hence not fully comparable with other clear web/dark web generic crawlers.

D. DEMONSTRATION OF THE ARTEFACT

In this activity of the design science research method, the artefact is demonstrated. In this case, a web crawler was built to extend an already existing toolset. In Figure 2, the updated toolset called DIDECT2S in presented. The crawler component i highlighted in red.

The logic of the program was briefly explained in the previous section, however, the complete source code is available on: https://gitea.dsv.su.se/jebe8883/DIDECT2S.

E. EVALUATION OF THE ARTEFACT

The artefact designed and developed based on the outcomes of an exhaustive systematic literature review was a comprehensive Tor web crawler that was integrated in an already existing toolset of dark web cybercrime investigative tools. To concretely exhibit how well the developed artefact fulfilled the requirements specified for it, this section presents a requirement and artefact evaluation, as the final activity in the design science research method.

To evaluate the artefact, confirm its requirement fulfilment, and assess its overall usefulness and effectiveness, a couple of experiments were conducted. The experiments were done in a laboratory environment setting using realistic case scenarios and authentic websites as experiment objects.

The experiment was divided into two parts: the first was to crawl both clear and dark websites, and the second was to crawl only a dark marketplace, protected by authentication and CAPTCHA on the Tor network. Both experiments were designed to verify that the underlying requirements of the artefact were met and fulfilled.

Non-functional requirements such as RQ11 - that the software should be open source, RQ1 - that the crawler should support crawling .onion addresses, and RQ9 - that it should be possible to self-host the crawler were implicitly fulfilled as the crawler was built and the source code was published. Similarly, RQ7 - support for parallel processing, was considered fulfilled since the the Selenium web driver, which was used to build the crawler, can be executed in parallel according to its official documentation [78]. All other functional requirements were affirmed in the experiment, as depicted in Figure 3.

The first part of the experiment was to crawl the same set of websites located on both the regular Internet (clear web) and the Tor network (dark web) to verify that the Tor connection and crawling mechanisms work as well as the clear web ditto. The web pages crawled on the regular Internet should ideally be the same as the web pages crawled on the Tor network if the crawler works as intended. Of course, some content might be updated or changed between the crawls, but to a large extent the saved web pages should be the same since the website is the same.

Websites are hosted on web servers on both the regular Internet, as well as on the Tor network. On Tor, web servers are called "Onion Services" and the websites are referred to as "onionsites". The terms used in this article will be "clear websites" and "clear web" and "dark websites" and "dark web" respectively. The websites used in the evaluation experiment of the crawler were the following:

1) **Debian**
   - https://debian.org/
   - http://5ekxbftvqg26oir5wle3p27ax3wksbxcecnm6oemju7bjra2pn26s3qd.onion
2) **The Guardian**
   - https://theguardian.com/
   - https://www.guardian2zotagl6tmjucg3lrhxdk4dw3lhbqnkvvkywawy3oqfoprid.onion
3) **New York Times**
   - https://www.nytimes.com/
   - https://www.nytimesn7cgmftshazwhfgzm37qxb44r64ytbb2dj3x62d2lljsciiyd.onion
4) **Qube-OS**
   - https://www.qubes-os.org/
   - http://qubesosfasa4zl44o4tws22di6kepyzfeqv3tg4e3ztknltfxqrymdad.onion
5) **CIA**
   - https://cia.gov/
   - http://ciadotgov4sjwlzihbbgxnqg3xiyrg7so2r2o3lt5wz5ypk4sxyjstad.onion

The second part of the experiment was to further assess the crawler's capabilities. The crawler was set to crawl a dark marketplace protected by username and password authentication as well as a CAPTCHA token. This part of the experiment served the purpose of verifying that the crawler could solve a typical dark web investigation task of saving clues and evidence from for example a dark marketplaces or child abuse websites behind an authentication portal.

Since, there was no existing comparison data set for the scraped web content in the experiment, a manual verification of the data downloaded was done. The dark marketplace that was chosen for this crawling task was White House Market (WHM) http://hvilngbbx2yxtq7ilsrjsosv374phq4jx2nq5izot5baxlqyjy3u2cid.onion. The White House Market has been available on the Tor network since 2019, and has over 3000 vendors, according to the website.

WHM was considered a typical target for cybercrime investigation since it required authentication and CAPCTCHA solving, and the content of interest hosted on it was both text and images. In short, it was deemed a representative website to scrape for examining the fitness and performance of the developed crawler. An account was created to login on to and scrape the WHM homepage.

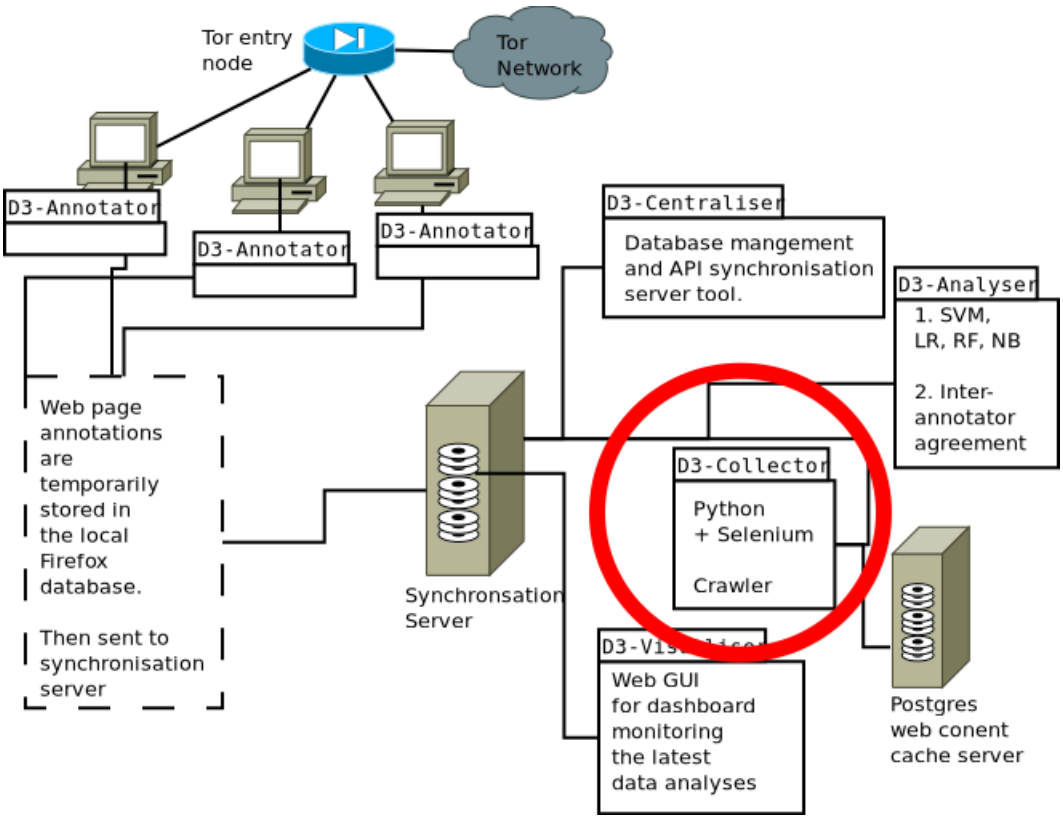


Figure 2. Topology of the updated D3 toolset - the Digital Detective's Comprehensive Tor Toolset (DIDEC2TS) with the new crawler component highlighted in red.

As a first step, a manual inspection of the website was done to assess its structure and content. It was concluded that there was no sensitive image material nor any private information such as email addresses or personal IDs that, according to ethical research codex, should not be downloaded, analysed, or processes without consideration. The WHM pages subject to scraping merely contained usernames, user avatars, product descriptions, prices, and other (mostly illegal) products META data.

The crawler built was instructed for all tasks to fetch all pages from the starting page, but not more than that. This limitation was set not to overload the web server or avoid being blocked from the website for the purpose of the experiment. In addition, a delay between each request was configured not to overload the servers more than necessary and avoid being blocked by the web server. Python's random library[10] was used to pseudo-randomly generate a delay of zero to four seconds between each request. Both the clear web and the dark web crawler were configured to, by default, collect and download all links and images on the page it was is given when started.

1) Evaluation of Crawling Clear- and dark Website Pairs

To make the clear web and dark web crawls as similar as possible in the first part of the experiment, the browsers were configured with the same settings and add-ons to avoid any discrepancies in the results due to misconfiguration. In an authentic cybercrime investigation, the investigators would not respect the robots.txt file when crawling, however, in this experiment the robots.txt are respected and none of the disallowed entries were scraped by the crawler for ethical reasons.

To evaluate how well the crawler performed the task of fetching web pages from the same websites on both the clear and the dark web, each pair of pages were compared by measuring their degree of similarity. Each pair consisted of a web page from the crawled clear website and the same web page crawled from the dark website, e.g. https://guardian.com /index.html and https://guardian.onion/index.html comprise a pair in this step of the evaluation.

The comparison of retrieved web pages was both manual and computational. Firstly, the directory contents were manually inspected to find out which files had been downloaded from each crawler. Secondly, the file content differences were manually inspected by the researchers using GNU command line program Diffutils[11]. To count the number of files that differ, GNU Wc word counter [12] was used. For explicitly extracting the files that differ, and not including the ones that

[10]https://docs.python.org/3/library/random.html

[11]https://www.gnu.org/software/diffutils/

[12]https://www.gnu.org/software/coreutils/manual/html_node/wc-invocation.html

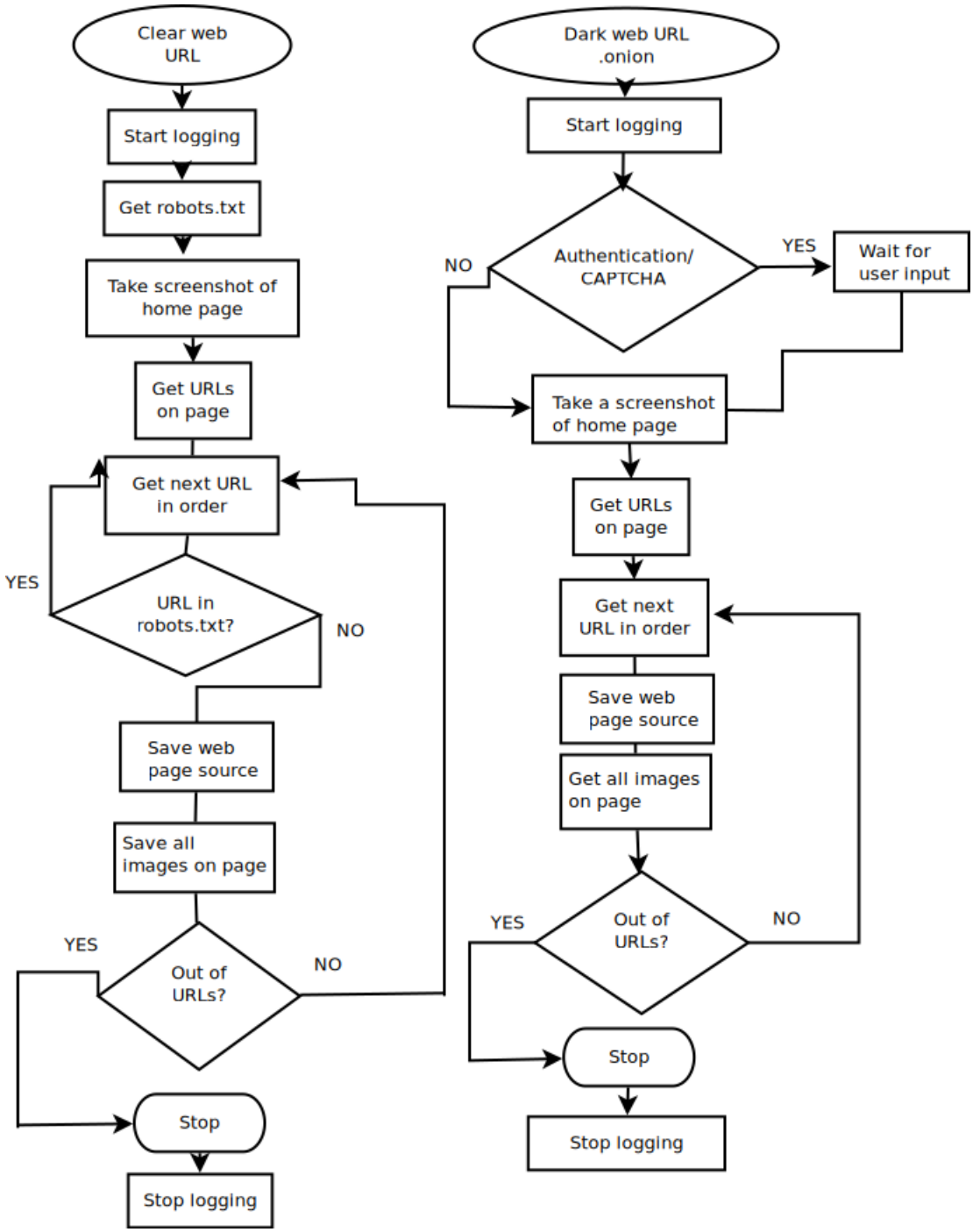


Figure 3. Flowchart depicting the two experiment scenarios in the artefact evaluation, the clear website crawling on the left hand side and the dark website crawling on the right hand side.

are the same, GNU Grep [13] was used in combination with Diff and Wc. The grep keyword was changed to "same" in order to find the files that Diff reported as identical. This results in a binary classification of files that differ and files that do not. In the next step, the similarity score will more reveal to what degree the files differ. The full command reads:

```
$ diff -q -s -N website-CW/webpage.html
website-DW/webpage.html |  grep "diff"
| wc -l
```

The computational comparison was done using a Python script that converted the websites' textual content to a vector of characters and then calculated the similarity measures between the vectors. In information retrieval, different similarity measuring algorithms are used to compare a search query string with a retrieved document. The most common algorithms used include: Cosine, Euclidean, Jaccard, and Okapi [30].

Similarity measure algorithms are often used on written "human language" where semantics are crucial. However, in this experiment, the text documents to be compared consisted of "machine language" - HTML, JavaScript, and CSS, therefore they documents were converted into character based vectors instead of word-based vectors, which would not take into account machine language characters.

The cosine similarity score is calculated from the cosine angle between two documents represented as vectors, e.g. A and B, with their inner product divided by the vector product of A and B. Formally expressed as:

$$\cos\theta = \frac{A \cdot B}{|\vec{A}||\vec{B}|}$$

Each web page was converted into a vector of characters and frequency using Scikit's CountVectorizer[14]. As a second step in the process, the cosine similarity[15] and Jaccard similarity scores[16] were calculated for each clear web page and the dark web ditto. The clear web and dark web pages that had the same title and hence filename were considered a "pair". Orphan, i.e. single, web pages with no clear- or dark web "partner" were excluded from the similarity calculation. An example of a clear web- dark web page pair is the support.html page of Debian's websites (curly brackets are only a pair indicator):

```
{https://debian.org/support.html,
http://5ekxbftvqg26oir5wle3p27ax3wksbxcec
nm6oemju7bjra2pn26s3qd.onion/support.html}
```

As a final step in the clear and dark website comparison, the images saved by the crawler from the website pairs were compared with each, comprising "image pairs" in similar way to the web pages. The purpose was to confirm whether they were exactly the same or not. Textual web page content, such as HTML and JavaScript is more volatile than pictures since it might change for different locations, browser agents, IP addresses, or the current time. Visual web page content in form of images does not change as dynamically. For this reason, a hash sum comparison between images retrieved from the clear- and dark websites was deemed adequate to estimate any differences in content retrieval discrepancy.

The comparison of clear web and dark web images was done using by calculating the SHA1 hashsum for each respective image in the pair. Practically this was done with the with Sha1deep[17] as follows:

```
$ sha1deep -m website-CW/*.jpg >
cw_hash_sums.txt
$ sha1deep -m cw_hash_sums.txt
website_DW/*.jpg
```

2) Evaluation of Crawling a Dark Marketplace

As a second part of the artefact evaluation, the crawler was evaluated on a dark marketplace crawling task. Since there was ground truth data set to compare the crawled dark marketplace data with, it was manually verified that the crawler had scraped all links available; there were too many images to manually verify that they were correctly fetched by the crawler, therefore only images from the starting page were verified.

The crawler was set to crawl the White House Market and scrape all links and download all link pages, as well as images on the starting page. Since WHM does not allow JavaScript to be enabled, for security reasons, it was disabled in the Selenium Tor Browser. After the scraping was done, the White House Market's homepage source code was manually inspected to verify that all links were correctly fetched by the crawler.

V. RESULTS

The results from the first segment of this research study, the systematic literature review, were presented in previous section. In this section, the results from the implementation and evaluation of the developed clear web and dark web crawler is presented.

A. CLEAR WEB AND DARK WEB CRAWLING RESULTS

The crawler was implemented for scraping both clear web and dark websites, the data collected from each web type was compared using couple of different techniques and measures.

First, the semi-manual inspection of the website pairs done was using GNU Diffutils. Diffutils identified discrepancies between the scraped web content files. Table 5 shows the number of pages that were downloaded from the clear web (CW) and the dark web (DW) versions of the websites in question. The web pages were saved as local files by the crawler, and the third column presents how many of the files that had identical content. The fourth column presents how

[13] https://www.gnu.org/software/grep/manual/html_node/index.html

[14] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[15] https://numpy.org/doc/stable/reference/generated/numpy.cos.html

[16] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.jaccard_score.html

[17] http://md5deep.sourceforge.net/

many of the file names, in this study equivalent to the titles of the web pages when downloaded, that were identical. The duration of the crawling process is found in column five in Table 5.

The crawler fetched the same number of pages from both the clear web- and the dark websites for Debian, QubeOS, and CIA. In the case of CIA's website, however, the index.html was downloaded twice from the clear web crawler. This was due to a programmatic error related to internal URLs in the clear web in the crawler where both the index referrers "/" and "https://cia.gov/" were downloaded downloaded.

The Guardian's website, there were 12 files that were not retrieved from its onionsite. The random wait was set to 0-4 seconds to avoid blocking and therefore the complete scraping of the 201 web pages took circa 26 minutes. The scraping of the clear web version of The Guardian took circa six minutes with the same random delay of 0-4 seconds between each HTTP request. The files that were missing from the scraping of The Guardian's Onion were URLs that were not available over their Onion site, see example in Figure 4. In total there were nine such web pages that were not retrievable from their dark website compared to the clear website. The web pages that did match and comprise a clear and dark web page pair had a cosine similarity score between 0.9324 and 0.9999 as can be seen in Table 6.

The New York Times' Onion website blocked the dark web crawler from collecting certain pages, as can be seen in see Figure 5. Due to this data collection disruption, there were pages collected from the clear website but not the dark website. According to the message displayed, the crawler was IP address blocked and required a CAPTCHA to be solved in order to continue. The reason for blocking the crawler, was most likely the fact that the IP address was shared with other Tor users, as opposed to the clear web crawler which used Stockholm University's IP address range.

The similarity scores were calculated for each clear web - dark web page pair. Due to the vast number of web page pairs, only the highest, lowest, the mean, and the median of the similarity scores for each websites are presented in Table 6. Note that the cosine similarity score is abbreviated as "CS". The exact scores for each pair can be found online[18].

In addition to web pages, images were downloaded by the crawler as well. Since there were a discrepancy between the number of downloaded web pages from the clear- and dark web respectively, there was naturally a discrepancy between the number of images downloaded from each website. The number of images retrieved from each respective website is presented in Table 7.

B. DARK MARKETPLACE CRAWLING RESULTS

The DIDECT2S dark web crawler was used to crawl a dark marketplace in order to demonstrate and validate that fits its purpose as a digital investigation tool.

[18]https://gitea.dsv.su.se/jebe8883/SLR/

The crawler was tasked to scrape all images, links, and linked pages from the home page of a the dark marketplace White House Market Onion site. At the time of scraping, September 2022, the manual observation found 251 links on the homepage of the dark marketplace in question. The crawler managed to scrape the pages for all links identified on the home page, as well as the images on those pages. Detailed can be found in table 8.

In total, 250 pages including 2881 images were fetched in 20 minutes and 47 seconds including a request delay of zero to four seconds. On average the crawler scraped 12 web pages including source code and images, per minute. The image sizes varied between 5758 bytes and 663 bytes.

VI. DISCUSSION

This research article was divided into two segments: one theoretical systematic literature review, and one practical design science implementation based on the theoretical findings. The results from the systematic literature review in segment one concluded that the programming language Python, in combination with the web debugging and scraping library Selenium was the most used combination for developing dark web crawlers in academic studies. Consequently, the experiment results from the developed Tor web crawler in segment two demonstrated that it was a compelling duo.

The systematic literature review results show that the authors build their own crawler in most scientific articles concerning dark web crawlers. In addition, few crawlers were publicly released as open source code: only four out of 34. The absence of open source crawlers conforms with the results from previous research by Kumar, Bhatia, and Rattan [51], namely that few researchers use open source crawlers and that few researchers mention which open source crawler was used in their study.

The practical implementation based on the theoretical findings was evaluated in a controlled experiment which cannot be generalised and is not applicable to the real, constantly changing world. However, the results indicate that the crawler developed and presented is usable, effective, and efficient under certain circumstances.

The experiments were successfully completed, and the functional requirements were consequently fulfilled. The results from the experiments also showed that the crawler was capable of crawling both clear web and dark websites. Due to the fact that two different Selenium drivers were used for each type of web, the experiment evaluation served the purpose of verifying that both of them work correctly; i.e. that web pages and images are fetched in their entirety from websites requested.

When scraping websites available on both the Tor network, the dark web, and the regular Internet, many of the same pages and the same number of pages, were fetched. In the case of Debian's dark and clear websites, 45 out of 50 the web pages were identical; 61 out of 64 from Qubes-OS websites and 204 of the web pages fetched from the Guardian's websites, had the same URLs, except for the base domain, and the same

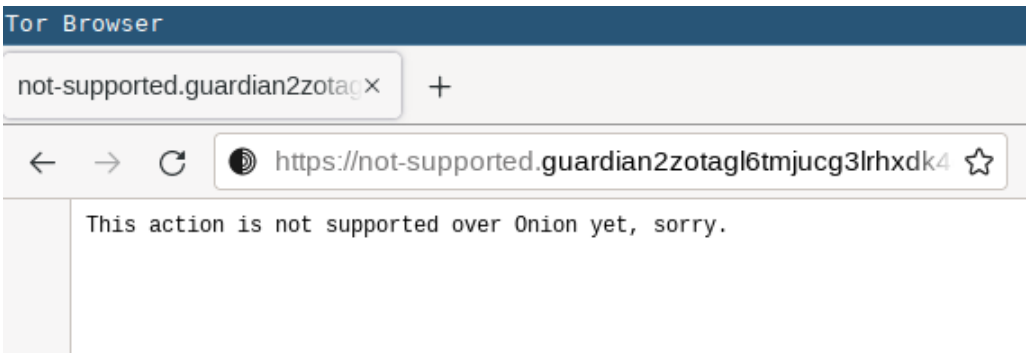


Figure 4. Screenshot of a web page that was unavailable on the The Guardian's Tor website.

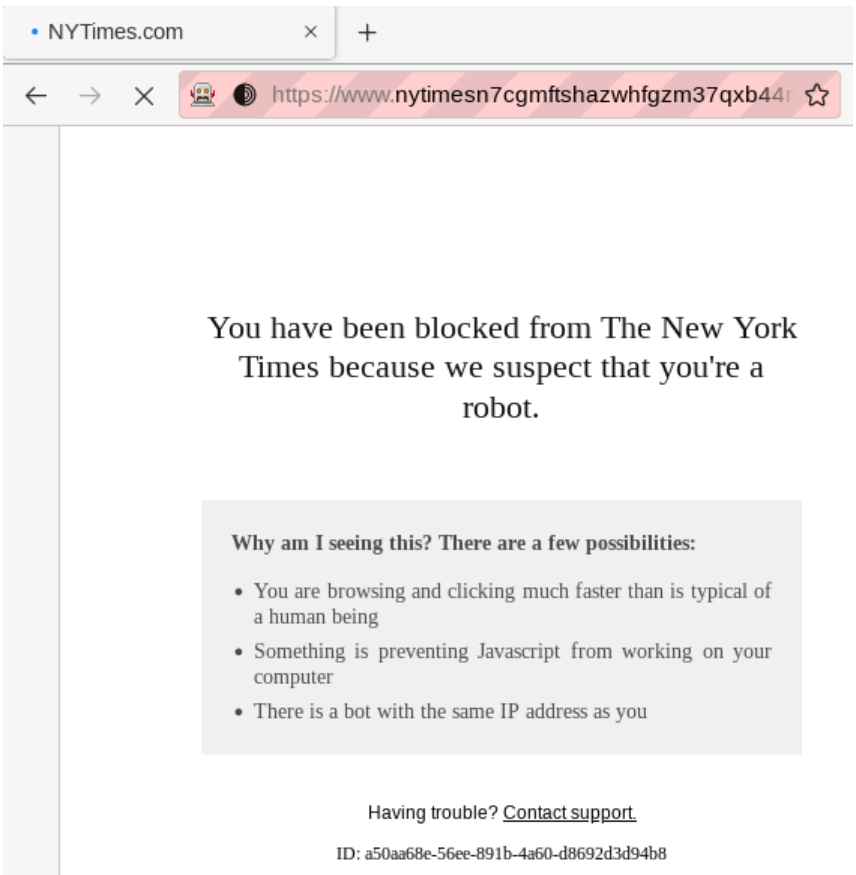


Figure 5. Screenshot of a web page shown when the crawler was blocked from The New York Times Tor website based on its Tor IP address.

CW/DW Website	Web Pages (CW, DW)	Identical Files	Identical File Names	Date, Timestamp(CW,DW) (CET)
Debian	50, 50	45	49	2022-09-09, 09:26:02-09:26:40, 09:09:13-09:21:31
Qube-OS	64, 64	17	61	2022-09-01, 13:15:57-13:18:40, 13:46:10-13:58:57
The Guardian	223, 211	0	204	2022-09-09, 10:16:43-10:24:13, 09:40:10-10:13:29
New York Times	200, 78	0	49	2022-09-16, 11:50:32-12:04:55, 11:18:14-11:48:12
CIA	42, 41	1	32	2022-09-09, 11:06:44-11:08:41, 11:18:57-12:14:46

Table 5. Number of pages downloaded from each website, the number of identical files, the number of files with the same title and file name, and the timestamps of the execution of the crawls.

CW/DW Website	Lowest (CS)	Highest (CS)	Mean (CS)	Median (CS)
Debian	0.9977	1.0	0.9999	0.9999
Qube-OS	0.9926	1.0	0.9966	0.9999
The Guardian	0.5584	0.9999	0.9927	0.9985
New York Times	0.3056	0.9999	0.9698	0.9981
CIA	0.8824	1.0	0.9842	0.9998

Table 6. Cosine (CS) similarity scores for the least similar and most similar web pages scraped from each web type of the same website - the clear (CW) and dark (DW) websites respectively.

CW/DW Website	Number of Images (CW, DW)	Identical Images	Difference
Debian	15, 15	14	1 different picture
Qube-OS	56, 56	38	18 different pictures
The Guardian	95 , 52	12	135 different pictures
The New York Times	77, 21	0	98 different pictures
CIA	24, 24	10	14 different pictures

Table 7. The total number of images downloaded by the crawler from each respective website together with the number of images that differed between the websites in each website pair.

Timestamps	2022-09-08, 13:58:58-14:18:45 (CET)
Duration	20 minutes 13 seconds
Pesudo-random page request delay interval	0-4 seconds
Links on homepage	251
Downloaded web pages	251
Unique images downloaded	2881
URL	http://hvilngbbx2yxtq7ilsrjsosv374phq4jx2nq5izot5baxlqyjy3u2cid.onion

Table 8. Details of a scraping of a dark marketplace website using the implemented crawler.

titles, although the content partially differed. The number of pages fetched from the New York Times' websites were very dissimilar due to the fact that the crawler was blocked from its onionsite. As a whole, the crawler managed to scrape the onionsites just as well as the clear websites; indicating that the Tor connection and crawler logic worked as intended.

The cosine similarity scores showed that the content for some pages differed, although there were a high similarity for the web pages that comprised pairs. The main point with the similarity scores was to verify that the dark web and clear web pages were to a large extent the same; i.e. that the sports page on The Guardian's clear and dark website was the same even though some content differ between the two web types.

The images downloaded by the crawler differed notably between the clear and dark websites of the Guardian and the New York Times. The log file of the crawler indicated that the few of the image elements found were actually downloaded, most likely due to the fact that the image resources the Guardian and New York Times onionsites were located on the clear web, where Tor exit node IP addresses were blocked from retrieving content. However, this does probably not affect onionsite crawling in the cybercrime context too much, since illicit and illegal onionsites seldom redirect to external resources like news websites do.

The crawling library, Selenium was originally built as a web testing tool, but has rendered into an effective web scraping tool. However, it is not in the most efficient library for web crawling and web scraping in terms of speed; although it has a few advantages that are relevant in digital investigations: (1) it allows user interaction, (2) it mimics human behaviour well, (3) it works visually and hence gives an investigator the option to watch as the crawler fetches each page, in this way the process is "invigilated". Arguably, this would increase the credibility of the scraping process as a means of collecting potential clues and evidence, subject to court admissibility.

Furthermore, by using Selenium, the crawler operator has the option to manually tweak and configure settings during runtime, such as bookmarking a page, clicking buttons, go back or refresh a page, enable or disable JavaScript, or establish a new Tor circuit.

The performance in regards to speed was for the Tor crawler slower than the clear web crawler, much due to the fact that the Tor network is slower network than the regular Internet not using the Onion Routing protocol. In addition, a delay of one to four seconds for fetching web pages was programmed into the crawlers in order to avoid being blocked by the web servers, although this was not a completely successful action since the New York Times' Onion site blocked the crawler, as noted in the experiments section.

The results from the experiments showed that the Tor

crawler managed to scrape 251 pages in 20 minutes, i.e. 12 pages per minute from the dark marketplace White House Market, while it took around eight minutes to scrape The Guardian's clear website of 223 pages, which is equal to circa 27.8 web pages per minute. Scraping 211 pages from The Guardian's dark web site took circa 33 minutes - an average of circa 6.4 pages per minute. In summary, the dark web crawler was more than four times slower.

Performance figures for crawlers akin to the one designed and implemented in this research article include number of pages retrieved and content classification accuracy rates. Unlike clear web crawling, time is not a relevant performance metric in dark web retrieval due to its resource intense encryption and routing scheme that requires more time and power than the clear web. One out of 44 articles mentions crawling duration and retrieval of web pages per minute and how it is increases with additional instances of browsers running the crawler, namely [71].

However, the clear web crawler developed could be compared to other clear web crawlers. According to Kumar, Bhatia, and Rattan [51], Mercator is one of the fastest open source crawlers, capable of retrieving 112 pages per second in 1999 [33]. There are numerous modern crawlers that reportedly are faster than the crawler presented in this paper, which averages to 0.46 web pages per second. However, it should be noted that the hardware requirements differ, as well as the software libraries used. The library used in this research was Selenium, which runs single-threaded in a graphical interface browser for usability reasons, this is a significantly slower architecture than multi-threaded distributed crawling engines used by for example or the Internet Archive or big search engines. There is a trade-off between speed and usability in this case, where the crawler presented in this research favoured usability over speed to fit its ultimate purpose.

Nevertheless, performance is not measured not only in execution speed, but also in efficacy and accuracy. Albeit, not all efficacy metrics were calculated and measured in equivalent manners as in this research. the Cosine similarity score for the pages downloaded by the crawlers implemented in this research are not reproducible and comparable with other crawlers' since web pages change. The relevancy is between the different versions of the crawlers developed as part of this research. Comparisons made would have been misguiding and misleading, not to say skewed and possibly incorrect.

VII. CONCLUSIONS AND FUTURE RESEARCH

The current research article addresses the problem with data collection in cybercrime cases; dark web related cybercrime cases in particular. The research problem presented pointed out the need for data collection tools in dark web investigations and suggests a solution to the problem by presenting a prototype that fulfilled a number of requirements for such a dark web investigative software tool. The scientific foundation that preceded the development of the suggested software consisted of a novel systematic literature review that included 58 research articles concerning crawling the dark web in favour of data collection; potential clues and evidence.

The main purpose of this research study was to establish knowledge regarding dark web crawlers in academic research. From this knowledge, a dark web crawler was developed to fit an already existing dark web cybercrime toolset called D3.

In combination with machine learning-based annotation and categorisation tools in D3, the crawler developed and presented in this article, will capacitate the toolset to automatically collect and classify web content based on previously annotated web pages. Ultimately, this will save manual labour for cybercrime investigators, without losing control over the crawling process. Neither will it compromise the forensic soundness of the overall process, since a certain amount of operator presence and interaction is necessary for URL selection, crawling scope specification, and user authentication for example. A logical continuation of this research would be to further elaborate on and test the toolset, and also make an expert or user evaluation of it.

A further assessment of crawler blocking mechanisms would be essential to establish a methodology for improving crawler performance in general, and on the Tor network in particular. From the over a decade long experience that the LEA community has from investigating illicit and illegal Tor websites in particular, and cybercrime in general, it can be suspected that Tor website administrators and programmers will improve their crawler blocking mechanisms and strengthen their authentication mechanisms; given that there is no competence deficiency amongst the unethical and criminal web developers, the digital cat-and-mouse game will continue.

ACKNOWLEDGEMENT

The work is partially supported by the NordForsk Grant No. 80512 for the project "Police Detectives on the TOR-network". The literature review data was downloaded from Scopus API via http://api.elsevier.com and http://www.scopus.com.

References

[1] Gunes Acar, Marc Juarez, and individual contributors. *tor-browser-selenium - Tor Browser automation with Selenium*. https://github.com/webfp/tor-browser-selenium. 2020.

[2] Judith Aldridge and David Décary-Hétu. "Hidden wholesale: The drug diffusing capacity of online drug cryptomarkets". In: *International Journal of Drug Policy* 35 (Sept. 2016), pp. 7–15. DOI: 10.1016/j.drugpo.2016.04.020. URL: https://doi.org/10.1016%2Fj.drugpo.2016.04.020.

[3] Abdullah Alharbi et al. "Exploring the Topological Properties of the Tor Dark Web". In: *IEEE Access* 9 (2021), pp. 21746–21758. DOI: 10.1109/access.2021.3055532. URL: https://doi.org/10.1109%2Faccess.2021.3055532.

[4] Konstantin Avrachenkov, Bruno Ribeiro, and Jithin K. Sreedharan. "Inference in OSNs via Lightweight Partial Crawls". In: *Proceedings of the 2016 ACM*

SIGMETRICS International Conference on Measurement and Modeling of Computer Science. ACM, June 2016. DOI: 10.1145/2896377.2901477. URL: https://doi.org/10.1145%2F2896377.2901477.

[5] Andres Baravalle, Mauro Sanchez Lopez, and Sin Wee Lee. "Mining the Dark Web: Drugs and Fake Ids". In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. 2016, pp. 350–356. DOI: 10.1109/ICDMW.2016.0056.

[6] Frederick Barr-Smith and Joss Wright. "Phishing With A Darknet: Imitation of Onion Services". In: *2020 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, Nov. 2020. DOI: 10.1109/ecrime51433.2020.9493262. URL: https://doi.org/10.1109%2Fecrime51433.2020.9493262.

[7] Jesper Bergman and Oliver B. Popov. "The Digital Detective's Discourse - A toolset for forensically sound collaborative dark web content annotation and collection". In: *Journal of Digital Forensics, Security and Law*. Vol. 15. 2022. DOI: https://doi.org/10.15394/jdfsl.2022.1740.

[8] Massimo Bernaschi et al. "Spiders like Onions: On the Network of Tor Hidden Services". In: *The World Wide Web Conference*. WWW '19. San Francisco, CA, USA: Association for Computing Machinery, 2019, pp. 105–115. ISBN: 9781450366748. DOI: 10.1145/3308558.3313687. URL: https://doi-org.ezp.sub.su.se/10.1145/3308558.3313687.

[9] Martin Bouchard, Kila Joffres, and Richard Frank. "Preliminary Analytical Considerations in Designing a Terrorism and Extremism Online Network Extractor". In: *Computational Models of Complex Systems*. Ed. by Vijay Kumar Mago and Vahid Dabbaghian. Cham: Springer International Publishing, 2014, pp. 171–184. ISBN: 978-3-319-01285-8. DOI: 10.1007/978-3-319-01285-8_11. URL: https://doi.org/10.1007/978-3-319-01285-8_11.

[10] Gwern Branwen et al. *Dark Net Market archives, 2011-2015*. https://www.gwern.net/DNM-archives. dataset. Accessed: 2020-02-02. July 2015. URL: https://www.gwern.net/DNM-archives.

[11] Sergio Castillo-Pérez and Joaquin Garcia-Alfaro. "Onion routing circuit construction via latency graphs". In: *Computers and Security* 37 (Sept. 2013), pp. 197–214. DOI: 10.1016/j.cose.2013.03.003. URL: https://doi.org/10.1016%2Fj.cose.2013.03.003.

[12] Alessandro Celestini and Stefano Guarino. "Design, implementation and test of a flexible tor-oriented web mining toolkit". In: *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics*. ACM, June 2017. DOI: 10.1145/3102254.3102266. URL: https://doi.org/10.1145%2F3102254.3102266.

[13] Edward Crowder and Jay Lansiquot. *Darknet Data Mining – A Canadian Cyber-crime Perspective*. 2021. arXiv: 2105.13957 [cs.CR].

[14] Vincent D. D'Agostino. *Complaint: United States of America v. Blake Benthall*. Retrieved: 2022-05-05. 2014. URL: https://www.justice.gov/usao/nys/pressreleases/November14/BlakeBenthallArrestPR/Benthall%2C%20Blake%20Complaint.pdf.

[15] Ashwini Dalvi et al. "From Hidden Wiki 2020 to Hidden Wiki 2021: What Dark Web Researchers Comprehend with Tor Directory Services?" In: *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*. IEEE, Oct. 2021. DOI: 10.1109/iscon52037.2021.9702384. URL: https://doi.org/10.1109%2Fiscon52037.2021.9702384.

[16] Ashwini Dalvi et al. "Link Harvesting on the Dark Web". In: *2021 IEEE Bombay Section Signature Conference (IBSSC)*. IEEE, Nov. 2021. DOI: 10.1109/ibssc53889.2021.9673428. URL: https://doi.org/10.1109%2Fibssc53889.2021.9673428.

[17] Ashwini Dalvi et al. "SpyDark: Surface and Dark Web Crawler". In: *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*. IEEE, May 2021. DOI: 10.1109/icsccc51823.2021.9478098. URL: https://doi.org/10.1109%2Ficsccc51823.2021.9478098.

[18] B. David, M. Delong, and E. Filiol. "Detection of crawler traps: formalization and implementation—defeating protection on internet and on the TOR network". In: *Journal of Computer Virology and Hacking Techniques* 17 (2021), pp. 185–198. DOI: https://doi.org/10.1007/s11416-021-00380-4.

[19] Baptiste David, Maxence Delong, and Eric Filiol. "Detection of crawler traps: formalization and implementation—defeating protection on internet and on the TOR network". In: *J Comput Virol Hack Tech* 17.3 (Apr. 2021), pp. 185–198. DOI: 10.1007/s11416-021-00380-4. URL: https://doi.org/10.1007%2Fs11416-021-00380-4.

[20] Gemma Davies. "Shining a Light on Policing of the Dark Web: An Analysis of UK Investigatory Powers". In: *The Journal of Criminal Law* 84.5 (2020), pp. 407–426. DOI: 10.1177/0022018320952557. eprint: https://doi.org/10.1177/0022018320952557. URL: https://doi.org/10.1177/0022018320952557.

[21] M. Delong, B. David, and E. Filiol. "Detection of crawler traps: Formalization and implementation defeating protection on internet and on the TOR network". In: cited By 1. 2020, pp. 775–783. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083024838&partnerID=40&md5=fbeb0e86415687386c466e0bb82ab7d3.

[22] Roger Dingledine, Nick Mathewson, and Paul Syverson. "Tor: The Second-Generation Onion Router". In: *Proceedings of the 13th USENIX Security Symposium*. Aug. 2004.

[23] Europol. *Definition of concern in English by Oxford Dictionaries*. Retrieved 02/10/2021. URL: https://www.europol.europa.eu/activities-services/main-reports

/internet-organised-crime-threat-assessment-iocta-2020.

[24] Mohd Faizan and Raees Ahmad Khan. "A Two-Step Dimensionality Reduction Scheme for Dark Web Text Classification". In: *Advances in Intelligent Systems and Computing*. Springer Singapore, 2020, pp. 303–312. DOI: 10.1007/978-981-15-1518-7_25. URL: https://doi.org/10.1007%2F978-981-15-1518-7_25.

[25] Philip L. Frana. "Before the Web There Was Gopher". In: *IEEE Annals of the History of Computing* 26.1 (2004), pp. 20–41. DOI: 10.1109/MAHC.2004.1278848.

[26] Tianjun Fu, Ahmed Abbasi, and Hsinchun Chen. "A focused crawler for Dark Web forums". In: *J. Am. Soc. Inf. Sci.* (2010), n/a–n/a. DOI: 10.1002/asi.21323. URL: https://doi.org/10.1002%2Fasi.21323.

[27] Keisuke Furumoto et al. "Extracting Threat Intelligence Related IoT Botnet From Latest Dark Web Data Collection". In: *2021 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing &amp Communications (GreenCom) and IEEE Cyber, Physical &amp Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*. IEEE, Dec. 2021. DOI: 10.1109/ithings-greencom-cpscom-smartdata-cybermatics53846.2021.00034. URL: https://doi.org/10.1109%2Fithings-greencom-cpscom-smartdata-cybermatics53846.2021.00034.

[28] S. Ghosh et al. "ATOL: A framework for automated analysis and categorization of the dark web ecosystem". In: vol. WS-17-01 - WS-17-15. cited By 5. 2017, pp. 170–178. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85046091132&partnerID=40&md5=97c23712172301caf6a2182866596ed2.

[29] Gaopeng Gou et al. "Discovering abnormal behaviors via HTTP header fields measurement". In: *Concurrency Computat.: Pract. Exper.* 29.20 (Aug. 2016), e3926. DOI: 10.1002/cpe.3926. URL: https://doi.org/10.1002%2Fcpe.3926.

[30] Y. Gupta et al. "Fuzzy logic based similarity measure for information retrieval system performance improvement". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8337 LNCS (2014). cited By 2, pp. 224–232. DOI: 10.1007/978-3-319-04483-5_23. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84958551881&doi=10.1007%2f978-3-319-04483-5_23&partnerID=40&md5=4f33184741d7c985b77f0a84e38dd7f4.

[31] D. Hayes, F. Cappa, and J. Cardon. "A Framework for More Effective Dark Web Marketplace Investigations". In: 9.8:186 (2018). DOI: 10.3390/info9080186.

[32] Darren Hayes, Francesco Cappa, and James Cardon. "A Framework for More Effective Dark Web Marketplace Investigations". In: *Information* 9.8 (July 2018), p. 186. DOI: 10.3390/info9080186. URL: https://doi.org/10.3390%2Finfo9080186.

[33] A. Heydon and M. Najork. "Mercator: A scalable, extensible Web crawler". In: *World Wide Web 2* (1999), pp. 219–229. URL: https://doi.org/10.1023/A:1019213109274.

[34] Svea Hovrätt. *Fällande dom i Flugsvamp 2.0-målet*. Retrieved: 2022-07-05. 2022. URL: https://www.domstol.se/nyheter/2022/06/fallande-dom-i-flugsvamp-2.0-malet/.

[35] Chih-Yuan Huang and Hao Chang. "GeoWeb Crawler: An Extensible and Scalable Web Crawling Framework for Discovering Geospatial Web Resources". In: *ISPRS International Journal of Geo-Information* 5.8 (2016). ISSN: 2220-9964. DOI: 10.3390/ijgi5080136. URL: https://www.mdpi.com/2220-9964/5/8/136.

[36] Hunchly. *Support*. Retrieved 22/5/2022. 2022. URL: https://hunch.ly//#support-faqs.

[37] IETF. *HTTP/1.0*. Retrieved 8/8/2022. 1996. URL: https://datatracker.ietf.org/doc/html/rfc1945.

[38] IETF. *HTTP/1.1*. Retrieved 6/8/2022. 1999. URL: https://datatracker.ietf.org/doc/html/rfc2616#section-5.

[39] IETF. *HTTP/3*. Retrieved 8/8/2022. 2022. URL: https://datatracker.ietf.org/doc/html/rfc9114.

[40] Christos Iliou et al. "Hybrid Focused Crawling for Homemade Explosives Discovery on Surface and Dark Web". In: *2016 11th International Conference on Availability, Reliability and Security (ARES)*. IEEE, Aug. 2016. DOI: 10.1109/ares.2016.66. URL: https://doi.org/10.1109%2Fares.2016.66.

[41] Christos Iliou et al. "Hybrid focused crawling on the Surface and the Dark Web". In: *EURASIP J. on Info. Security Journal on Information Security* 2017.1 (July 2017). DOI: 10.1186/s13635-017-0064-5. URL: https://doi.org/10.1186%2Fs13635-017-0064-5.

[42] Marc Juarez et al. "A Critical Evaluation of Website Fingerprinting Attacks". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS '14. Scottsdale, Arizona, USA: Association for Computing Machinery, 2014, pp. 263–274. ISBN: 9781450329576. DOI: 10.1145/2660267.2660368. URL: https://doi.org/10.1145/2660267.2660368.

[43] George Kalpakis et al. "Interactive Discovery and Retrieval of Web Resources Containing Home Made Explosive Recipes". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2016, pp. 221–233. DOI: 10.1007/978-3-319-39381-0_20. URL: https://doi.org/10.1007%2F978-3-319-39381-0_20.

[44] E.B. Gueguim Kana, Stefan Schmidt, and R.H. Azanfack Kenfack. "A web-enabled software for real-time biogas fermentation monitoring – Assessment of dark fermentations for correlations between medium conductivity and biohydrogen evolution". In: *International Journal of Hydrogen Energy* 38.25 (Aug. 2013),

pp. 10235–10244. DOI: 10.1016/j.ijhydene.2013.06.019. URL: https://doi.org/10.1016%2Fj.ijhydene.2013.06.019.

[45] Yuki Kawaguchi, Akira Yamada, and Seiichi Ozawa. "AI Web-Contents Analyzer for Monitoring Underground Marketplace". In: *Neural Information Processing*. Springer International Publishing, 2017, pp. 888–896. DOI: 10.1007/978-3-319-70139-4_90. URL: https://doi.org/10.1007%2F978-3-319-70139-4_90.

[46] B. Kitchenham. *Joint Procedures for Undertaking Systematic Reviews*. Technical Report. Accessed: 2021-11-02. 2004.

[47] Barbara Kitchenham. "Procedures for performing systematic reviews". In: *Keele, UK, Keele University* 33.2004 (2004), pp. 1–26.

[48] Barbara Kitchenham and Stuart Charters. "Guidelines for performing systematic literature reviews in software engineering". In: (2007).

[49] Paris Koloveas et al. "A Crawler Architecture for Harvesting the Clear, Social, and Dark Web for IoT-Related Cyber-Threat Intelligence". In: *2019 IEEE World Congress on Services (SERVICES)*. IEEE, July 2019. DOI: 10.1109/services.2019.00016. URL: https://doi.org/10.1109%2Fservices.2019.00016.

[50] Paris Koloveas et al. "inTIME: A Machine Learning-Based Framework for Gathering and Leveraging Web Data to Cyber-Threat Intelligence". In: *Electronics* 10.7 (Mar. 2021), p. 818. DOI: 10.3390/electronics10070818. URL: https://doi.org/10.3390%2Felectronics10070818.

[51] Manish Kumar, Rajesh Bhatia, and Dhavleesh Rattan. "A survey of Web crawlers for information retrieval". In: *WIREs Data Mining and Knowledge Discovery* 7.6 (2017), e1218. DOI: https://doi.org/10.1002/widm.1218. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1218. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1218.

[52] Jinhee Lee et al. "Shedding Light on Dark Korea: An In-Depth Analysis and Profiling of the Dark Web in Korea". In: *Information Security Applications*. Springer International Publishing, 2020, pp. 357–369. DOI: 10.1007/978-3-030-39303-8_27. URL: https://doi.org/10.1007%2F978-3-030-39303-8_27.

[53] Andrey I. Levin and Igor A. Voronov. "The investigation of the possibility of automated collection of information in the hidden segment of the Internet". In: *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. IEEE, Jan. 2018. DOI: 10.1109/eiconrus.2018.8317031. URL: https://doi.org/10.1109%2Feiconrus.2018.8317031.

[54] Runchuan Li et al. "Edge-Based Detection and Classification of Malicious Contents in Tor Darknet Using Machine Learning". In: *Mobile Information Systems* 2021 (Nov. 2021). Ed. by Ke Gu, pp. 1–13. DOI: 10.1155/2021/8072779. URL: https://doi.org/10.1155%2F2021%2F8072779.

[55] Ben Light, Peta Mitchell, and Patrik Wikström. "Big Data, Method and the Ethics of Location: A Case Study of a Hookup App for Men Who Have Sex with Men". In: *Social Media + Society* 4.2 (Apr. 2018), p. 205630511876829. DOI: 10.1177/2056305118768299. URL: https://doi.org/10.1177%2F2056305118768299.

[56] Xiaofeng Liu et al. "Fingerprinting Web Browser for Tracing Anonymous Web Attackers". In: *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*. IEEE, June 2016. DOI: 10.1109/dsc.2016.78. URL: https://doi.org/10.1109%2Fdsc.2016.78.

[57] Usha Lokala et al. "Global trends, local harms: availability of fentanyl-type drugs on the dark web and accidental overdoses in Ohio". In: *Comput Math Organ Theory* 25.1 (Oct. 2018), pp. 48–59. DOI: 10.1007/s10588-018-09283-0. URL: https://doi.org/10.1007%2Fs10588-018-09283-0.

[58] Envolve Forensics LTD. *Products*. Retrieved 22/5/2022. 2021. URL: https://en.fawproject.com/products/.

[59] Yadav M. and Goyal N. "Comparison of Open Source Crawlers - A Review". In: *Int. J. Sci. Eng. Res.* 6.9 (2015). ISSN: 2229-5518. URL: https://www.ijser.org/researchpaper/Comparison-of-Open-Source-Crawlers--A-Review.pdf.

[60] Sergio Mauricio Martinez Monterrubio et al. "Black Widow Crawler for TOR network to search for criminal patterns". In: *2021 Second International Conference on Information Systems and Software Technologies (ICI2ST)*. IEEE, Mar. 2021. DOI: 10.1109/ici2st51859.2021.00023. URL: https://doi.org/10.1109%2Fici2st51859.2021.00023.

[61] Vadim S. Murov and Anton V. Arzhskov. "Vulnerability Research Onion Sites TOR". In: *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. 2020, pp. 423–425. DOI: 10.1109/EIConRus49466.2020.9039300.

[62] n.A. "12th International Conference on Security, Privacy, and Anonymity in Computation, Communication, and Storage, SpaCCS 2019". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11637 LNCS (2019). cited By 0. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85069854790&partnerID=40&md5=6e17b01d0612876707646833de17c0e9.

[63] n.A. "AHFE International Conference on Human Factors in Cybersecurity, 2018". In: *Advances in Intelligent Systems and Computing* 782 (2019). cited By 0. URL: https://www.scopus.com/inward/record.u

ri?eid=2-s2.0-85049642445&partnerID=40&md5=213b5b75511f3e9b31f803c566322954.

[64] n.A. "DSC 2018 - 2018 IEEE Conference on Dependable and Secure Computing". In: cited By 0. 2019. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85062544984&partnerID=40&md5=9d4180dfcdf29933615a1ea30125c61c.

[65] n.A. "Financial Cryptography and Data Security - FC 2011 Workshops, RLCPS and WECSR 2011, Revised Selected Papers". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7126 LNCS (2012). cited By 0. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84861444321&partnerID=40&md5=380a67d62b187e9ff74b4392abed1e74.

[66] Midas Nouwens et al. "Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Apr. 2020. DOI: 10.1145/3313831.3376321. URL: https://doi.org/10.1145%2F3313831.3376321.

[67] Juha Nurmi and individual contributors. *Ahmia Crawler*. https://github.com/ahmia/ahmia-crawler. 2021.

[68] OSIRT. *Support*. Retrieved 22/5/2022. 2022. URL: https://www.osirtbrowser.com/.

[69] Mandeep Pannu, Iain Kay, and Daniel Harris. "Using Dark Web Crawler to Uncover Suspicious and Malicious Websites". In: *Advances in Intelligent Systems and Computing*. Springer International Publishing, June 2018, pp. 108–115. DOI: 10.1007/978-3-319-94782-2_11. URL: https://doi.org/10.1007%2F978-3-319-94782-2_11.

[70] Jonghyeon Park, Hyunsu Mun, and Youngseok Lee. "Improving Tor Hidden Service Crawler Performance". In: *2018 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, Dec. 2018. DOI: 10.1109/desec.2018.8625103. URL: https://doi.org/10.1109%2Fdesec.2018.8625103.

[71] Jonghyeon Park, Hyunsu Mun, and Youngseok Lee. "Improving Tor Hidden Service Crawler Performance". In: *2018 IEEE Conference on Dependable and Secure Computing (DSC)*. 2018, pp. 1–8. DOI: 10.1109/DESEC.2018.8625103.

[72] E. Perjons and P. Johannesson. *An Introduction to Design Science*. Springer International Publishing, 2014. ISBN: ISBN 978-3-319-10632-8. DOI: DOI10.1007/978-3-319-10632-8.

[73] Kien Pham, Aécio Santos, and Juliana Freire. "Understanding Website Behavior based on User Agent". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, July 2016. DOI: 10.1145/2911451.2914757. URL: https://doi.org/10.1145%2F2911451.2914757.

[74] O. Popov, J. Bergman, and C. Valassi. "A Framework for a Forensically Sound Harvesting the Dark Web". In: *CECC 2018: Proceedings of the Central European Cybersecurity Conference 2018*. ACM, 2017, pp. 1–7. DOI: https://doi.org/10.1145/3277570.3277584.

[75] K. Porter. "Analyzing the DarkNetMarkets subreddit for evolutions of tools and trends using LDA topic modeling". In: cited By 1. 2018, S87–S97. DOI: 10.1016/j.diin.2018.04.023. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85068692166&doi=10.1016%2fj.diin.2018.04.023&partnerID=40&md5=d67e695593797d5d78d299c62ee69275.

[76] Kyle Porter. "Analyzing the DarkNetMarkets subreddit for evolutions of tools and trends using LDA topic modeling". In: *Digital Investigation* 26 (July 2018), S87–S97. DOI: 10.1016/j.diin.2018.04.023. URL: https://doi.org/10.1016%2Fj.diin.2018.04.023.

[77] R. S. Portnoff et al. "Tools for Automated Analysis of Cybercriminal Markets". In: *International World Wide Web Conference Committee (IW3C2)*. ACM, 2017, pp. 1–5. DOI: https://doi.org/10.1145/3038912.3052600.

[78] Selenium Project. *Limitations of scaling up tests in Selenium 2*. Retrieved 19/09/2022. 2022. URL: https://www.selenium.dev/documentation/legacy/selenium_2/parallel_execution/#running-parallel-selenium2.

[79] Tor Project. *JavaScript | Tor Project | Support*. Retrieved: 2022-07-05. 2022. URL: https://support.torproject.org/glossary/javascript/.

[80] John T. Rabaut. *Complaint: United States of America v. Alexandre Cazes*. Retrieved: 2022-07-05. 2017. URL: https://www.justice.gov/opa/press-release/file/982821/download.

[81] I Gede Surya Rahayuda and Ni Putu Linda Santiari. "Crawling and cluster hidden web using crawler framework and fuzzy-KNN". In: *2017 5th International Conference on Cyber and IT Service Management (CITSM)*. IEEE, Aug. 2017. DOI: 10.1109/citsm.2017.8089225. URL: https://doi.org/10.1109%2Fcitsm.2017.8089225.

[82] Mohammad Al-Ramahi, Izzat Alsmadi, and Joshua Davenport. "Exploring hackers assets". In: *Proceedings of the 7th Symposium on Hot Topics in the Science of Security*. ACM, Aug. 2020. DOI: 10.1145/3384217.3385619. URL: https://doi.org/10.1145%2F3384217.3385619.

[83] Romil Rawat et al. "Dark Web—Onion Hidden Service Discovery and Crawling for Profiling Morphing, Unstructured Crime and Vulnerabilities Prediction". In: *Lecture Notes in Electrical Engineering*. Springer Singapore, 2021, pp. 717–734. DOI: 10.1007/978-981-16-0749-3_57. URL: https://doi.org/10.1007%2F978-981-16-0749-3_57.

[84] Abdul Razaque et al. "Influence of COVID-19 Epidemic on Dark Web Contents". In: *Electronics* 10.22

(Nov. 2021), p. 2744. DOI: 10.3390/electronics10222744. URL: https://doi.org/10.3390%2Felectronics10222744.

[85] G. Roja et al. "A credit card fake detection system using image cryptography". In: *International Journal of Recent Technology and Engineering* 7.6 (2019). cited By 1, pp. 118–122. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85067982842&partnerID=40&md5=1731fb681f03efec95814a4bd5bde7a4.

[86] G Rossi. "Halo shapes, initial shear field, and cosmic web". In: *J. Phys.: Conf. Ser.* 484 (Mar. 2014), p. 012049. DOI: 10.1088/1742-6596/484/1/012049. URL: https://doi.org/10.1088%2F1742-6596%2F484%2F1%2F012049.

[87] Thabit Sabbah et al. "Hybridized term-weighting method for Dark Web classification". In: *Neurocomputing* 173 (2016), pp. 1908–1926. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2015.09.063. URL: https://www.sciencedirect.com/science/article/pii/S092523121501396X.

[88] Scopus. *Scopus Source List*. 2022. URL: https://www.elsevier.com/?a=91122.

[89] Ryan Scrivens et al. "Searching for Extremist Content Online Using the Dark Crawler and Sentiment Analysis". In: *Methods of Criminology and Criminal Justice Research*. Emerald Publishing Limited, Aug. 2019, pp. 179–194. DOI: 10.1108/s1521-613620190000024016. URL: https://doi.org/10.1108%2Fs1521-613620190000024016.

[90] Stavros Shiaeles, Nicholas Kolokotronis, and Emanuele Bellini. "IoT Vulnerability Data Crawling and Analysis". In: *2019 IEEE World Congress on Services (SERVICES)*. IEEE, July 2019. DOI: 10.1109/services.2019.00028. URL: https://doi.org/10.1109%2Fservices.2019.00028.

[91] Vidyesh Shinde et al. "CrawlBot: A Domain-Specific Pseudonymous Crawler". In: *Communications in Computer and Information Science*. Springer International Publishing, 2021, pp. 89–101. DOI: 10.1007/978-3-030-84842-2_7. URL: https://doi.org/10.1007%2F978-3-030-84842-2_7.

[92] R. Singh et al. "Characterizing the nature and dynamics of tor exit blocking". In: cited By 19. 2017, pp. 325–341. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85056378676&partnerID=40&md5=1d5c86adccea06078c655008daf814b0.

[93] Tor-Project. *Can I use Tor with a browser besides Tor Browser?* Retrieved 09/09/2020. 2022. URL: https://support.torproject.org/tbb/.

[94] Tor-Project. *How do I check if my application that uses SOCKS is leaking DNS requests?* Retrieved 09/09/2022. 2022. URL: https://support.torproject.org/ca/misc/check-socks-dns-leaks/.

[95] W3C. *Facts About W3C*. Retrieved: 2020-09-05. 2022. URL: https://www.w3.org/Consortium/facts.

[96] Joseph Williams and Paul Stephens. "Analysis of the 'Open Source Internet Research Tool': A Usage Perspective from UK Law Enforcement". In: *Human Aspects of Information Security and Assurance*. Ed. by Nathan Clarke and Steven Furnell. Cham: Springer International Publishing, 2020, pp. 341–352. ISBN: 978-3-030-57404-8.

[97] Philipp Winter et al. "How Do Tor Users Interact With Onion Services?" In: *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018, pp. 411–428. ISBN: 978-1-939133-04-5. URL: https://www.usenix.org/conference/usenixsecurity18/presentation/winter.

[98] Allison Woodruff et al. "An investigation of documents from the World Wide Web". In: *Computer Networks and ISDN Systems* 28.7-11 (1996). Cited by: 27; All Open Access, Green Open Access, 963 â 980. DOI: 10.1016/0169-7552(96)00064-5. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-33750402884&doi=10.1016%2f0169-7552%2896%2900064-5&partnerID=40&md5=a7b3de0659b4c6fbd7c25327da48e9dc.

[99] Yubao Wu et al. "Python Scrapers for Scraping Cryptomarkets on Tor". In: *Security, Privacy, and Anonymity in Computation, Communication, and Storage*. Springer International Publishing, 2019, pp. 244–260. DOI: 10.1007/978-3-030-24907-6_19. URL: https://doi.org/10.1007%2F978-3-030-24907-6_19.

[100] Yingying Xu et al. "Research on Dark Web Monitoring Crawler Based on TOR". In: *2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*. IEEE, Dec. 2021. DOI: 10.1109/iciba52610.2021.9687954. URL: https://doi.org/10.1109%2Ficiba52610.2021.9687954.

[101] Desheng Yang and Pree Thiengburanathum. "Scalability and Robustness Testing for Open Source Web Crawlers". In: *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*. 2021, pp. 197–201. DOI: 10.1109/ECTIDAMTNCON51128.2021.9425701.

[102] Li Yang et al. "Discovering topics from dark websites". In: *2009 IEEE Symposium on Computational Intelligence in Cyber Security*. IEEE, Mar. 2009. DOI: 10.1109/cicybs.2009.4925106. URL: https://doi.org/10.1109%2Fcicybs.2009.4925106.

[103] Ying Yang et al. "Hadoop-based Dark Web Threat Intelligence Analysis Framework". In: *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. IEEE, Oct. 2019. DOI: 10.1109/imcec46724.2019.8984106. URL: https://doi.org/10.1109%2Fimcec46724.2019.8984106.

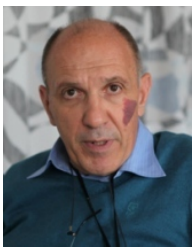
[104] Ying yang et al. "Crawling and Analysis of Dark Network Data". In: *Proceedings of 2020 the 6th International Conference on Computing and Data Engineering*. ACM, Jan. 2020. DOI: 10.1145/3379247.3379272. URL: https://doi.org/10.1145%2F3379247.3379272.

[105] Hengrui Zhang and Futai Zou. "A Survey of the Dark Web and Dark Market Research". In: *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. IEEE, Dec. 2020. DOI: 10.1109/iccc51575.2020.9345271. URL: https://doi.org/10.1109%2Ficcc51575.2020.9345271.

[106] Ning Zhang et al. "Counteracting Dark Web Text-Based CAPTCHA with Generative Adversarial Learning for Proactive Cyber Threat Intelligence". In: *ACM Trans. Manage. Inf. Syst. Transactions on Management Information Systems* 13.2 (June 2022), pp. 1–21. DOI: 10.1145/3505226. URL: https://doi.org/10.1145%2F3505226.

[107] Ahmed T. Zulkarnine et al. "Surfacing collaborated networks in dark web to find illicit and criminal content". In: *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. IEEE, Sept. 2016. DOI: 10.1109/isi.2016.7745452. URL: https://doi.org/10.1109%2Fisi.2016.7745452.

[108] Ahmed T. Zulkarnine et al. "Surfacing collaborated networks in dark web to find illicit and criminal content". In: *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. 2016, pp. 109–114. DOI: 10.1109/ISI.2016.7745452.



OLIVER B. POPOV holds a Ph.D. in Computer Science (Artificial Intelligence, 1987) from the Missouri University of Science and Technology, Rolla, USA. He is a professor of Computer science on the area of Information security and forensics at the Department of Computer and Systems Sciences, Stockholm university. Dr. Popov has been a professor at Faculty of Computer Science and Computer Engineering, Saints Cyril and Methodius University and a professor at the Department of Information Technology and Media, Mid Sweden University. In the last thirty years he has participated in more than fifty international research projects, and the corpus of work comprises of 200 peer-reviewed publications including journal and conference articles, book chapters, and nine books.

PLACE PHOTO HERE

JESPER BERGMAN read for a BSc degree in Computer and Systems Sciences at Stockholm university, where he also completed his MSc degree with a major in Information Security in 2016. After a several years in the industry, he became a teaching and research assistant at the Department of Computer and Systems Sciences, where he started his PhD studies in digital forensics in 2017. In the last five years, Mr. Bergman has participated in a several EU and national research projects.