Snorkeling in dark waters: A longitudinal surface exploration of unique Tor Hidden Services (Extended Version)

Alfonso Rodriguez Barredo-Valenzuela[†‡], Sergio Pastrana Portillo[‡], Guillermo Suarez-Tangil[†]

[†]*IMDEA Networks Institute*, [‡]*Universidad Carlos III de Madrid*

Abstract—The Onion Router (Tor) is a controversial network whose utility is constantly under scrutiny. On the one hand, it allows for anonymous interaction and cooperation of users seeking untraceable navigation on the Internet. This freedom also attracts criminals who aim to thwart law enforcement investigations, e.g., trading illegal products or services such as drugs or weapons. Tor allows delivering content without revealing the actual hosting address, by means of `.onion` (or hidden) services. Different from regular domains, these services can not be resolved by traditional name services, are not indexed by regular search engines, and they frequently change. This generates uncertainty about the extent and size of the Tor network and the type of content offered.

In this work, we present a large-scale analysis of the Tor Network. We leverage our crawler, dubbed Mimir, which automatically collects and visits content linked within the pages to collect a dataset of pages from more than 25k sites. We analyze the topology of the Tor Network, including its depth and reachability from the surface web. We define a set of heuristics to detect the presence of replicated content (mirrors) and show that most of the analyzed content in the Dark Web ($\approx 82\%$) is a replica of other content. Also, we train a custom Machine Learning classifier to understand the type of content the hidden services offer. Overall, our study provides new insights into the Tor network, highlighting the importance of initial seeding for focus on specific topics, and optimize the crawling process. We show that previous work on large-scale Tor measurements does not consider the presence of mirrors, which biases their understanding of the Dark Web topology and the distribution of content.

Index Terms—Tor Network Measurement, Tor Content Analysis, Mirror Detection

I. INTRODUCTION

The Onion Router (Tor) has grown considerably since its emergence in 2002 [21], [6], [42],

and has become a widely used platform for users and communities seeking privacy and anonymity while navigating the Internet. For example, it is the main social technology used for cooperative work for marginal communities, such as LGTBI+, in countries where their activities are forbidden and even penalized [13]. Moreover, Tor offers a preferred online space for political activists to organize and share information [17]. Besides allowing clients to navigate anonymously, Tor also allows content providers to offer web services without revealing the actual IP address hosting the server. This is done using so-called *hidden services* or *.onion sites*. To access these services, users need to establish a connection through the Tor Network, thus preserving the anonymity of clients and servers. This neutral technology poses a dilemma since it can be used both for benign purposes (e.g., censorship avoidance, whistle-blowing, and activism), and all sorts of malicious services (e.g., drug dealing, terrorism,, distribution of child abuse material, or data breaches) [19].

The Tor Network provides the infrastructure to what is referred to as the Dark Web, i.e., the portion of the Web that can not be easily accessed by standard tools.

The Dark Web often hosts marketplaces trading digital goods that are key to conducting cyberattacks [39]. The Dark Web also hosts online forums where criminals often publish data leaks from ransomware attacks or data breaches [8]. Consequently, the analysis of the Dark Web is important to understand modern cyber threats and adversaries. However, there are several challenges that analysts must face when collecting and analyzing such information, like the volatility of the information or the unreliability of the connections.

Various research works have studied the Dark Web in the past, most of them focusing on specific content (e.g., marketplaces [36], [39], [24] or criminal activities [22], [16])

These studies help to understand a niche of the Dark Web but do not provide a general overview of its size, connectivity, and overall contents offered. Prior work that provides an overview of the Tor network dates back several years [28], [2], [14], [4], [6], [7] and does not consider the prevalence of replicated hidden services scattered through the Tor network. Thus, they offer a biased and outdated picture of the prevalence of content on the Dark Web. As Tor is a volatile network [6], [10] with duplicated content across the board, there is a need for a more comprehensive and generic study of the topology and the prevalence of different types of contents on the Dark Web.

In this work, we propose a crawling methodology designed to longitudinally and reliably map second-level domain services in Tor. We conduct shallow, but in-breath, crawling, i.e., once the crawler visits the landing page of a site, it moves to different ones — avoiding an in-depth exploration of the site. Our aim is to maximize the crawling coverage of the Dark Web

to capture sites related to deviant content (i.e., "actions that violate social norms, which may include both informal social rules or more formal societal expectations and laws" [18]), often linked to cybercrime activities, without overloading the network with a deeper crawling of the sites we visit. As such, sites of particular interest can be efficiently detected to conduct in-depth analysis (as we show in the case study described in §IV). As a key distinction, our methodology pinpoints Web pages that are predominantly the same or very similar (namely mirrors). We thus perform the first mirrorless network analysis to study the topology and the type of content hosted on the Dark Web, their language, and the role that mirrors play when measuring hidden services behind Tor. We present a case study of how our crawler reaches 159 sites hosting child abuse content, including live cams with pornographic content for pedophiles.

Overall, our main contributions are:

- We build a custom crawler (Mimir) that allows us to efficiently navigate through the Tor Network (§II). The crawler departs from a set of seeds that are obtained systematically, by querying, in a novel way, dedicated .onion search engines for trending topics in underground surface forums. Our results (§III) show that our crawling methods are more effective than those in other works (§VI), reaching a larger portion of the network from a reduced set of seeds.
- To measure the prevalence of replication, we develop and validate an effective heuristic-based algorithm for mirror detection (§II). We notice that most ($\approx 82\%$) of the landing pages in our dataset are a replica (§III). This insight is key when accurately measuring the prevalence of genuine deviant content (§V) in the Dark Web, which we conduct in this work by filtering out duplicates for the first time in the literature.
- Our insight above motivates us to characterize the type and study the amount of activities potentially linked to cybercrime on the Dark Web. We source a known dataset of labeled sites to build a custom content classifier (§II). We design an efficient learning pipeline that performs well despite the scarcity of data . The classification allows us to validate our crawling strategy. We show that a large proportion of the unique sites of our dataset (74.4%) is dedicated to cybercrime activities (§III).
- As a case study, we illustrate how Mimir detects 159 Tor sites related to the distribution of child abuse material, including a site offering live cams over underage (§IV). These sites have been reported to our Law Enforcement agency and are currently under investigation.

II. METHODOLOGY

Our methodology comprises three building blocks (see Figure 1): a crawler for the data collection, a mirror analysis algorithm for the identification of replicated (or slightly modified) sites, and a module for network analysis and content classification that serves to validate the effectiveness of our proposed crawling pipeline. It takes as input an initial set of seeds which, unlike prior work, are extracted systematically.

A. Challenges

Analyzing and comprehensively understanding the Dark Web presents staggering challenges that have an impact on the design of our methodology, which we describe next.

Scheduling strategy. Volatility is a well-known characteristic of the Darknet Tor [6]. Some Hidden Services (HS) may be up for intermittent periods, and thus it is necessary to attempt connection to a given site multiple times before discarding the link for not being online. This increases the overload of the network. To be compliant with the safety guidelines of the Tor Research Safety Goals [33], i.e., minimization, we need to find a trade-off between the coverage of our study, and the overload caused by the network. Accordingly, one design requirement of our system is to perform a crawling that aims to maximize the number of unique websites encountered, even at the cost of not exploring each site in its entirety. As such, our system prioritizes breadth over depth in its crawling strategy and performs what we call a 'shallow' crawl. By prioritizing breadth, our system gains a wider perspective of the Dark Web.

Content analysis. Media content such as images and GIFs are common ways to deliver information. Images and GIFs are a common way to convey information. Such media contains valuable information, and it is frequently used in information retrieval processes. However, due to legal and ethical advice, we can not use this information in Tor. As our case study demonstrates, longitudinal crawling often encounters sites distributing illegal content like child abuse media. Note that the download of child abuse content alone is illegal and considered a serious crime. Thus, we require our crawler to retrieve only text and our classification systems to perform accurately using non-media content. Other studies such as the one conducted by Claudia Peersman et al. [31] have addressed the challenge of analyzing content in child abuse images through a strict collaboration with law enforcement and under a very specific context. The broad nature of our crawling process does not meet the necessary requirements.

B. Crawling Process

The crawling process collects (Dark) Web pages using several steps. The process starts with the crawling of the sites obtained from an initial set of seeds (which we describe below). For each site, the crawler extracts the information defined in Table I, including links to other pages. These are added in a TODOLIST for subsequent crawling iterations. After a link is successfully crawled, Mimir continues with the next item in the TODOLIST until it is empty. Since sites may sometimes be offline, and also to ensure that non-availability is not due to an eventual network error, we revisit (at most 5 times) each URL responding with an HTTP code different than 200.

If an URL remains unavailable after all 5 attempts, we remove it from the TODOLIST and flag it as *Unreachable*. Next, we detail the crawling process.

Initial seeding.

We use a systematic approach to collect the initial seed of .onion sites. Specifically, we automatically query three popular Tor Search Engines (TSE) for hidden services with relevant
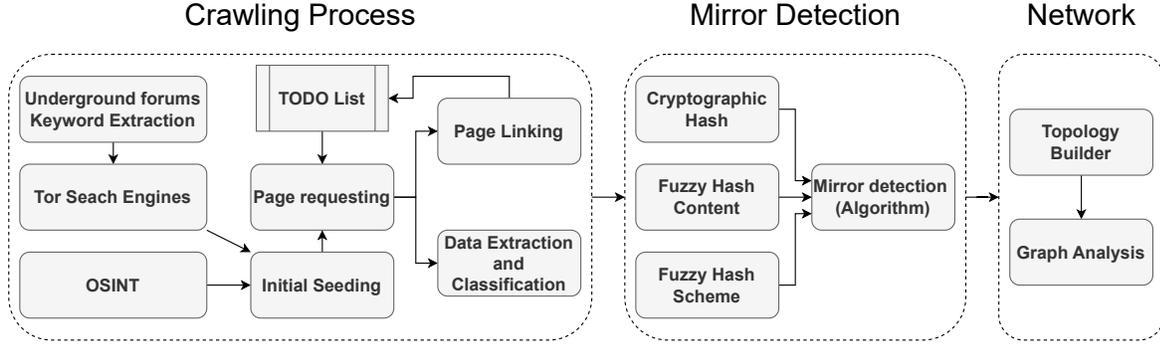
Figure 1: Overview of the methodology followed for our study.

Table I: Attributes extracted for each page crawled.

Attribute	Description
URL	Full `.onion` address
Metadata	Metadata of the page
Link list	A list of the links contained in the page
Referenced	List of the pages which has references to this page
HTML	The raw HTML code of the page
Timestamp	String with the Timestamp in format "dd-MM-yy HH:mm:ss"
Language List	A list of the languages contained in the page
Depth	The number of links from the seed to this page

keywords: *Ahmia* [27], *Torch* [38] and *VisiTor* [40]. To provide these keywords, we extract the most relevant words from the forum titles in underground forums [30] using the Term Frequency – Inverse Document Frequency (TFIDF) approach, as we further explained in Section II-E. Our hypothesis is that discussions in underground forums are a useful source for the initial seeding of deviant content in Tor. We note that, while this work focuses on content related to deviant behavior, Mimir can be used to investigate other topics by using a different corpus to extract keywords for the seeding process.

Finally, to verify that prominent .onion sites appear in our initial seed we use:

1) Domain knowledge and references from previous works (e.g., Hidden Wiki or specific GitHub repositories);
2) Literature search to improve the previous domain knowledge and contextualize the current state of art seeking for new relevant seeds;
3) Open Source Intelligence (OSINT), i.e., TSE to collect new seeds or to assess if existing seeds are informative.

All of the tasks mentioned above require some level of manual effort but result in the inclusion of additional resources that complement the TSE search.

Page requesting. We use *workload windows* to run parallel crawling and dynamically allocate links that need to be crawled by different running threads. This strategy avoids the use of additional synchronization mechanisms. As depicted in Figure 2, Each thread has only one link to crawl per window, as the number of links in each chunk is equal to the number of threads. Thus, if a given thread finishes with its link, it crawls $Link_{k+\#threads}$, where k is the link number. To remove links from TODOLIST(), all links for the window must be completed (i.e., either a valid HTTP response was received or the request timed out). Unsuccessful links are added again to the TODOLIST if they have been attempted less than 5 times. Otherwise, they are marked as *unreachable*. If the links

are successfully requested, we run the steps detailed in the following sections.
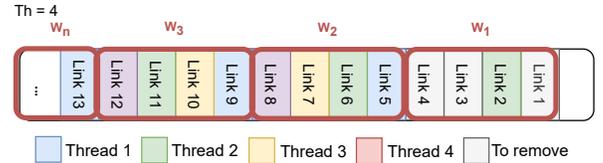

Figure 2: Dynamic workload algorithm example with 4 threads.

Data extraction. We process all .onion sites we crawl and we extract the information shown in Table I. As we crawl, we scrape the list of links to other hidden services and add them to the TODOLIST. We also scrape the following information, which is then used for the analysis: i) the language of the site, by using *langdetect* library [35], ii) the depth from the seed which leads to a page being currently crawled, and iii) the timestamp of the crawl. Finally, we also store the raw HTML, for further content and mirror analysis.

Page linking. This step uses regular expressions to extract linked .onion pages in a given site, to further guide the crawler. The relationship between pages lets us determine the network topology of Tor. We classify the links we extract into three categories:

1) *Own links:* Sub-pages hosted within the same domain.
2) *External .onion links:* These are links to other hidden services (i.e., pointing to an external .onion site) extracted from the raw HTML.
3) *Surface links:* These are the links that have a Top Level Domain other than .onion.

Since our goal is to conduct horizontal crawling, only external .onion links are added to the TODOLIST (after removing all characters at the right side of the domain), for being crawled in subsequent iterations. During this process, we build a link list for each site that allows us to rebuild the crawling path from the original seed. The path is updated to build a network graph upon observing new cross-links.

C. Mirror Analysis

To reduce our dataset to a set of singular pages, this step identifies mirrors. A mirror is an exact or nearly equal copy of

another `.onion` site. Our rationale is that the presence of mirrors introduces bias to existing Tor content- or topology-driven analysis. For example, the presence of the same market under different `.onion` addresses leads to the false impression that there are more markets of the same kind. While most of the mirrors are an exact clone of a site, an exploratory analysis shows a frequent deployment of sites with minor modifications (e.g., a market with the same products but with prices in a different currency — typically cryptocurrencies vs. dollars). We also see the same site translated into different languages and we thus consider them as mirrors. This insight informs the development of a similarity measure robust in the face of inconsequential modifications as we discuss next.

Web content vs web structure. There are two main elements of a web page that we compare to understand how similar they are: 1) the content of the page, and 2) its structure (namely HTML scheme). Comparing two pages based on their content alone would not capture situations where the page offers the same content but in different formats (e.g., translations of the same content into different languages, or pricing in different currencies based on the locale of the user). Conversely, relying only on the HTML scheme would flag as mirrors sites that use the same underlying framework (e.g., two blogs with the same WordPress template). The scheme is extracted by using an algorithm that keeps only HTML tags.

Our methodology distinguishes this casuistry and applies a custom heuristic-based system that we describe next.

Hybrid hashing. To assess if two pages are mirrors, we apply a combined approach, using two different hash functions, as depicted in Figure 3. We first rely on the *cryptographic hashing* algorithm MD5 [34] to determine whether two pages are *exact* copies. This has a binary outcome, i.e., the pages are equal or not. In case it is not equal, we use a *fuzzy hashing* function that scores how *similar* two pages are, with a certain tolerance for changes. Our implementation uses the CTPH algorithm [23], which breaks down pages into several components and calculates a fuzzy hash for each one having as a result a final hash combining them all.

Same language. The next step is to determine if two non-equal pages are in the same language by comparing the language information we obtain in the data extraction step (see Section II-B). If the language is different, we only compare the scheme structure of the page (since the content inherently differs). When the language is the same, we rely on the output of the fuzzy hashing algorithm applied to the whole site (i.e., the entire HTML file). If we see that this HTML is the same, we determine that the two pages are mirrors. Instead, if the HTML files of the pages are different, we perform a further step.

Same language and different HTML. We then apply a method to capture inconspicuous modifications. These modifications may be attributed to changes in the products offered in the same market across the period of the crawl of the mirrors. Thus, we distinguish whether these differences stem from small changes in the scheme of the HTML (e.g., the CCS style), or in the content.

As there are different types of reasons justifying inconspicuous modifications, we use a modular approach that can be fine-tuned to different scenarios. For this, we compare the pages using a weighted sum of the output of the fuzzy hashing algorithm for their *schemes* (only the HTML tags), and for their *content* (only the text). In particular, we apply two weights to the content and the scheme of a page (denoted as W_c and W_s respectively in Figure 3) that capture the relative importance each of these two elements has in the overall similarity measure. The value for the weights and the threshold are established by empirical evaluation of the effectiveness of the algorithm to detect mirrors. Our implementation also empirically identifies that a similarity measure above 90% implies that two pages are the same. When the similarity is below this threshold, we consider that the two pages have significant differences.

We detail empirical measures in Section III, but we note here an important methodological step. We validate mirror detection using a custom tool that renders the HTML (without retrieving the media content, i.e., only the text). This lets us open the page in a browser and visualize potentially sensitive or even illegal content.
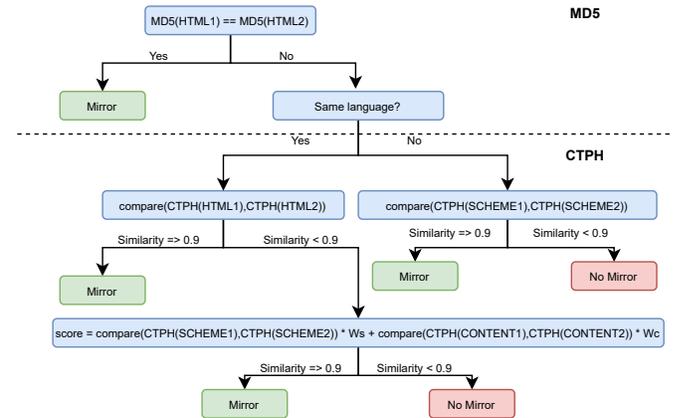

Figure 3: Workflow of the mirror detection algorithm. The output provides a similarity score (1 being the most similar).

D. Network Analysis

To analyze the topology of Tor we use graph theory. For this, we first create a graph that models the Page Linking relationship described in Section II-B. In the graph, nodes represent singular pages and edges the relationship between pages (i.e., the pages are linked to each other). We next detail how we build and analyze the graph.

First, we build this graph ensuring that all links between the pages are added and linked using the same path as the crawling process. From the seeds to the endpoint pages, each site is connected based on the links referred between them.

When a new page is crawled, the list of links is updated accordingly in the graph. Note that mirrors are not added to the graph to avoid redundant pages. However, we account for this information by annotating the page node with the mirrors it has. Several other features are added to each node of the graph to enrich the analysis: 1) title of the page, 2) language, 3) surface connection (boolean), 4) depth (i.e., number of

pages to reach a given one), 5) timestamp, 6) category, 7) mirror existence (boolean), 8) mirrors URL's. Table II summarizes the features we use to annotate the nodes.

Finally, through a reachability analysis, we compare the segment of the network we reach through the different seeds (e.g., OSINT or underground forum sources). This lets us assess how effective the automatic seeding method is when compared to the manual one and how informative the different keywords extracted from underground forums are.

Table II: Features of a node in the Pages Graph.

Feature	Description
Title	The title of the page if it exists
Language	For analysis purposes related to statistics. Indicates
Surface	if the page is connected to the Surface Web or not
Depth	Minimum path to reach this page from the seed
Timestamp	List of timestamps where the page was crawled/attempted
Category	The type of content offered by the site
Has_Mirror	Boolean value indicating whether a given site has mirrors
Mirrors	List of mirrors that have not been included in the network

E. Content Analysis

One of the main goals of our study is to determine whether the type of content of the hidden services reached with Mimir is related to the seeds provided as input. This is of great interest to researchers (e.g., to understand the types of content being delivered in anonymous services) and to Law Enforcement investigators (e.g., to understand new forms of cybercrime or to identify sites trading deviant content). To this end, we build a probabilistic model using Logistic Regression (LG). We choose LG due to its performance in other text-based learning tasks in comparison with other well-known algorithms that are good in this field as Naive Bayes, Random Forest, Decision Tree, or Support Vector Machines [32]. To corroborate this, we explored the use of more complex algorithms (see Section §III-E), obtaining similar accuracy at a higher latency. We use the categories in the Duta-10K dataset [2] as ground-truth and build a multi-class classification algorithm.

Duta-10K offers about 10K .onion sites labeled with 25 different categories that range from "Counterfeit" or "Drugs" to "Market" or "Hacking". The dataset was initially collected in 2017, and consequently, most of the 10K sites are no longer reachable. However, we got access to a large subset of raw HTML files from the authors of the dataset. We observe that some of the categories were underrepresented, and thus we rely on the 11 (out of the 25) categories. During training, we limited the number of samples to 200 per category to achieve a balanced dataset, using 200 samples when available or fewer when a category had less. This threshold was experimentally established to avoid biases from under-represented categories. We excluded two types of categories: first, those with low samples count (e.g., Art, with 14 samples or Politics, with 2 samples), which were excluded due to insufficient support for consistent classification; and second, categories with a larger number of samples whose content overlaps with that of other categories, such as Personal (417 samples) and Services (284 samples). Additionally, we removed the Empty category (1,350 samples), as it provided no useful content. Also, due to the
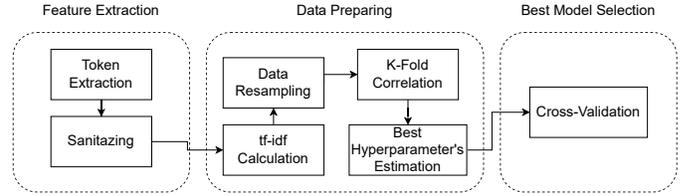

Figure 4: Content classification pipeline.

nature of the ground truth, we limit the content analysis to only English sites. In total, we removed 2,314 samples (26.8%) and retained 6,321 samples (73.2%).

The training process of the content classifier is divided into three tasks as shown in Figure 4: Feature Extraction, Data Preprocessing, and Best Model Selection.

1) **Feature Extraction**. To extract features from the raw documents, the HTML tags are removed and the resulting content is turned into tokens. These tokens are then filtered by removing stop words, and the remaining tokens are lemmatized to convert each word to its inflected form or *"lemma"*.

2) **Data Preprocessing**. This process first calculates the Term Frequency-Inverse Document Frequency (tf-idf), which expresses the relevance of a word in a document from a set of documents, as defined by the following equation:

$$\mathtt{tf\text{-}idf} = \frac{freq(term_n, document_m)}{\log_2(\frac{\#documents}{\#documents\_with\_term_n + 1})}. \quad (1)$$

The tf-idf is calculated using a Sublinear Term Frequency Scaling. This prevents repeated words from getting unnecessary relevancy, e.g., due to the existence of a list in the text or lack of vocabulary in the document. This means that the frequency is not natural but logarithmic. Then, we use bootstrapping to balance the training set. Finally, we shuffle the data and perform a division of the entire data set by using k-fold with k = 10.

3) **Best Model Selection**. To estimate the optimal parameters of the Logistic Regression model, we perform a grid search over the parameters using a fraction of the training set and measure the penalty and the regulation strength:

  • For the penalty, we take into account Lasso regularization (l1) which makes variable coefficients go to 0 in case they do not contribute as others do, ridge regression (l2) that decreases those coefficients but never lets them arrive at 0, elasticnet where l1 and l2 are added, and also we have check models without any penalty.

  • We also use the regulation strength (C) to determine the impact of the penalty on the coefficients, the stronger the regulation, the smaller the coefficients.

The three steps above are critical to find a proper trade-off between obtaining a general model or a fine-tuned model tailored to our training set. A poor regularization will make the model take into account non-relevant variables, whereas a too-strong regulation may weaken the model.

To assess the influence that the different parameters have over the performance of the model, we leverage the F_{beta} score (with $beta$ equal to 1), defined by the following equation:

$$F_\beta = \frac{(1 + \beta^2) * Precision * Recall}{\beta^2 * Precision + Recall}. \tag{2}$$

Our best setting uses a logistic regression with an *all vs all* strategy.

F. Ethics

This research requires data collection from public sites. The researchers have experience dealing with the crawling of online underground communities hosted both on the surface and the Dark Web. We follow due precautions to minimize potential risks and harm to online users. First, the information we gather is obtained from public sources, which we reach by means of automatic navigation. The crawler is instrumented to gather textual content, and thus, we mitigate the risk of downloading illegal material such as indecent images of minors. Also, our analysis does not target specific groups or individuals and is conducted over aggregated data. Data is stored in an encrypted disk on our servers, and it is only accessible to the researchers involved in this project. Finally, following high ethical and legal standards, every evidence of illegal activities, especially those related to child pornography, is reported to the authorities. Overall, the benefits of our research outweigh the risks. We have obtained approval from our IRB and carefully considered Tor safety guidelines [33].

III. ANALYSIS

In this section, we report our findings as a result of applying the methodology described above. First, we describe our experimental setup, including the description of the dataset collected. Second, we present the network analysis, and finally, the content analysis.

A. Experimental Setup

The dataset we use in this paper has been collected for a period of 8 months. We use a server with 24 cores and 47 GB of RAM. The crawler runs uninterruptedly except for some infrequent events (e.g., due to electricity cuts or server overloads).

The crawl starts after feeding the seeds to Mimir. We collect nearly 7k seeds, out of which about 2% stem from manual seeds (using OSINT searches), and the remaining are seeds collected automatically (using systematic queries to TSE using keywords from underground surface forums). We see that 83% of the `.onion` addresses in the initial set of seeds are unreachable. This highlights that the search engines and OSINT might provide outdated results, and motivates the use of online collection tools such as Mimir. As a result, we discard unreachable `.onion` addresses and start the crawling process by visiting 1,157 seeds (17% of the initial set).

Departing from the initial seeding, we reach 24.9k different sites. Table III offers a characterization of our datasets. We note also that the crawling workload grows almost exponentially as most of the content of the site includes a link to other sites.

At the end of the 8-month crawl, the crawler has made a connection attempt to 34% of the sites (5,002) in the TODOLIST. Likewise, we see that the crawler has gone through more attempts for a smaller proportion of URLs. Once the number of attempts reaches the maximum value set (i.e., 5), those URLs are appended to the *Unreachable* list. In total, around 8K (21.4%) of the URLs we attempt to connect to are currently unreachable, the large majority stemming from the set of living seeds.

Table III: Dataset characterization. *Seeds* and *sites* (left to right) reached from the seeds.

Seeds		Sites attempted/reached			
6,816		Accessible	24,911	Base	4,121
				Mirrors	20,790
Accessible	Unreach.	Unreachable	8,381 (sites)		
		4 attempts	0		
1,157	5,659 (seeds)	3 attempts	5		
		2 attempts	796		
		1 attempts	5,002		
		Total	39,095		

B. Network analysis

We first analyze the network topology. Concretely, we study the connectivity of the network, looking for Dark Web "bubbles", namely subgraphs — i.e., a set of sites directly or indirectly connected to each other while being unconnected to the others.

In particular, we see 1,040 subgraphs inside the network. We find that 99 of the subgraphs have 10 or fewer nodes, which shows that the Dark Web is highly fragmented and "bubbles" generally have a small size. Instead, we see that a small number of subgraphs have a large number of nodes that are all deeply connected. This shows that the topology of the Dark Web follows a power-law distribution much like the Surface on the advent of the Web [15]. Table IV shows the top 5 subgraphs judging by the number of nodes, together with the number of edges. We also show how many nodes have direct access to the surface web. All the subgraphs are weakly connected (i.e., not all pages link to all other pages).

An interesting finding is that the largest subgraph represents $\approx$66% of the total share of Tor that we reach, being the Top 5 at 69% of the network. Also, we observe that 9% of the sites are directly connected from links on the surface. Our crawler only visits surface links when they are given as seeds. This means that our systematic extraction of seeds lets us reach 1K subgraphs, with our crawler discovering most (91%) of the Dark Web.

Subgraphs of order 1 (i.e., just one node) represent the seeds that are unreachable. We can not confirm that these seeds will

Table IV: Size of the network and its top 3 largest subgraphs. #LS is the number of nodes linked from the surface web.

	#Nodes	#Edges	#LS
Full network (1k subgraphs)	4,121	3,360	412
Subgraph 1	2,757	3,011	411
Subgraph 2	43	50	0
Subgraph 3	27	27	0

become available at a later stage, the same way that it is hard to get a static snapshot of the network due to the volatility of Tor. However, our results in terms of the coverage show that our methodology produces a more effective mapping of the Dark Web when compared to prior work (see Section VI for a detailed comparison with the related work).

C. Contribution of the seeding process

We measure how our systematic seeding process improves the coverage of the network in terms of nodes found (i.e., onion sites). For this, we rely on the network analysis described above and perform an ablation study of the sources that seed our crawler. Recall that our seeding process stems from two sources: queries in Tor-focused search engines using keywords systematically extracted from underground surface forums and manual entries obtained from OSINT.

We initially look at the `.onion` sites we reach from the manual seeding and take this set as a baseline. We then study the contribution that a keyword has to the coverage of the crawl over the baseline. For this, we first extract the subgraph of Tor sites that we visit because of all the manual seeds, namely Manual Seeds Subgraph (MSS). We then extract the subgraph that stems from each keyword seed k and name it the Keyword Subgraph (KS_k). We finally compute the set difference (SD) as $SD = KS - MSS$, which retains only novel *.onion* sites attributed to the keyword k, i.e., we remove all sites we reach through manual seeding (including the intersection).

Table V shows the top 10 keywords that contribute the most to our crawl of the Dark Web. We observe that a single keyword like "drugs" or "hosting" leads the crawler to reach as much as 10.05% and 9.29% respectively of the total gathered pages. This shows that crawling TSE with targeted keywords allows Mimir to obtain sizable improvements in terms of coverage when compared to the manual seeds. Table V provides the contribution of each keyword individually with respect to the MSS. Since there is an overlap in the contributions of various keywords, the percentages in Table V are non-accumulative.

To understand the contribution of all the keywords as a whole, we repeat this process with $k = k_1 \cup k_2 \cup ...k_n$, where n is the total number of keywords. We denote this subgraph as All-Keywords Subgraph (AKS). As shown in the bottom row of Table V, the contribution to the coverage of all keywords together over the manual baseline is 81.85%. In practice, this means that the number of `.onion` sites reached increases a 177.55% when compared to manual seeding. Furthermore, the set difference between AKS and MSS is the empty set. This means that MSS is a subset of AKS, and thus, all the `.onion` sites we reach using the manual seeding method are covered by the automatic seeding approach. Plus, the automatic seeding approach covers a significantly larger portion of the Dark Web. All in all, *this shows that our proposed automatic approach is an effective mechanism to avoid the cumbersome task of manually looking and entering a set of initial seeds or keywords.* As all prior works require manual seeding, we see how our crawling strategy offers a competitive advantage in terms of scalability.

Table V: Individual systematic seeding keyword nodes contribution per category.

	drugs	free	hosting	software	hacking	forum	carding	counter	services	service	TCC
Counterfeit	2.69%	1.70%	2.14%	1.70%	2.06%	2.35%	2.06%	2.26%	1.63%	1.33%	19.92%
Crypto	1.75%	2.48%	1.55%	1.99%	1.46%	1.50%	1.63%	1.43%	0.51%	1.09%	15.38%
Down	0.73%	0.61%	0.63%	0.73%	0.44%	0.41%	0.58%	0.27%	0.46%	0.46%	5.11%
Drugs	1.16%	0.53%	0.49%	0.58%	0.41%	0.41%	0.44%	0.39%	0.34%	0.39%	5.14%
Forum	0.29%	0.46%	0.41%	0.41%	0.36%	0.66%	0.05%	0.34%	0.32%	0.36%	3.66%
Hacking	0.49%	0.53%	0.68%	0.68%	1.24%	0.56%	0.49%	0.46%	0.78%	0.68%	6.58%
Hosting	0.95%	1.29%	1.89%	1.12%	1.04%	0.99%	0.56%	0.41%	1.41%	1.12%	10.77%
Locked	0.78%	0.56%	0.41%	0.49%	0.39%	0.53%	0.66%	0.29%	0.39%	0.41%	4.90%
Market	0.70%	0.49%	0.39%	0.39%	0.34%	0.41%	0.34%	0.46%	0.53%	0.39%	4.44%
Porn	0.12%	0.78%	0.12%	0.10%	0.22%	0.12%	0.05%	0.39%	0.12%	0.22%	2.23%
SN	0.39%	0.44%	0.58%	0.53%	0.39%	0.27%	0.15%	0.24%	0.27%	0.24%	3.49%
AKS	10.05%	9.85%	9.29%	8.71%	8.35%	8.23%	6.99%	6.94%	6.75%	6.70%	81.85%

Similarly, we analyze how influential keywords contribute to the crawling of the different categories within the network. The rows in Table V indicate the sub-graph of MSS considering only nodes of the given category (the last row being the full AKS graph). It can be observed that all the keywords primarily improve reachability to Counterfeit services, which is the most prevalent category in our dataset. After that, we observe interesting patterns since each individual keyword reaches the specific category related to it (e.g., drugs or hosting).

While a keyword is not confined to influencing only its related category, it exerts the greatest impact within it. Note that multiple keywords can contribute to the same node.

D. Mirror analysis

As Section III-B analyzes the topology of singular pages in the Dark Web, we next evaluate our method to detect mirrors and provide a characterization. First, we perform an empirical analysis to set the scores W_s and W_c (introduced in Section 3) and we establish $W_s = 0.3$ and $W_c = 0.7$.

We manually analyze English pages with at least one mirror (450 and 53.38% in total) using a custom tool to render HTML sites without media content, thus preventing the exposition of the validator to indecent content. During this step, we remove some (43) sites for several reasons, including wrong language detection and isolated problems rendering some of the HTML with the tool we develop for the validation. All in all, we validate 408 (48%) sites, of which only 22 are miss-classified, leading to a precision of 97%. Out of the 48% sites, our algorithm detects 801 mirrors (355 identified with CTPH and 424 with MD5). Since hash match means that pages are exactly the same, we only need to manually evaluate the remaining 355.

Our crawler collects a total of 24,911 sites, out of which 83% (20,790) are mirrors, 16% (4,008) are unique `.onion` sites and 0.4% (113) are 'unique' surface links wrongly provided as seeds at the beginning of the crawling process. Note that these unique sites are merely representatives of their respective mirrors. Thus, we define a unique site as the first intance of replicated site that we crawled, being the rest the mirrors. Note that the large majority of the links we see in our dataset are mirrors. Only 19% of the unique sites have mirrors, and most of them (54%) have only a single pair-wise match. There are also 11% with 2 mirrors and 6% with 3. The remaining 29% have four mirrors or more. Interestingly, we spot 57 sites with more than 100 mirrors, with 2 sites having 1,126 and 1,203 mirrors respectively.

More than half of the mirrors are exact copies (54.4%), i.e., they have the same MD5 hash. The remainder (45.6%) contains some minor modifications. Understanding the exact

reasons for mirroring would require dedicated tools, and it is out of the scope of this paper. However, during our empirical evaluation of the mirror detection algorithm, we carried out manual validations over samples, which allow us to understand the differences. Most of the changes among the mirrors are minor and involve either the content, such as titles, or the HTML schema, such as updates in the design framework. Accordingly, we analyze two types of changes: one for easy-comparing content fields and another to analyze the modifications in the raw HTML.

We first run an automated analysis over sites detected as mirrors with CTPH to count changes in the language and also identify information (i.e., cryptowallet addresses and email addresses). We find 15 pages in English that are also mirrored in other languages, mostly Italian, German, and French. There are 1,409 mirrors that have modifications in at least 1 Bitcoin address and one of them with at least one different Monero address. Regarding emails, there are 1,486 mirrors with at least one different email from one another. We hypothesize that minor changes may occur due to various reasons. First, some modifications may have occurred within the time lapse between the crawls of these mirrored sites. Second, it is possible that an unauthorized entity impersonated the legitimate site with malicious intent, potentially engaging in phishing or scams [5]. And third, in cases where mirrors are legitimately managed by the same operator, there might not be an automatic synchronization process, causing some pages to remain outdated until the administrator updates them (if any). However, confirming these hypotheses is left for future work.

Second, we analyze differences in raw HTML files flagged as mirrors. Specifically, we manually analyze 216 pages in English that have at least one mirror, together with their corresponding 335 CTPH-detected mirrors. The main modifications are seen in the HTML structure (content not included) as depicted in Figure 5. The second most frequent difference relates to pages with changes in links of the site, followed by content modifications (only the text of the site). In some sites dedicated to trading, we see that the physical currency (e.g., dollar or euro) remains the same, but the equivalent price in currencies changes. We also see some pages with modifications in their cryptowallets. We see some sites generating one address per each purchase, and such changes should be studied carefully. Finally, we only see a few pages with modifications in the physical currency (FIAT money). Considering that pages are often crawled on different days, we see that some of the sites do not propagate changes to all mirrors automatically.

The `.onion` addresses can be separated by versions, being their difference in length: v2 addresses are 16 characters, while v3 is 56 characters.[1] In the 4,008 unique `.onion` sites, there are 1,061 which are v2 (26.4%) and 2,795 v3 (69.7%). There are also 152 URLs (3.8%) that do not belong to either v2 or v3 and correspond to malformed URLs that, while resembling .onion addresses, link to a surface website. As for mirrors, there are 1,065 which are v2 (5.1%) and 19,680 v3 (94.9%), i.e., a larger proportion of the mirrors belong to the newest
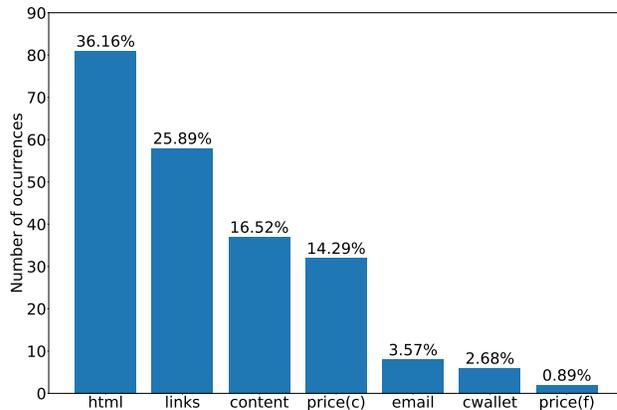

Figure 5: Minor modifications in the HTML of mirrored pages.

version.

Regarding connectivity, even though only 9% (440) has direct access *from* the surface (according to our initial seeds), 44% of the sites reference a link *to* the surface web. The longest path from a seed to a site crawled, i.e., the maximum depth, is 6. However, this is just one exceptional case, and most of the pages are spread between the depths of 1 to 5. Specifically, there are 28% (1,157) sites of depth 0 (i.e., seeds), 17% (706) sites of depth 1, 8.2% (340) sites of depth 2, 19.5% (803) sites of depth 3, 21.8% (902) sites of depth 4, (5.1%) 212 of depth 5, and the one mentioned site in depth 6 (<1%).

Mirror detection benchmark. We build up a benchmark for comparing our algorithm with two of the most well-known state-of-the-art approaches, namely MinHash [9] and SimHash. We first determine the optimal thresholds for each algorithm to conduct a fair benchmarking. To this end, we randomly choose 2,000 samples: 1,000 real unique-mirror pairs and 1,000 pairs of pages that are not mirrors. We then calculate the number of correctly detected mirrors and non-mirrors for both methods across 10 different thresholds, ranging from 0 to 1 in increments of 0.1. This experiment indicates the optimal threshold score for each method, i.e., 0.8 for SimHash, and 0.4 for MinHash.

Table VI: Duplication Algorithms Benchmark

Method	TP	FP	FN	Precision	Recall	F1-Score	Repetitions
Mimir	10,498	9	3	0.99	0.99	0.99	0
SimHash	10,495	15,440	6	0.40	0.99	0.57	15,336
MinHash	10,499	6,634	2	0.61	0.99	0.75	6,579

Table VI summarizes the results of the benchmark. The three methods obtain similar True Positive (TP) and Recall rates. However, a key issue arises with the False Positive (FP) rate. SimHash and MinHash frequently identify the same site as a mirror for multiple representative pages, significantly reducing their precision. In contrast, Mimir stands out for its robustness in avoiding this issue, demonstrating greater resistance to incorrectly matching the same page as a mirror for different representatives.

Overall Mimir is particularly effective for measuring HTML document similarity. While SimHash and MinHash are good approaches, they have limitations with non-nearly identical pages. SimHash uses hamming distance for detecting near-duplicate detection small changes that may affect the similarity

[1] We note that, by the time our crawling started, v2 .onion addresses were not yet deprecated.

score. MinHash's reliance on text tokenization may introduce excessive noise due to HTML redundancy, reducing its effectiveness when using Jaccard similarity as a metric.

E. Content analysis

One of our key goals is to analyze the content hosted in the Tor network in order to localize cybercrime-related HS. For such purpose, we first look at the language used. Then, we categorize the content, and finally, we detect the mirrors. Overall, there are 44 languages. We see that 88% of the pages are written in English. The remaining 12% are widespread in terms of language since each one accounts for less than 1%. These results are consistent with previous studies on the Tor network [4], and confirm that most of the content is still offered in English.

Model Benchmark: For comparison, we trained various machine learning models using the same training set. Apart from Logistic Regression (LR), we select state-of-the-art algorithms for text classification, i.e., Convolutional Neural Networks (CNN), Bidirectional Encoder Representations from Transformers (BERT), and a recent variant focused on Dark Web content, DarkBERT [20].

Table VII: Duplication Algorithms Benchmark

Model	Accuracy	Precision	Recall	F1	F2	Time
LG	0.863	0.757	0.864	0.798	0.833	0.0169
CNN	0.834	0.699	0.690	0.691	0.689	1.2637
BERT	0.847	0.682	0.697	0.685	0.690	15.3399
DarkBERT	0.862	0.704	0.712	0.704	0.708	46.0361

Table VII shows the results of our benchmark experiment. The CNN, BERT, and DarkBERT achieve similar accuracy to that of LG. However, the three models have suboptimal precision and recall. Additionally, the time required to classify the test set increases significantly in comparison with Logistic Regression. Simple models like Logistic Regression and Naïve Bayes often outperform complex models like BERT or CNNs in text classification when data is limited. These traditional models, relying on hand-crafted features like TF-IDF and word n-grams, excel in specific tasks without needing long-term dependency understanding [11], [3], [25]. In contrast, deep models tend to overfit without large datasets, learning irrelevant patterns or missing the correct ones. Overall, for our specific task and dataset size, Logistic Regression outperforms the rest of the models.

Categories. We built a custom ML classifier (see Section II-E). The training has been performed with the set of labeled pages from the Duta-10K dataset [2]. From the total list of 10,367 URLs, only 1,099 (10%) were still available, leading to an imbalanced dataset. After balancing the data and removing under-represented categories, the final training set accounts for 2,200 sites and 11 categories (200 samples per category). The final test set contains 5,628 pages, offering an accuracy of 86% (75% precision and 86% recall). Overall, we get 79% F1 and 83% F2 scores.

The trained model is used to classify all sites. In addition to that, we also measure the reliability of the classifier by using conformal evaluation to calculate the percentage of

the unreliability of a given classification, i.e., the difference between the most probable class and the second one, which can be expressed as: $1 - (P(C_{1^{st}}) - P(C_{2^{nd}}))$.

According to this metric, we find that the reliability of more than half (52.52%) of the pages was higher than 90%, with $\approx21\%$ being reliable at 99%. Only 8.6% of the pages were below the 10% of reliability, followed by a $\approx12\%$ that are between 11% and 20%. Therefore, it shows the classifier is not only efficient but also reliable.

Using the trained model, we infer the category for all the English sites in our dataset. Figure 6 shows the resulting categories. For a better understanding, these are described in Table IX. We indicate which categories might be related to deviant activities (e.g., Drugs or Locked). We note, however, that some of the others can also contain cybercrime-related content, e.g., crypto, market, forum, or porn (as we show in §IV). We see that the predominant class is "Counterfeit", which mainly aims to sell stolen credit cards. It is followed by the "Hosting" category which is related to Servers, File Sharing, and Links Directories. The third category with more matches is "Porn" which provides explicit sexual content. In total, we see 1,571 (38.12%) unique pages whose categories inherently define deviant content (i.e., Counterfeit, Drugs, Hacking, and Locked). However, we see 18,474 (74.4%) if we take into account the mirrors. Table VIII offers a breakdown of the prevalence of cybercrime in our dataset per category. We note that these figures only offer a lower bound as other HS may have deviant content also (e.g., "Porn" or "Markets").

Table VIII: Detailed cybercrime related sites numbers.

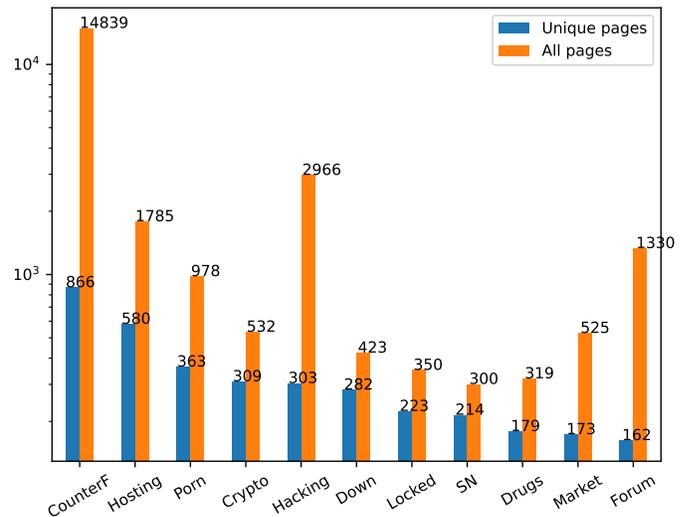|	Counterfeit	Drugs	Hacking	Locked	TOTAL
Unique	866 (55.2%)	179 (11.4%)	303 (19.3%)	223 (14.2%)	1,571 (38.1%)
Mirrors	13,972 (82.9%)	132 (0.8%)	2,652 (15.7%)	84 (0.5%)	16,840 (81.3%)
TOTAL	14,839 (80.3%)	319 (1.7%)	2,966 (16.1%)	350 (1.9%)	18,474 (74.4%)

Figure 6: Histogram representing the distribution of sites per categories with and without filtering out mirrors.

Mirrors. Figure 6 shows with the orange bar the number of pages (including mirrors) per category. If we were not to

Table IX: Classifier categories description, with (*) denoting cybercrime-related activities.

CATEGORY	DESCRIPTION
Counterfeit*	Carding (i.e. credit card), Fakes ID and Money sales.
Crypto	Cryptocurrency buying, selling and mining
Drugs*	Drug marketplaces or other related sites
Forum	Discussion forums of any type.
Hacking*	Hacking services or pages related to hacking.
Locked*	Log in, sites closed by the authorities and so on.
Down	Pages that are not available anymore.
Market	Different marketplaces as virtual items, weapons, pharmacy...
Porn	Pages related to pornography.
Soc.-Network (SN)	Blog, Chat, Email.
Hosting	File-Sharing, Folders, Search-Engine, Server, Directory.

filter mirrors out, we may observe some important differences. First, some classes are really over-represented as "Counterfeit" which the number of pages appears to be 1,713% higher if we do not identify which of those sites are mirrors. Second, if we want to understand which content represents the cybercrime in the Tor Dark Net better at some point in time, this over-representation may lead to misunderstanding the real picture of the network. For example, the category "Hacking" may look like it is the second service more extended in the network, but in reality, it holds the 5th position, and which is more interesting as well as "Forum" which seems to be fourth when actually is the last one. This shows that accounting for mirrors when analyzing the Dark Web is important to avoid biases in the interpretation of the results, and it can also help law enforcement to make diligences more effective. For example, in the case of Counterfeit, only 5.8% of the sites require in-depth inspection (i.e., the unique ones), 10% in the case of Hacking sites, or 56.11% in terms of Drugs. n Our results also open new research questions on the type of pages that are being mirrored, and for what purpose. A deeper analysis looking at the raw HTML of the most mirrored Top-10 unique pages shows that 70% of them were "Counterfeit".

IV. USE CASE: DISTRIBUTION OF CHILD ABUSE MATERIAL

A staggering type of abusive content that finds shelter in the Dark Web is the distribution of child abuse material. We next see how our contribution has been helpful in the process of finding child abuse-related `.onion` sites, all of which have been reported to our national Law Enforcement contact.

During the manual validation in Section III-D, we identified 20 sites hosting child abuse content from the samples categorized as Porn using our ML classifier. We note that our manual inspection is restricted to text only. Our crawler does not download media content, and thus our manual validation went only through the text we extract offline from the HTML files we collected, and never through images or videos. The size of this set (20 sites) is too small to include child abuse as a category of the ML classifier used for content analysis (c.f. Section III-E). Thus, we developed a custom keyword search algorithm for this. Our algorithm flags `.onion` sites that contain child abuse-related keywords. We extract these keywords from the 20 manually detected child abuse sites that capture the jargon used by pedophiles and that it is exclusive to child abuse. For ethical reasons, we do not disclose such keywords since these would facilitate miscreants to reach these pages using a keyword search with Tor search engines.

Many forums and collaborative sites on the Dark Web explicitly prohibit child abuse content. In general, site administrators and moderators that forbid this type of content use short and clear statements. We identify these cases using formal language modeling and use it to avoid false positives. In particular, we model every sentence and extract the structures for adjectives and verbs. We then model negative words around our keywords or explicit forbids following the structure we define in Table X.

We use four general rules to model adjectives and verbs around the keywords of interest. Sentences can be positive or negative. In the case of verbs, negative means that they express prohibition, and positive verbs express the opposite. As prohibition can be manifested as positive and negative sentences, we model both cases and account for the meaning given by the adjective/verb as well as for negative abbreviated forms like *"n't"*.

Table X: Structures to detect child abuse content allowance. KEY: Keywords, VBZ: Infinitive verb, VBN: Past participle verb, JJ: Adjective, WILL, NOT/N'T: Specific words. The symbols "+" and "-" denote positive and negative words, with verbs indicating allowance or prohibition.

Structure	Example
KEY + VBZ + NOT/N'T + (+)VBN	[keyword] is not allowed
KEY + WILL + VB + (-)VBN	[keyword] will be removed
KEY + VBZ + NOT/N'T + (+)JJ	[keyword] is not welcome
KEY + VBZ + (-)VBN	[keyword] is censored
KEY + VBZ + (-)JJ	[keyword] is forbidden

We apply the algorithm to the sites that have been classified in the category 'Porn', and find 180 pages potentially distributing child abuse material. We perform a validation of all 180 hits (again, only judging the text), and confirm that 165 unique sites (91,66% accuracy) claim to allow the distribution of indecent images or videos with minors. We attribute the errors to some expressions that were not modeled and to the lack of some keywords in the child pornography lingo. Notably, after the validation process, we examined additional mirror sites identified by Mimir, increasing the detection count to 505 pages (a 306% increase).

Whereas we rely on a dataset collected up until April'22 (c.f., Section III-A), we observe that 78 of these sites (49%) were operative in December'23, and that 45 are still operative by March'25 — at the time of this writing. Alarmingly, we found a hidden service (which was mirrored in 4 .onion sites) that provides interactive pornographic content with minors through live cams which, if confirmed, would pose systematic aggression and possible child trafficking. Unfortunately, this site is still online in 2 out of the 4 mirrored sites in March'25, almost three years after we first observed it. This suggest that, even if Law Enforcement is able to take down some of these sites, the service remain operative as it is mirrored in other sites. This highlights the importance of considering mirrors when analysing the Dark Web. Accordingly, all these sites have been reported to our national Law Enforcement contact point and are currently under investigation. Preliminary feedback confirmed that many of them actually offer illegal content.

V. Discussions

This work provides new insights into the Tor Network. In this section, we discuss the main limitations of our work together with key takeaways.

A. Limitations

Measuring the topology and contents of Tor is challenging and requires dedicated tools that have limitations.

Coverage. Our analysis is bounded by the coverage of our crawler. This is a limitation prevalent in previous work, although our crawling methodology is more exhaustive. In particular, we crawl 233% more reachable sites than M. Bernaschi et al. [6] for as twice as long (see §VI).

The novel systematic extraction of seeds offers a key advantage with respect to prior work in terms of scalability, which translate into larger coverage without the need for manual seeds crafted by experts.

We note that our dataset represents a static snapshot of the network (i.e., the one we got in our crawling period). Having dynamic information requires living systems to continuously conduct crawls, a capability that the proposed crawler Mimir contains. However, we opted to stop the crawling after one snapshot was retrieved following Ethical advice. Our goal is to provide a singular content and network characterization of Tor, and not study its temporal evolution. Thus, our research effort focuses on performing a single crawl and does not require a production system that performs a continuous crawl. This way we avoid increasing the overall workload on the Tor network.

Ground-truth.

Another important limitation stems from the lack of ground truth to build a content classifier.

Our work relies on the annotations made as part of the Duta project [1], [2].

However, some categories were underrepresented in the data set and we thus remove them. We showed that for the categories we retain we are able to obtain reliable performance. The lack of good quality ground truth is a prevalent problem in cybersecurity.

As part of our future work, we aim to expand the number of categories used in content characterization. However, manually labeling content is both time-consuming and raises important ethical and legal concerns. It is important to note that content analysis is not a primary contribution of this paper. Rather, it is used to validate the effectiveness of Mimir in discovering pages related to the provided keywords. In fact, since Mimir is a modular architecture, integrating other machine learning models is straightforward.

Rich- and media-content. Our work primarily uses text to identify mirrors and classify the content of the sites. Ethical and legal concerns have prevented us from downloading multimedia content from Dark Web sites that may contain deviant content (e.g., child abuse material, actionable exploits for de-anonymization[12], or malware [37]). We leave for future work the use of ethical and legal methodologies to analyze this content without harming researchers, e.g., automatic image processing. Different from the clearnet [29], our study entails a higher risk of downloading illegal content, which brings

unique challenges that require careful consideration and active cooperation with law enforcement agencies for this task.

Landing page scope. While vertical crawling may offer more in-depth insights, since our main goal is to discover as many hidden services as possible, we set a horizontal crawling approach, not only to limit the cost in time but also the bandwidth exposed to the Tor network, which is quite important when using the Tor protocol. Our results in category classification and mirror assessment have shown that data beyond landing pages is not essential in this task except for cases where the landing page is a login page or has a Captcha due to the limited textual content available for classification and the structural similarity of the data. Such cases require tailored efforts, which are out of our scope. Furthermore, our ethics protocol steems us to sign in to sites that may offer child abuse content, which limits our ability to further explore this conclusion. We see that there are not many such cases, two (0.48%) captchas and four (0.97%) cases with a Login form. Vertical crawling would introduce an important overhead to the Tor network, jeopardizing the objectives of this work.

Mirror detection algorithm and copycat sites. We correctly identify the 97% of Mirrors. The 3% misclassification is primarily due to two reasons: small HTML files with minimal content and sites that use the same framework but in different configurations (e.g., a personal email server). However, differentiating between original sources and copycat mirrors is challenging, especially given Tor's anonymity features. Minor differences between mirrors are insufficient for attribution. Reliable attribution requires a deeper analysis beyond the scope of our contribution.

B. Key Findings

While our study presents some limitations as discussed above, we offer a unique analysis of singular services on the Dark Web, with a special focus on the impact of duplicate content in the network.

Our findings constitute a large longitudinal crawl that requires a comparatively small number of seeds. We next summarize the key takeaways of our work and discuss their implications for research.

By leveraging an innovative method for seeding and crawling, we were able to reach a larger proportion of the Tor Network. Using trending topics in underground surface forums is a promising direction when seeking deviant content in Tor. This method reaches over twice the number of onion sites compared to manual seeding without missing any. Furthermore, by making our crawler aware of the network context, we reduce the impact of Tor's volatility, leading to further increased page reach. As a result, Mimir significantly reduces the time required to reach hidden services. Additionally, our horizontal crawling strategy shows better network coverage with a higher number of sites per URL visited compared to prior work. In particular, Mimir has discovered 25k pages from 1k seeds, achieving an amplification factor of 2.2.

Our analysis confirms that Tor needs to be studied as a dynamic network, due to the high volatility of its content. Depending on the crawling period (and even the time when a

page is visited), the connectivity patterns change. Our crawler makes an important contribution to the study of Tor. We can not claim robustness on our network analyses since we ignore whether the connections observed during our crawling period will remain available in the near future. However, our crawling methodology is an important stepping stone toward understanding the dynamics of the Dark Web, exchanging the limitation that this entails for an advantage.

Mirror matter when analyzing `.onion` sites content. A remarkable takeaway from our study, achieved by the proposed detection algorithm, is the amount of replicated content (mirrors) present in Tor. Around 82% of the landing pages accessed through our crawler are replicated content. We emphasize that this figure includes exact copies, and also pages with minor modifications in the landing pages. Our results comparing Hacking against Porn or Cryptocurrencies pages show the limitations after not considering mirrors. Without mirror detection, hacking-related pages appear as the most prevalent category (inflating their importance over personal-related sites by $\approx$2,966%). When looking at singular pages instead, Porn pages are a more prevalent category followed by Cryptocurrencies.

Thus, confirming that factoring in mirrors is essential, which was a common gap in the existing literature. Accounting for mirrors can also help make informed decisions when prioritizing actions designed to thwart these activities, making prosecution more effective.

Our experimental work lets Mimir focus the search on cybercrime. Our analysis reveals that more than 74% of the sites in our dataset relate to cybercrime or deviant activities. The remaining sites may or may not contain such activities, but by itself, it demonstrates that Mimir focuses on cybercrime. There is a significant presence of "Counterfeit" and "Porn" sites. The former provides trading material for various illegal activities such as carding, fake IDs, and money counterfeiting, while the latter offers both licit and illicit pornographic content. Together with counterfeit, hacking services stand out to be the most replicated HS for victimizing individuals through means such as credential theft or Denial of Service (DoS) attacks. It is worth noting that if we consider only unique sites, the percentage of cybercrime-related sites decreases to approximately 38%. However, the fact that 81.32% of the mirrors are related to cybercrime highlights the substantial effort that cybercriminals put into expanding their reach and attracting more clients. This suggests that the primary purpose of mirrors is to increase their visibility in the network, and hence, gain more market share.

Moreover, we conducted a use case focused on detecting child abuse material, achieving an accuracy of 88.33% in their detection. Specifically, we departed from an especially scarce dataset of just 20 manually detected child abuse sites and we devised a method to reliably identify child abuse material. We were able to detect 159 unique sites (and 505 mirrors) pages containing such material with an accuracy of 91.66%, out of which 45 (28.3%) were still operative from the same `.onion` address at the time of this writing. These sites have been reported to Law Enforcement and are currently being investigated.

Applications: A key contribution of our study is that we show that guiding a Tor crawler from the beginning, using a set of keywords, amplifies its efficiency to reach sites from areas of interest. Mimir enhances the search on specific topics or threats and enables research in the desired specific domain. This approach helps to prioritize the discovery of relevant sites, ensuring that critical targets can be identified quickly and comprehensively. Besides the mentioned benefits for Law Enforcement when prosecuting illegal activities, Mimir can assist in informing appropriate cybersecurity policies, e.g., facilitating studies on current cybersecurity threats, vulnerabilities, and attack trends. In this regard, mirror detection plays a crucial role in enhancing threat intelligence and stakeholder analysis. By identifying multiple mirrors of a site, investigators can ensure continuous monitoring of deviant activities, even if some of the domains are taken down or moved.

While mirror sites amplify the detection coverage, they can create challenges when interpreting data. Indeed, we believe that our study should drive researchers to get more accurate insights by considering the noise and nuances of mirrors in Tor measurements. Without considering mirror effects, analysis may result in a distorted view of the dark web ecosystem. This, in turn, may lead to incorrect conclusions (e.g., an increase in a specific type of malware or a surge in demand for a particular illegal item), causing necessary actions to be overlooked due to decision fatigue or cognitive overload.

VI. RELATED WORK

Previous work on Tor has focused on understanding criminal activities, though often looking at particular sites, e.g., forums or markets [36], [26], [39], [24]. Other works have also studied the entire network ecosystem, focusing on particular activities and looking at the topology of Tor and its content. We next provide an overview of the most relevant ones, which we summarize in Table XI.

Table XI: Related work compared for crawling period, #seeds, crawl depth, URLs seen, Hidden URLs (# HS), Hidden Second-Level Domains (# HSLD) analyzed, reachable (200), and categories analyzed (# Cat). Symbol – indicates unavailable or unclear information.

	Period (until)	#Seeds	Depth	#URL	#HS	#HSLD	200	#Cat
[14]	1 month (m)	–	1	–	–	26K	2,125	31
[4]	1 week	20K	max	7-67M	34K		7,566	30
[1]	2m (07/'16)	–	2	–	250K		7,931	26
[2]	2m (07/'17)	–	2	–	[1] + 125K		10,367	25
[7] [6]	4m (05/'17)	–	–	–	3M	30K	10,685	– [1]
[42]	1m (07/'18)	20K	4	1.2M	40K	1,766	7,782	9
[10]	4m (01/'19)	-	-	144.5k	20.4k	-	7,831	6
Us	8m (05/'22)	6,816	1	n/a	n/a	50,294	24,911	12

Crawling. Al-Nabki et al. crawled in 2017 over 250k sites [1]. Then, they manually labeled sites into 24 classes, leading to the DUTA dataset with 7k sites available, which we use in our study. DUTA was later updated in 2019 [2] — leading to 3k new addresses — and used recently in 2022 [7], [6]. However, these works do not account for mirrors, as evidenced in our paper, which are a prevalent phenomenon in Tor.

Mirrors. Our work is motivated by prior work observing mirrors in Tor [10], [41] in just 7.8k `.onion` services collected in 105 days [10]. Authors in [41] use edit distances

to detect phishing and mirror websites, finding the latter in the links of the original HS or in those so-called "yellow pages" services.

However, their contribution differs from our research. First, the way they account for mirrors (cryptographic hashes, screenshots) can not effectively capture mirror modifications as we do (cf. Figure 5). In particular, they rely on hashes of the HTML content and hashes of the screenshots of the pages for their detection using Jaccard Similarity or Hamming distance, which have been proven to not be as effective as our approach in our benchmark on §III-D. For instance, the latest work is DarkBERT [20] . Authors conducted mirror detection using MinHash, achieving a duplication rate of 18.69% [9]. However, they do not specify the similarity threshold or how they fine-tuned the algorithm. Furthermore, since it relies on text tokenization into sets, redundancy in HTML might introduce noise, reducing its effectiveness when using Jaccard similarity. Second, **they do not *measure* the impact mirrors have on the network topology**.

Our work. Overall, our work differs from related work on three main axes. First, we offer a novel and systematic mechanism to discover seeds that prove to be more effective in discovering hidden services. Second, we consider for the first time the role of mirrors in the measurement of the Dark Web. Finally, our system provides a more persistent method to crawl Tor services that is more suitable when dealing with volatile content. Table XI shows a comparison in terms of coverage of our work with existing works in the literature. Since our work does not consider URLs, we do not report these figures (both surface and Hidden Services) seen. Our seeding, however, led to nearly 50k Hidden Second-Level Domains (25k available). This shows that our crawler sees more sites than previous works, even with a shallow-crawling strategy.

VII. Conclusions

In this paper, we presented the first mirrorless and the largest measurement study of the Tor Network to date. We developed a crawler that, together with an innovative seeding method, allows us to study nearly 25k hidden services visited over a period of eight months. While only visiting landing pages, we were able to cover a similar or higher number of sites as in previous works digging further. We showed that the topology of the Dark Web is highly fragmented, with thousands of interconnected independent networks (namely, "bubbles") of varying sizes, with a small number of networks having a large number of nodes all highly connected.

We designed a custom algorithm to detect mirrors and showed that a large proportion of our dataset (83% according to our estimates) consists of replicated content. Together with an analysis categorizing the content of the different hidden services, we showed the importance of taking into account mirrors when analyzing the Dark Web. We found that a large proportion of the network (67.33% unique sites and 74.4% counting also mirrors) was related to cybercrime activities. Also, we showed the benefits of our methodology to law enforcement and cybercrime researchers through a case study analyzing sites that potentially distribute child abuse material.

Overall, our study offers new insights into the content and structure of Tor, highlighting the importance of using tools that account for duplicate content and paving the way for future research in this area.

References

[1] M. W. Al Nabki, E. Fidalgo, E. Alegre, and I. de Paz, "Classifying illegal activities on tor network based on web textual contents," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 35–43. [Online]. Available: https://aclanthology.org/E17-1004

[2] M. W. Al-Nabki, E. Fidalgo, E. Alegre, and L. Fernández-Robles, "Torank: Identifying the most influential suspicious domains in the tor network," *Expert Systems with Applications*, vol. 123, pp. 212–226, 2019.

[3] M. A. Arshad, S. Shahriar, and K. Anjum, "The power of simplicity: Why simple linear models outperform complex machine learning techniques–case of breast cancer diagnosis," *arXiv preprint arXiv:2306.02449*, 2023.

[4] G. Avarikioti, R. Brunner, A. Kiayias, R. Wattenhofer, and D. Zindros, "Structure and content of the visible darknet," *arXiv preprint arXiv:1811.01348*, 2018.

[5] F. Barr-Smith and J. Wright, "Phishing with a darknet: Imitation of onion services," in *2020 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2020, pp. 1–13.

[6] M. Bernaschi, A. Celestini, M. Cianfriglia, S. Guarino, F. Lombardi, and E. Mastrostefano, "Onion under microscope: An in-depth analysis of the tor web," *World Wide Web*, pp. 1–27, 2022.

[7] M. Bernaschi, A. Celestini, S. Guarino, F. Lombardi, and E. Mastrostefano, "Spiders like onions: On the network of tor hidden services," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 105–115. [Online]. Available: https://doi.org/10.1145/3308558.3313687

[8] Breachsense, "Ransomware threat actors," https://www.breachsense.io/ransomware-gangs/, last accessed: 22-03-2023.

[9] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, "Minwise independent permutations," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998, pp. 327–336.

[10] P. Burda, C. Boot, and L. Allodi, "Characterizing the redundancy of darkweb .onion services," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ser. ARES '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3339252.3339273

[11] L. Choshen, D. Eldad, D. Hershcovich, E. Sulem, and O. Abend, "The language of legal and illegal activity on the darknet," *arXiv preprint arXiv:1905.05543*, 2019.

[12] M. Conti, S. Crane, T. Frassetto, A. Homescu, G. Koppen, P. Larsen, C. Liebchen, M. Perry, and A.-R. Sadeghi, "Selfrando: Securing the tor browser against de-anonymization exploits." *Proc. Priv. Enhancing Technol.*, vol. 2016, no. 4, pp. 454–469, 2016.

[13] M. DeVito, A. Walker, J. Birnholtz, K. Ringland, K. Macapagal, A. Kraus, S. Munson, C. Liang, and H. Saksono, "Social technologies for digital wellbeing among marginalized communities," in *CSCW 2019 Companion - Conference Companion Publication of the 2019 Computer Supported Cooperative Work and Social Computing*, ser. Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW. Association for Computing Machinery, Nov. 2019, pp. 449–454, publisher Copyright: © 2019 Copyright is held by the owner/author(s).; 22nd ACM Conference on Computer-Supported Cooperative Work and Social Computing, CSCW 2019 ; Conference date: 09-11-2019 Through 13-11-2019.

[14] M. Faizan and R. A. Khan, "Exploring and analyzing the dark web: A new alchemy," *First Monday*, 2019.

[15] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," *ACM SIGCOMM computer communication review*, vol. 29, no. 4, pp. 251–262, 1999.

[16] K. M. Finklea, "Dark web," Congressional Research Service, Tech. Rep., 2017.

[17] A. Forte, N. Andalibi, and R. Greenstadt, "Privacy, anonymity, and perceived risk in open collaboration: A study of tor users and wikipedians," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2017, pp. 1800–1811.

[18] E. Hartney. (2023, May 01) Deviant behavior: Definition, causes, and types. [Online]. Available: https://www.verywellmind.com/socially-acceptable-to-socially-deviant-addictions-22243

[19] E. Jardine, "The dark web dilemma: Tor, anonymity and online policing," *Global Commission on Internet Governance Paper Series*, no. 21, 2015.

[20] Y. Jin, E. Jang, J. Cui, J.-W. Chung, Y. Lee, and S. Shin, "Darkbert: A language model for the dark side of the internet," *arXiv preprint arXiv:2305.08596*, 2023.

[21] G. Kadianakis and K. Loesing, "Extrapolating network totals from hidden-service statistics," *Tor Tech Report*, no. 01 (001, 2015.

[22] J. A. Kloess and M. van der Bruggen, "Trust and relationship development among users in dark web child sexual exploitation and abuse networks: a literature review from a psychological and criminological perspective," *Trauma, Violence, & Abuse*, p. 15248380211057274, 2021.

[23] J. Kornblum, "Identifying almost identical files using context triggered piecewise hashing," *Digital investigation*, vol. 3, pp. 91–97, 2006.

[24] V. Labrador-Ortiga and S. Pastrana, "Examining the trends and operations of modern dark-web marketplaces," in *Proceedings of the 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2022.

[25] C. G. Mena, A. De Caigny, K. Coussement, K. W. De Bock, and S. Lessmann, "Churn prediction with sequential data and deep neural networks. a comparative analysis," *arXiv preprint arXiv:1909.11114*, 2019.

[26] S. Nazah, S. Huda, J. Abawajy, and M. M. Hassan, "Evolution of dark web threat analysis and detection: A systematic approach," *IEEE Access*, vol. 8, pp. 171 796–171 819, 2020.

[27] J. Nurmi, "Ahmia — search tor hidden services." [Online]. Available: https://ahmia.fi/

[28] G. H. Owenson and N. J. Savage, "The tor dark net," 2015.

[29] S. Pastrana, A. Hutchings, D. Thomas, and J. Tapiador, "Measuring ewhoring," in *Proceedings of the Internet Measurement Conference*, 2019, pp. 463–477.

[30] S. Pastrana, D. R. Thomas, A. Hutchings, and R. Clayton, "Crimebb: Enabling cybercrime research on underground forums at scale," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1845–1854.

[31] C. Peersman, C. Schulze, A. Rashid, M. Brennan, and C. Fischer, "icop: Automatically identifying new child abuse media in p2p networks," in *2014 IEEE Security and Privacy Workshops*, 2014, pp. 124–131.

[32] T. Pranckevičius and V. Marcinkevičius, "Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification," *Baltic Journal of Modern Computing*, vol. 5, no. 2, p. 221, 2017.

[33] T. Project. (2023, Feb,27) Tor research safety board. [Online]. Available: https://research.torproject.org/safetyboard/

[34] R. Rivest, "The md5 message-digest algorithm," Tech. Rep., 1992.

[35] N. Shuyo, "Language detection library for java," 2010. [Online]. Available: http://code.google.com/p/language-detection/

[36] K. Soska and N. Christin, "Measuring the longitudinal evolution of the online anonymous marketplace ecosystem," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 33–48.

[37] G. Suarez-Tangil, J. E. Tapiador, F. Lombardi, and R. Di Pietro, "Thwarting obfuscated malware via differential fault analysis," *Computer*, vol. 47, no. 6, pp. 24–31, 2014.

[38] TorSearch, "Torch search – search the dark net," https://torchsearch.wordpress.com/, 2019, last accessed: 19-05-2022.

[39] R. Van Wegberg, S. Tajalizadehkhoob, K. Soska, U. Akyazi, C. H. Ganan, B. Klievink, N. Christin, and M. Van Eeten, "Plug and prey? measuring the commoditization of cybercrime via online anonymous markets," in *27th USENIX security symposium (USENIX security 18)*, 2018, pp. 1009–1026.

[40] VisiTor, "Visitor — search engine for the dark-web," http://uzowkytjk4da724giztttfly4rugfnbqkexecotfp5wjc2uhpykrpryd.onion/search/?q=, 2022, last accessed: 19-05-2022.

[41] C. Wang, J. Luo, Z. Ling, L. Luo, and X. Fu, "A comprehensive and long-term evaluation of tor v3 onion services," in *Proceedings of the 42nd IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2023.

[42] M. Zabihimayvan, R. Sadeghi, D. Doran, and M. Allahyari, "A broad evaluation of the tor english content ecosystem," in *Proceedings of the 10th ACM Conference on Web Science*, 2019, pp. 333–342.