Future Generation Computer Systems 145 (2023) 12–26

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

On the gathering of Tor onion addresses

Javier Pastor-Galindo *, Félix Gómez Mármol, Gregorio Martínez Pérez

Department of Information and Communications Engineering, University of Murcia, 30100 Murcia, Spain

ARTICLE INFO

Article history:
Received 9 July 2022
Received in revised form 13 February 2023
Accepted 26 February 2023
Available online 15 March 2023

Keywords:
Onion services
Tor network
Dark web
Cybersecurity
Cybercrime
Anonymity

ABSTRACT

Exploring the Tor network requires acquiring onion addresses, which are crucial for accessing anonymous websites. However, the Tor protocol presents a challenge, as it lacks a standard method for finding these complex links composed of either 16 or 56 base32-coded characters and featuring the unique ".onion" top-level domain. This study delves into the existing literature analyzing onion services and categorizes the various strategies employed to gather their addresses. The success of each approach is measured by the number of addresses obtained, while the relevancy of the work is evaluated by comparing the number of services uncovered to Tor's official count. The results indicate that the most used techniques are Tor crawling and repositories, whereas the most effective methods are relay injection, repositories, and Tor crawling. This paper also estimates the representativeness of literature collections, revealing that most past works explored a small portion of the Tor network. The study also uncovers the limitations of onion gathering and sheds light on the challenges for future research to provide more representative datasets for dark web exploration.

1. Introduction

The Onion Router (Tor) is a network that overlays the traditional network by implementing a privacy-preserving protocol. Users connect via client applications to the services offered, currently called onion services and formerly known as hidden services, with anonymity for both sides [1]. Tor is not the only network with peculiar privacy, accessibility, and operations. There are others of a similar nature, such as Freenet, I2P, or ZeroNet. While the content accessed with traditional browsers is known as the surface web, these anonymous networks constitute the dark web [2].

Criminal actors or organized groups exploit the identity protection raised from Tor to host illicit activities in onion services (e.g., a black market of weapons, drugs or stolen credentials, sale of child sexual abuse content, or cybercrime-as-a-service) [3]. For this reason, Tor attracts the attention of researchers seeking to characterize its performance, security, or structure [4]. Also, Law Enforcement Agencies (LEAs) are especially interested in monitoring onion services for situational awareness and forecasting purposes [5].

In such a scenario, obtaining the onion addresses that give access to onion services is essential. Tor v2 employed 16 random base32-encoded characters,[1] but this version was deprecated in October 2021. Instead, v3 onion addresses consist of 56 base32-encoded characters,[2] allowing for the potential existence of $32^{56} \cdot 10^{84}$ onion services.[3] The standard ".onion" special-use domain name [6] is employed to locate and reach a given service in the Tor network, which is not resolvable by the standard domain name system (DNS) infrastructure.

The task of discovering, identifying, and managing this type of link is challenging, particularly due to the anonymous nature of Tor, the random and non-mnemonic onion addresses, the absence of a DNS within the Tor network, and the low longevity of onion services, which sometimes change their addresses dynamically [7]. The problem is compounded by the need to analyze extensive collections of onion services to develop robust research and intelligence [8]. Several resolution systems have been proposed to support meaningful domains instead of random-looking onion addresses, but they end up compromising the privacy and security of the Tor protocol [9].

Analysts employ gathering techniques to obtain unknown and never-seen onion addresses in the wild. This collection is a critical aspect of dark web research, where the methodology employed, the number of onion addresses found, and the representativeness of the outcomes are essential [10]. Researchers from Princeton University studied how people interacted with onion services through 17 interviews and an online survey of 517 users [11].

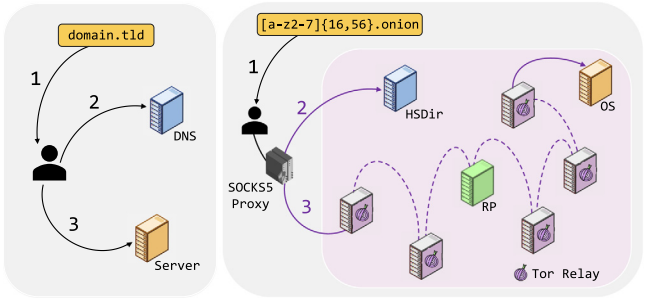
* Corresponding author.
E-mail addresses: javierpg@um.es (J. Pastor-Galindo), felixgm@um.es (F. Gómez Mármol), gregorio@um.es (G. Martínez Pérez).

[1] E.g., facebookcorewwwi.onion.

[2] E.g., facebookwkhpilnemxj7asaniu7vnjjbiltxjqhye3mhbshg7kx5tfyd.onion.

[3] Assuming that the number of atoms in the entire visible universe is estimated to be 10^{80}, the search space is huge.

https://doi.org/10.1016/j.future.2023.02.024



(a) Non-onion domains (b) Onion domains

Fig. 1. Simplified access to online servers.

They exposed the problem of obtaining onion services and authenticating them as trusted sites, suggesting the need for effective mechanisms for discovery. Results revealed that 48% of respondents identified them on social networks, 46% on onion search engines, 46% browsing the web, 19% by word of mouth, and 16% by onion lists. Florian and Lux [10] identified two techniques for gathering onion addresses: web crawling, where a program navigates the web, following links to find and retrieve addresses; and setting up Hidden Service Directories (HSDirs) to gather the onion addresses of services that publish their accessibility as descriptors. However, they caution against relying solely on web crawlers to collect these addresses as they cannot uncover services such as SSH, FTP, email, chat, or botnets [12,13]. The authors emphasize the importance of selecting appropriate starting points, or seed pages, for a web crawler, pointing out that lists compiled by third parties, existing Tor-based search engines, or previous works often result in a significant bias due to the preselection made by the compiling party. Trujillo and Ruíz-Martínez [4] determined that the discovery and measurement of onion services was a significant area of research. The reviewed articles employed several methods to obtain onion addresses, including manually locating services, operating HSDirs, or using search engines with specific keywords. However, they conclude that no studies validated and compared these discovery methods.

There are surveys related to anonymous communication (such as Tor, Freenet or I2P) studying the use of anonymous network technologies for illegal activities [14], the potential attacks and protection mechanisms [15], or the state of research in the field and its challenges [16]. Another study compares the Tor and I2P networks through strengths and weaknesses in specific applications [17]. Other reviews focus explicitly on the Tor network, delving into its benefits and drawbacks [18], providing an in-depth look into the unresolved issues and challenges [19], summarizing recent advancements in discovering Tor network bridge nodes [20], or focusing on the research being done [21]. Other works review the numerous attacks applied in Tor and their limitations [22], inspect de-anonymization attacks [23,24], or identify traffic analysis attacks [25]. The only survey related to onion services provides a comprehensive overview of research on these services and highlights its findings, limitations, and future issues [4]. However, no surveys inspect the critical aspect of gathering onion addresses and collecting methods. As a result, the current state of Tor-based research may be limited by the number of addresses gathered, which directly impact the size and representativeness of the datasets generated and, in turn, the generalizability of research results.

To fill the gap and elaborate the first review of onion address gathering, academic works exploring onion services are inspected in this paper to:

1. Identify, describe and compare the techniques used to obtain onion addresses in the literature.
2. Calculate the usage of the techniques through the number of times they have been adopted.
3. Measure the effectiveness of the techniques in the literature based on the number of detected addresses.
4. Estimate the representativeness of gathered onion samples with respect to the total existing onions services.

Accordingly, this paper presents the current limitations for collecting significant samples of onion domains and unveils some challenges to improve the performance of current methods and expand the potential reachable part of the dark web.

2. Background on Tor onion services

2.1. Discovery of non-onion and onion domains

Accessing online services has become a part of daily life, with conventional servers accessible through a simple process. As presented in Fig. 1(a), the first step is obtaining the domain name and top-level domain (TLD) of the server. The request then performs Domain Name Resolution (DNS), translating the domain name into an IP address. After obtaining the IP address, the client initiates the final connection to the server using the IP address and the port number of the desired service. The data exchange is performed through this socket and under a particular protocol.

For added privacy and security, the Tor network offers alternative access through the decentralized system to any online service, providing client-side anonymity. Furthermore, any administrator can deploy an onion service on the Tor network to gain server-side anonymity, which can only be reached through the Tor network. In this sense, accessing an onion service requires obtaining its unique onion address, which is a random string of 16 or 56 characters ending in ".onion", and a SOCK5S proxy, such as the one integrated into the Tor Browser, to establish the protocol for data to be transmitted over the Tor network. As shown in Fig. 1(b), reaching an onion service requires contacting a special node called Hidden Service Directory (HSDir) to get the contact information of the onion service (OS). Finally, the client creates a circuit through the network, routing the request through several randomly selected Tor relays and agreed Rendezvous Point (RP), which keeps the user's IP address anonymous and the connection encrypted.

2.2. Interaction with onion services

As introduced above, accessing an onion service is not directly entering a URL known by heart in the browser search bar and quickly interacting with the server hosting the site. In Fig. 2, the interaction with Tor onion services is divided into the phases of gathering onion addresses, access to onion services and analysis of them are modeled in depth.

2.2.1. Gathering of onion addresses

The process of collecting Tor onion addresses entails several phases that are illustrated in the flowchart. The initial step involves locating an onion advertiser, which can be done using various methods that will be further discussed in the paper. Once an advertiser is found, the next step is to search for a new onion domain within the advertiser. If the domain is not found, another advertiser must be located and the search process must be repeated. If a new onion domain is discovered, the next step is to validate it to ensure it is a valid onion address. This can be done by checking if the address is a 16-character base32-encoded string for version 2 addresses or a 56-character string

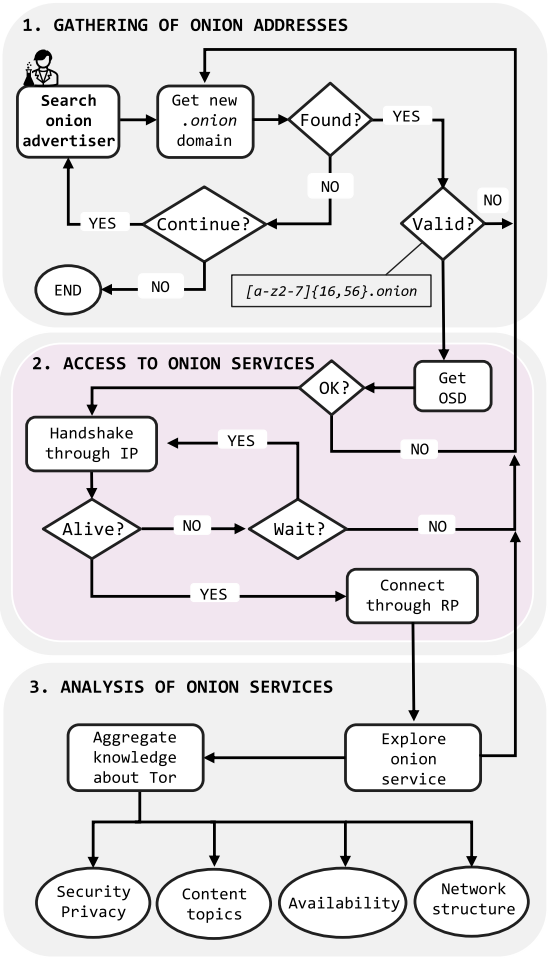


Fig. 2. Flowchart of interaction with Tor onion services.

for version 3 addresses. If the address is invalid, the process continues by searching for a different onion domain within the same advertiser.

However, if the obtained domain is a valid onion address, it can be accessed through the Tor network to reach the onion services it provides. The details of accessing onion services through the Tor network are discussed in Section 2. The process of collecting onion addresses ends when there is a decision to stop searching for more advertisers.

2.2.2. Access to onion services

Accessing a Tor onion service is a complex process comprised of several steps, which can be grouped into three main stages: the discovery of the service, the establishment of a connection, and the final connection to the service.

In the discovery stage, the user must obtain the Onion Service Descriptor (OSD) associated with the valid onion address previously gathered. The OSD is stored by one or more HSDir nodes, which act as a kind of index for onion services. The user sends a request for the OSD to an HSDir node, which returns the necessary information for establishing a connection, including a set of Introduction Points (IPs) and a Rendezvous Point (RP). If the descriptor is not retrieved, either because it has yet to be published or the onion service does not exist, then the connection cannot be established. The workflow continues with identifying a new onion domain in the last advertiser.

In the establishment of a connection stage, the client performs a handshake with the onion service through one of the (IPs). During the handshake, the client and the onion service negotiate encryption keys and verify the authenticity of each other, ensuring the security of the connection. Once the handshake is complete, the client and the onion service use the agreed keys to encrypt all further communication. The onion service may not be alive, so the client may wait to retry again or discard the onion service and continue the investigation with the next onion domain.

Finally, in the final connection stage, the client establishes a connection to the onion service through the Rendezvous Point (RP). The RP acts as a meeting point between the client and the onion service, allowing them to exchange data securely. The RP does not know the content of the data exchanged between the client and the onion service, preserving the privacy and security of the connection.

2.2.3. Analysis of onion services

From a research perspective, gathering onion addresses and accessing to onion services are necessary to explore the Tor network. In this sense, the main research areas are focused on evaluating security and demonstrating potential cyberattacks, automated content classification, or the challenge of discovering and categorizing these resources [4]. For example, deanonymization techniques, such as network traffic analysis, exploiting vulnerabilities and poor security practices, can expose users within the Tor network [1]. In this sense, onion services are crucial for researching deanonymization attacks on the Tor network and protecting user identities [26].

Tor sites are explored to identify illegal activities. The automated characterization of content is used to spot child sexual abuse, hacking services, or black market, which have been found on 45% of onion services [3]. Nevertheless, other work revealed that 47% were ethical sites offering content such as file sharing and discussion forums [27].

The study of the lifetime of the onion services contributes to operational tasks of pursuit. An experiment showed that 50% of observed onion services were deactivated after 24 h and reduced to 40% after 300 h, revealing that long-lived onion services were inconsistent, constantly going up and down [28]. In other experiments, only 36% of identified onion services were alive during the 18-week time window [8], and onion services switched addresses with 20% having multiple URLs [7].

Onion services are also investigated to understand the internal structure of the Tor network because it is different from the surface web. One study demonstrates that it forms a hub-and-spoke network with 30% of onion sites pointing to one node and 87% not referencing other services [29]. Some highly connected nodes act as authorities in the network, such as a hosting service and a bitcoin explorer, referenced by 14% and 5% of onion services, respectively [8]. Link directories also play a connecting role, referencing 71% of onion services.

3. Gathering onion addresses in the literature

Due to the lack of a standard manner to collect onion addresses, this section identifies and characterizes techniques employed by researchers. Complementary information and the analysis carried out are also publicly available.[4] The analysis of the results is presented next.

3.1. Review methodology

First, to locate works directly related to Tor services, we searched for the pattern "*dark web sites*" OR "*dark web services*"

[4] https://github.com/javier-pg/onions-discovery

OR "*dark web hidden services*" OR "*dark web pages*" OR "*dark web domains*" OR "*tor sites*" OR "*tor services*" OR "*tor pages*" OR "*tor domains*" OR "*tor hidden addresses*" OR "*onion addresses*" OR "*onion services*" OR "*onion domains*" OR "*hidden services*" in two of the most popular academic search engines, Google Scholar and Scopus. The former returned the maximum number of 10,000 entries[5] ordered by relevance, while the latter identified 282 results.[6] Considering peer-reviewed works published at journals and conferences, we found 54 works harvesting onion services for different purposes. These resulting papers were published from 2014 to 2022 and are limited to Tor v2. This is a limitation of this study and reveals the lack of research on the recently released version 3 [30].

3.2. Works collecting onion addresses

In studies related to the dark web and Tor network, it is indispensable to collect onion addresses as a preliminary step, as mentioned in Section 2.2.1. The following articles are grouped according to their purpose, although this analysis focuses on their Tor link gathering process.

First, different solutions directly face the challenge of the automatic identification and analysis of onion services. The out-of-band discovery concept [39] is introduced to use surface search engines, such as Google or Bing, for feeding crawling within the Tor network. The idea is tested with Google and Bing, collecting 4857 live sites from a total of 170,581 collected addresses. CARONTE [38] and POSTER [54] are tools adopting the same concept. The former crawls onion domains from Deep Web Links, The Hidden Wiki, Ahmia, DuckDuckGo, Google, or Bing to discover location leaks in them, reaching 48,418 onion services in which 1964 are active at some point. The latter is a containerized crawler in which seeds are taken from Google and Bing, analyzing 8157 active services from 16,683 onion addresses identified. The solution employs docker to achieve scalability, a virtualization technology also present in another crawler [63] that explores onion links obtained in Google, Yahoo, Bing, Baidu, and DuckDuckGo, obtaining 2527 active sites from 25,361 addresses identified in the experiments.

Following similar alternatives, Dark Crawler [47] is fueled with user-specified sites and can reach 10,163 onion services. ATOL [57] permits the automated analysis and categorization of Tor services, evaluated with 28,928 live onions from 100,000 addresses identified. Another crawler is developed to reach 12,396 onion services [75]. There are also more ambitious designs, such as the Dark Web Threat Intelligence Analysis (DWTIA) Platform [56], a framework to analyze dark web trends in correlation with other intelligence inputs launched with 8000 onion addresses extracted from Tor2Web Open Data streams.

Although most proposals are based on crawling, other alternatives exploit a vulnerability that allowed volunteer relays acting as HSDir to compile descriptors and associated onion addresses. This Tor v2 flaw was published in 2012 [31], demonstrating the collection of 6579 active onions. From that milestone, several works injected their relays into the Tor networks to obtain and analyze 12,000 [34], 13,337 [35], 4000 [80], 82,145 [69] or 3900 [64] onion services. Other works employ the aforementioned method and other complementary sources, such as crawlers seeded with tor-specific search engines to obtain 13,145 [37], and repositories (e.g., Pastebin, Reddit or The Hidden Wiki) to collect 46,562 Tor links [2]. Even onions obtained through the vulnerability are used as seeds for a crawler to finally analyze 53,466 persistent sites [61].

There is a strong trend in analyzing the content of sites to categorize the topics, themes, or legality of the dark web. Some reviewed studies with that objective query ready-to-use compilations (e.g., The Hidden Wiki, Snap BBS, Ahmia, VisiTOR, or Harry71) to obtain 1171 [32], 12542 [66], 3000 [73] or 5625 onion addresses [45]. Other ones implement spiders to reach 1450 [33], 4000 [48], 6227 [62], 3000 [65], 5101 [67], 1766 [68], 157 [77], 3595 [79] and 7954 [51] active onions, which are frequently seeded with predefined listings of links. Some researchers are particularly focused on spotting phishing sites, collecting 23,585 [50], 9084 [70], and 4000 [76] live onions through crawling for their experiments.

The research on the graph formed by onion services and their interconnections is notable. Studies infer the topological structure of the Tor network, exploring the hyperlinks between 5144 [52], 29,473 [8], 7178 [58], 1220 [59], 15,000 [60], 4041 [78] or 48,174 [74] active onion services. Two approaches even explore the links to the surface web, considering 7257 [43] and 3288 [72] Tor nodes in respective studies. A particular work proposes a ranking algorithm to identify influential nodes, proven in a graph of 3536 onion services [55]. These graph-based studies use predefined compilations (e.g., The Hidden Wiki, Ahmia, Onion Links, Pastebin, or Reddit) or Google to feed custom spider processes.

A considerable block of papers proposes algorithms and models to compromise onion services. Different fingerprint and deanonymization attacks are evidenced against 1000 [36], 1500 [41], 790 [42], 1714 [44], and 5144 [49] onion services, which are collected from existing listings (e.g., Ahmia) and onions search engines. The so-called eclipse attack is demonstrated to block 3399 onion services [40].

Other works collected onion addresses to analyze peculiarities of the associated services. For example, the existence of short-lived and long-lived onion services is evidenced by 3047 onion domains obtained from crawling Reddit and The Hidden Wiki [28]. A study [46] analyzes binding relations between sites and shows common or meaningful prefixes from a pool of 13,604 onion services obtained in The Hidden Wiki and Tor Links. MASDEAL [7] is a system to measure the redundancy of Tor sites and identify mirrors, tested with 7831 crawled and active onions.

Finally, the widely used Darknet Usage Text Addresses (DUTA) [46] was built by crawling 7931 sites extracted from Ahmia and Onion City. Another study analyzing open-source blocklisting [71] was able to detect 496 onion addresses.

Table 1 summarizes the main aspects of the works mentioned above. The gathering year represents the date on which the onions were collected (which mostly does not match with the article's publication), ranging from 2012 to 2021. The gathering duration specifies the number of days of the collection. The total onion addresses indicates the number of unique onion domains retrieved. As not all collected onion addresses may exist or be alive, researchers also document the proportion of them that are actually active and used for the experiments. Many papers simply present a number of onions without distinction, so we assume them as active. The gathering technique indicates the method employed to compile onion addresses. The tools used are differentiated in five different types, which are characterized in the next section.

4. Techniques for gathering onion addresses

We extract five main techniques of harvesting onion addresses. Three of them are out-of-band methods executed on the surface web, namely onion search engines, repositories, and generic search engines. The remaining two groups are in-band mechanisms performed within the anonymous network, namely Tor crawling and relay injection.

[5] Search executed on 4 Nov, 2022.

[6] Search executed on 10 Nov, 2022.

Table 1
Academic works collecting onion addresses.

Work	Gathering year	Gathering duration (days)	Total onion addresses	Active onion addresses	Gathering techniques				
					Relay injection	Onion Search Engines	Generic Search Engines	Repositories	Tor crawling
[31]	2012	2	39,824	6,579	✓				
[32]		32	1,171	1,171				✓	
[33]	2013	31	7,000	1,450					✓
[34]	2014	160	80,000	12,000	✓				
[35]		28	13,337	13,337	✓				
[36]	2015	90	2,000	1,000				✓	
[37]		120	48,418	13,145	✓	✓			✓
[38]		12	6,426	1,974			✓	✓	✓
[39]		35	170,581	4,857				✓	✓
[40]		30	3,399	3,399				✓	
[41]		1	1,500	1,500				✓	✓
[42]	2016	1	1,363	790				✓	
[43]		3	198,050	7,257				✓	✓
[44]		N.A.	1,714	1,714		✓			
[45]		2	12,882	5,625				✓	
[46]		60	250,000	7,931				✓	✓
[47]		N.A.	10,163	10,163					✓
[48]		60	4,000	4,000					✓
[49]		N.A.	5,295	5,295				✓	✓
[50]		N.A.	43,000	23,585				✓	✓
[51]		81	7,954	7,954				✓	✓
[52]	2017	42	5,144	5,144			✓	✓	✓
[28]		180	4,707	3,047				✓	✓
[8]		42	29,475	29,473			✓	✓	✓
[53]		210	13,604	13,604				✓	
[54]		9	16,683	8,157			✓		✓
[55]		90	124,589	3,536		✓		✓	✓
[56]		10	8,000	8,000				✓	✓
[57]		210	100,000	28,928				✓	✓
[58]		120	7,178	7,178				✓	✓
[59]		90	1,220	1,220				✓	✓
[60]		N.A.	15,000	15,000				✓	✓
[61]	2018	150	176,276	53,466	✓				✓
[7]		105	7,831	7,831					✓
[62]		30	25,742	6,227				✓	✓
[63]		60	25,361	2,527			✓		✓
[64]		1	3,900	3,900	✓				
[65]		413	3,000	3,000				✓	✓
[66]		34	27,248	12,542		✓		✓	
[67]		298	8,910	5,101				✓	✓
[68]		60	1,766	1,766				✓	✓
[69]		90	173,190	82,145	✓				
[70]	2019	N.A.	11,544	9,084				✓	✓
[2]		730	46,562	46,563	✓			✓	✓
[71]		188	496	496				✓	
[72]		27	3,288	3,288				✓	✓
[73]		1	3,000	3,000				✓	
[74]	2020	N.A.	48,174	48,174				✓	✓
[75]		73	12,396	12,396					✓
[76]		N.A.	4,000	4,000				✓	✓
[77]		N.A.	184	157					✓
[78]		N.A.	4,041	4,041				✓	✓
[79]	2021	N.A.	3,595	3,595					✓
[80]		14	4,000	4,000	✓				

4.1. Description of gathering techniques

4.1.1. Onion search engines

Given the difficulty of finding onion addresses on the web, tor-specific search engines have been developed to facilitate the discovery from the surface web. Fig. 3 presents the usage of this technique:

1. The researcher introduces a list of keywords of interest through a visual or programming interface.
2. The search engine transforms the input in a query to the internal database. The database is continuously populated and updated by automatic spiders deployed within the Tor network to index onion links.
3. The engine returns the results with the onion links that match the query according to particular search and ranking algorithms.
4. The researcher receives directly a list of onion links matching the initial keywords.

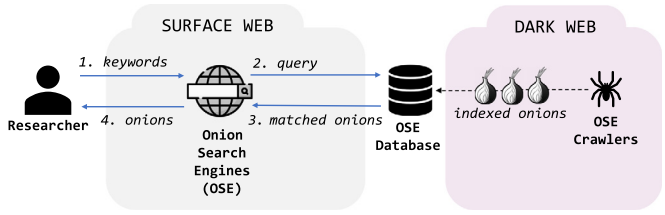


Fig. 3. Functioning of onion search engines.

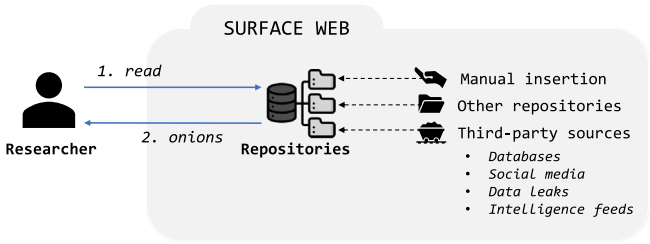


Fig. 4. Obtaining onion addresses in repositories.

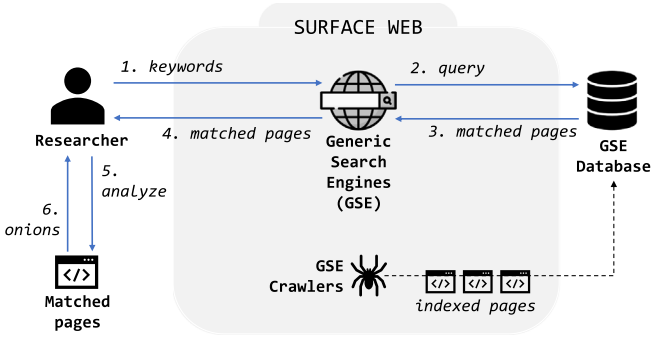


Fig. 5. Functioning of generic search engines.

This technique is accessible to the end-users and is often used for large-scale collections for its availability and ease of use. However, the result is a biased sample depending on the terms entered, the ranking and search algorithm of the tool, and the effectiveness of background spiders only reaching Tor links via crawling.

Reviewed works rely on initiatives such as the no longer existing Onion City, Onion Live,[7] Tor Link,[8] or Ahmia.[9] Although Ahmia is used as an onion search engines by some works, it is mainly employed as a repository of onion links.

4.1.2. Repositories

Some resources have specialized in compiling listings of onion addresses in a centralized service. Fig. 4 draws this straightforward procedure:

1. The researcher visits and reads the compilation of onion domains contained in repositories. These resources are built by the community and independent users, linked with other repositories, and fed with databases, social media streams or intelligence sources.
2. The list of onion addresses is directly downloaded. This step is usually automated with scripts or scrappers to be integrated into a whole system.

Using pre-created compilations is the most direct way to get onion addresses. However, the researcher should know their existence and rely on the potentially biased contributions of unknown contributors. Moreover, it is common to find outdated repositories or listings with inaccessible sites. Therefore, this technique may yield a considerable proportion of inactive onion services as repositories tend to have downed links.

According to our review, the most prominent and used tools are The Hidden Wiki,[10] and Ahmia.[11] Other works discover onion addresses in compilations such as Deep Web Links[12] Dark Web Links,[13] Reddit, Tor2Web Open Data project,[14] paste sites, or published datasets.

4.1.3. Generic search engines

The traditional search engines [5] are also valuable for identifying onion addresses through the surface web. Fig. 5 illustrates the employment of this technique:

1. The researcher introduces a list of keywords of interest through a visual or programming interface.
2. The search engine transforms the input in a query to the internal database. The database is continuously populated and updated by automatic spiders deployed on the surface web to index websites.
3. The engine returns the results with web pages that match the query according to particular search and ranking algorithms.
4. The researcher receives a list of web pages that matches the initial keywords.
5. The received pages are individually inspected to identify onion addresses. Thus, the HTML corpus is processed with string comparisons, link discovery or regular expressions matching strings of 16/56 base32-coded characters ending with ".onion".
6. The Tor links identified in the free text build the final list of onion services.

This technique stands intuitive for any user or analyst nowadays. Due to the generic nature of these tools, it is challenging to select proper keywords to avoid meaningless noisy websites without onion addresses. Additionally, it requires extracting and parsing components to obtain a clean listing of Tor links from returned content. Finally, it suffers from the bias of keyword selection, internal ranking and search algorithms, and the functioning of background crawlers. The result is a subset of onion links extracted from only indexed pages.

According to our review, researchers tend to type keywords, address patterns, or specific onion services on search bars, inspect resulting sites and get indexed onion links. Therefore, it is important to filter searches and analyze the results to capture onion addresses. Apart from the widely adopted Google, the reviewed works also used Bing, DuckDuckGo, Yahoo, or Baidu.

4.1.4. Tor crawling

This technique is based on automatically exploring the Tor network, following the links recursively, and saving the identified onion addresses. Fig. 6 demonstrates the crawling process in the Tor network:

7 https://onion.live (last access: 9 Feb, 2023).
8 https://tor.link (last access: 9 Feb, 2023).
9 https://ahmia.fi (last access: 9 Feb, 2023).
10 https://thehiddenwiki.org (last access: 9 Feb 2023).
11 https://ahmia.fi/onions (last access: 9 Feb 2023).
12 https://deepweblinks.co (last access: 9 Feb 2023).
13 https://darkweb.link (last access: 9 Feb 2023).
14 https://www.tor2web.org (last access: 9 Feb 2023).

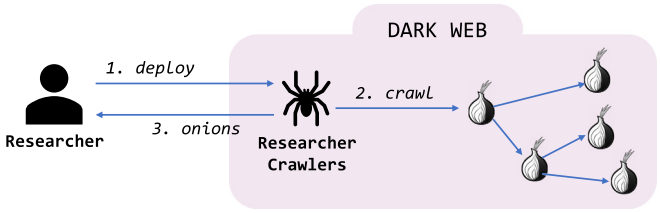


Fig. 6. Gathering onion addresses with Tor crawling.

1. The researcher designs, implements and deploys the crawlers on the dark web.
2. The spiders are configured with an individual onion, lists or repositories to start the crawling process from those seeds. This software automatically connects to the Tor network, recursively visits onion services through links found, and scrapes dark sites to extract onion addresses.
3. The list of onion services visited and scraped is returned to the researcher in streaming, batches, or at the end of the crawling. The crawler may be scheduled for a limited duration or configured to reach a certain number of onions. It may also end in the absence of new links to follow.

This technique has the complexity of designing and implementing the automatic spiders to navigate the Tor network. Some needs are necessary to bootstrap the solution, being common to use repositories and search engines to feed the crawling process. The process must be configured correctly to reach the Tor network, avoid memory leaks, and handle connectivity errors. In addition, spiders may bypass anti-crawling obstacles (access restrictions, logins, captchas, redirections, malicious links, IP banning, DDoS protection services, rate limits, obfuscation, etc [81].) placed by onion service administrators. Results will be biased by the initial seed list and only contain web-based onion services with incoming links. While Tor crawling is a more tedious discovery technique than the previous ones, the resulting onions are more likely to be active (at least, at the time of gathering). However, the performance of this technique is highly dependent on the configuration (number of threads, maximum depth, timeouts, etc.), the resources (number of machines, available memory, network bandwidth, etc.), and the duration of execution (time-limited or continuous).

Analysts and researchers in the literature implement crawlers via (i) connecting to the Tor network through a SOCKS proxy (such as Polipo or Privoxy), (ii) automating the interaction with the Tor Browser simulating a human user (with libraries such as Selenium); or (iii) using Tor gateways (such as Tor2Web) that act as proxies between the requests sent from the surface web (usual browsers) and the Tor network (consequently losing anonymity). It is worth noting that a group of revised solutions, such as CARONTE, POSTER, Dark Crawler or ATOL, directly designed and implemented this type of technique.

4.1.5. Relay injection

Any user can deploy a voluntary relay node on the Tor network to amplify the anonymous infrastructure and make the Tor network faster, more robust, more stable, and safer. As part of the protocol, directory authorities assign flags to relays under determined circumstances in compliance with the specification,[15] such as the *guard*, *exit*, *fast*, *stable*, *fast*, or *HSDir* flags. In this context, Fig. 7 describes the procedure employed by researchers to obtain onion addresses through relays in Tor v2:

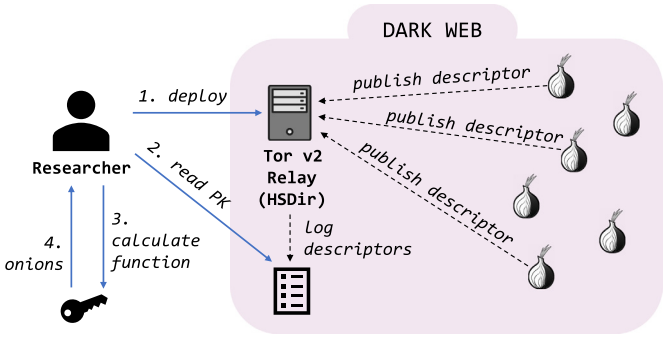


Fig. 7. Deployment of HSDir to gather onion addresses.

1. The analyst deploys one or various relays on the Tor network to eventually receive the HSDir flag. The latter can be assigned whether a node has been up for at least 96 h and has the *stable* and *fast* flags. From that point, it is ready to register the informative descriptors of other onion services needed by the Tor protocol for translating the onion address of an onion service into a list of introduction points.
2. Once the deployed relay becomes a HSDir, the researcher can access the descriptors being published by onion services. In particular, the associated public key (PK) is the essential part of this technique.
3. The analyst calculates a reversing function to derive the onion address from the public key of the onion descriptor. The latter is possible because the onion address is the base32 encoding of the first 80 bits of a SHA1 hashing of the public key (which is, in turn, derived from a private key).
4. A list of onion addresses can be directly obtained from the onion services that posted their descriptor to the deployed relay.

The inspection of descriptors returns a considerable variety of random onion domains. However, this technique is against the ethical guidelines for Tor research, being explicitly considered as an unacceptable activity.[16] The inference of the onion address from the public key of the descriptor is no longer possible in version 3 due to the employment of blind signatures so this technique can be viewed as deprecated.

4.2. Usage of gathering techniques

Considering the number of times the gathering techniques have been employed in the selected research works (see Table 1), their relevance is evaluated (Fig. 8).

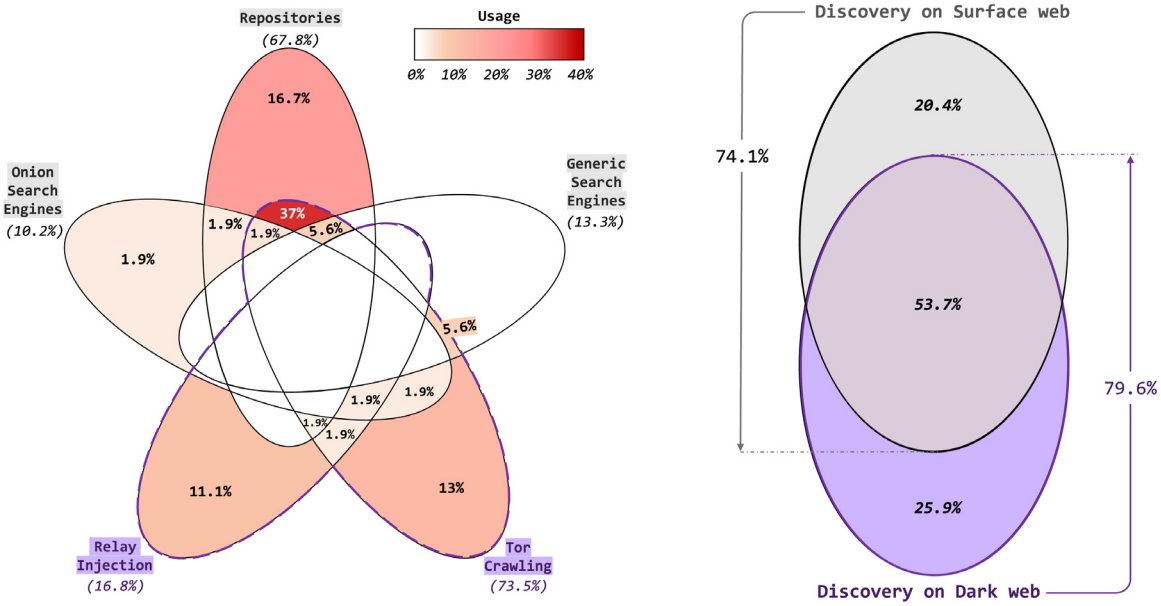
4.2.1. Methodologies of collection

Fig. 8(a) shows that techniques are used in isolation, in pairs, or even in groups of three, understanding how onion addresses have been uncovered in academic works up to the present day.

First, we count the works that somehow employ each technique, noting the associated proportion in brackets. In this sense, Tor crawling (73.5%) and repositories (67.8%) are the most commonly used collecting methods. Less widespread methods are relay injection (16.8%), generic search engines (13.8%), and onion search engines (10.2%). Therefore, analysts tend to use spiders seeded with repositories to collect onion domains.

15 https://github.com/torproject/torspec/blob/main/dir-spec.txt (last access: Feb 9, 2023).

16 https://blog.torproject.org/ethical-tor-research-guidelines (last access: Feb 9, 2023).



(a) Venn diagram of the five identified techniques

(b) Venn diagram of the spaces of gathering

Fig. 8. Techniques for gathering onion addresses.

The collecting methodologies can be characterized through the Venn diagram. The external portions (in set theory, the difference of each set with respect to the rest of sets) reveal the proportion of works applying a unique technique. The monolithic discoveries are primarily based on collecting onion addresses from ready-to-use repositories (16.7%). With less impact, others works use automatic crawling (13%), or relay injections (11.1%) exclusively. The isolated usage of onion search engines (1.9%) or generic ones (0%) is rare or null. The sum of these sets amounts to 42.6%, i.e., almost half of the projects apply a single gathering technique.

In the case of employing two techniques jointly (in set theory, visualized as intersections between two sets exclusively), the most used combination is repositories with Tor crawling (37%). Next, it is common to fuel Tor crawling with results of generic search engines (5.6%). Sparsely, onion search engines (1.9%) and relays (1.9%) complement Tor crawling. Therefore, the employment of techniques in pairs is present in 46.3% of the revised works, where one of them is in charge of providing the initial seed list to fuel the second technique responsible for searching or crawling.

Last, some collecting methodologies employ three techniques simultaneously. There are three works employing repositories and generic search engines to feed the Tor crawling (5.6%). Less frequently, Tor crawling is complemented with repositories and onion search engines (1.9%), repositories and relays (1.9%), and onion search engines with relays (1.9%). The combination of three techniques sums 11.1%.

4.2.2. Spaces of gathering

Analyzing the discussed usage from another perspective, Fig. 8(b) reveals that 11 projects (20.4%) employ only out-of-band techniques (identification on the surface web), 14 works (25.9%) obtain onion domains uniquely with in-band techniques (within the deep web), and 29 approaches (53.7%) rely on both spaces to discover onion addresses.

4.3. Effectiveness of gathering techniques

The various papers reviewed employ gathering techniques. Inferring their effectiveness is helpful to improve them and adopt those that collect a more significant number of onion addresses. In addition to the number of active onion addresses they discover, the time spent is critical. It is not adequate to compare the results of a collection that takes one day to one launched over six months. Therefore, the gathering effectiveness is the number of onion domains obtained divided by the gathering duration. The latter is used to normalize the number of discovered onion addresses and not directly compare absolute amounts.

Fig. 9 shows the gathering effectiveness of each technique, indicating the minimum, average and maximum values among the works specifying a collecting duration and employing each technique, either individually or in combination. The latter is a limitation of the present analysis since it is not possible to accurately assess a particular technique's effectiveness when used in conjunction with others. Therefore, this estimation should be understood as the minimum, mean and maximum effectiveness achieved in the literature when each technique is present in the collecting methodology.

According to our review, incorporating relay injection in the collecting methodology is the most effective strategy. On average, it discovers 772 onion addresses per day, indeed reaching the maximum rate of the literature of 3900 links per day in one work. As mentioned in the description of this technique, it is powerful because it constantly collects random samples of onion addresses that are bootstrapped on the Tor network.

The use of repositories also implies satisfactory performance in literature collections. On average, the samples acquire 511 onion addresses per day of execution, reaching a maximum of 3000 in one of the studies. Despite the limitations of this technique, existing large compilations contain thousands of active onion addresses that researchers are efficiently exploiting.

Third, Tor crawling and generic search engines prove acceptable for finding onion addresses. With average rates of 372 and 345 onion domains per day, these techniques have lower effectiveness, possibly due to search bias, a high degree of noisy data processing and a limit on the potential onion services reachable.

The employment of onion search engines is the least effective, with a mean of 172 onion addresses per day. In addition to the limitations of the above techniques, there may be a lack of maturity in such specific tools. Except in relay injection, the

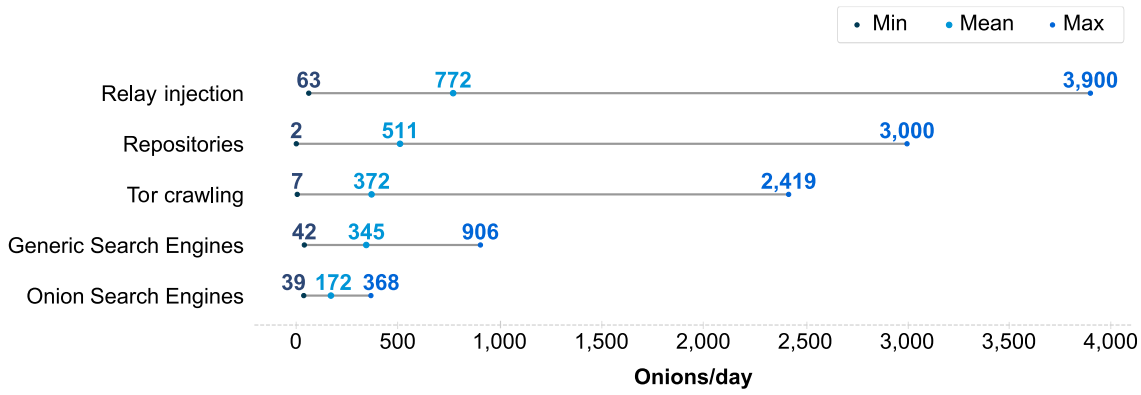


Fig. 9. Effectiveness of gathering techniques.

effectiveness of the techniques may degrade over time: it is more likely to have higher amounts of onion domains when the period is shorter. For example, the number of links downloaded on the first day in repositories will be much higher than in later days, where finding new onion addresses will be more challenging. Analogously, if not correctly configured, using Tor crawling and search engines will find it difficult to discover unseen onion addresses over time.

4.4. Overview of gathering techniques

The adoption of the techniques may depend on different factors, hence the different levels of use of the techniques. Table 2 compares the five collecting techniques considering the values of low, medium or high.

The accuracy of a technique refers to the ability to find onion addresses directly and cleanly, considering the possible overhead of needed processing, heuristics or analysis. Onion search engines, repositories and relay injection have high accuracy. The first obtains a list of indexed Tor links that contain onion addresses, the second compiles lists of onion domains, and the third infiltrates the network and captures onion addresses directly from descriptors. Tor crawling has medium accuracy as it yields HTML pages that should be parsed to match onion addresses, being likely to find them. Generic search engines are the least accurate because it has to explore conventional websites where it is challenging to find onion addresses.

The delay reflects the time involved in executing the technique and getting the onion addresses, which is especially important in operational tasks of early detection and pursuit. Onion search engines and repositories have a low delay as they are executed from the surface web and have a high accuracy in matching onion addresses. Generic search engines have a medium delay because it is necessary to visit and analyze all the resulting pages to identify onion addresses. Relay injection is the technique that takes the least time to identify new onion domains registered on the Tor network. However, the delay is medium, considering that the execution of the technique is not immediate, but involves the deployment of relays and waiting a few days to obtain the necessary flags to become a directory (HSDir). Tor crawling has high delay as it involves navigating the Tor network, following links to onion services, and establishing circuits that are not fast to set up and carry considerable latency.

Variety entails the diversity and randomness of onion services that each technique is capable of providing. Onion and generic search engines have low variety as the results are limited to indexed web-based onion services that indeed suffer from keyword selection bias and the functioning of the ranking and search algorithm. Tor crawling also has low variety because it suffers from seed bias and only visits web-based onion services explicitly linked and referenced. Repositories have medium variety because they compile contributions of unknown third parties, which may include onion sites from different sources and of different typologies, not only web services. Relay injection has high variety as it returns random samples of onion addresses.

The complexity of each technique indicates whether their use requires technical expertise or challenging deployments. Onion search engines, repositories and generic search engines have low complexity as they are easily accessible to end-users from the surface web or simple scripts with basic parsing operations. Tor crawling has medium complexity as it requires designing and implementing automatic spiders to analyze pages and navigate the Tor network. Relay injection has high complexity as it involves implementing Tor relays, introducing them in the Tor network, receiving the stable and fast flags, and inspecting received descriptors to calculate onion addresses.

The cost involved in each technique is related to the effort invested and the number of resources employed. Onion search engines, repositories and generic search engines have low costs as they can be implemented without much effort, even on a personal computer or laptop. Tor crawling has a medium cost since, in order to have acceptable performance, it is necessary to have the memory and bandwidth resources of a server. Relay injection has a high cost as it involves complex technical operations, specialized software and deployment of high-availability relays.

User privacy when using each technique may be important for people seeking onion addresses for censorship or anonymity issues, or to protect law enforcement or intelligence operations. Onion search engines, repositories, and generic search engines have low privacy as they are executed from the surface web and potentially reveal the identity of the client to Internet Service Providers (ISP), service operators, and associated third-party analytic or advertising services. On the contrary, Tor crawling and relay injection have high privacy as they are executed within the Tor network and preserve the identity of the user.

The usage of the techniques corresponds to the number of times it has been used in the different research works. As shown in Table 1 and Section 4.2, search engines and relay injection are lowly used as they are present in less than one-third of the investigations. On the other hand, repositories and Tor crawling are highly used as they are employed in more than two-thirds of the papers.

Finally, effectiveness is measured as the number of onion addresses the technique can obtain per unit of time. According to the average effectiveness calculated in Section 4.3, onion search engines have the lowest effectiveness. Repositories, Tor crawling and generic search engines manage to obtain medium effectiveness. The reference is set by relay injection with the highest average effectiveness.

Table 2
Comparison of gathering techniques.

	Accuracy	Delay	Variety	Complexity	Cost	Privacy	Usage	Effectiveness
Onion SE	●	○	○	○	○	○	○	○
Repositories	●	○	◐	○	○	○	●	◐
Generic SE	○	◐	◐	○	○	○	○	◐
Tor crawling	◐	●	○	◐	◐	●	●	◐
Relay injection	●	◐	●	●	●	●	○	●

○ low ◐ medium ● high.

5. Representativeness of literature collections

To the best of our knowledge, no previous work estimates the representativeness of onion samples and contextualized past analysis of onion services. For that purpose, the number of onion services identified by the revised papers are contrasted with respect to the official number of existing v2 onion services (ground truth).[17]

As mentioned earlier, Table 1 presents the discovered onion addresses and the subset being active at some point in reviewed experiments. In order to calculate the representativeness of collections, the active addresses gathered in the literature are compared with Tor figures of existing onion services. There are some reasons why this representativeness study should be considered as an estimation. As a consequence of the short lifetime or intermittency of onion services, Tor counts cumulative totals of onion services active at some point during the day. Similarly, the collection and status of onion services of the studies are linked to a specific point in time. In addition, those harvests that begin in one year but end in the following year have been considered on the year in which the gathering was active for most days.

Fig. 10 presents the average of existing v2 onion services from 2014[18] to 2021[19] through the dark blue bar plots. Within each bar, another bar in light blue represents the average number of discovered onion services by the studies launched in the associated year. Additionally, the dark blue line plot maps the maximum number of discovered onion domains by the most successful work of the year. The percentages indicate the proportion of the total existing onions.

Considering dark blue bar plots, Tor is expanding officially since the number of v2 onion services deployed is on a growing trend. Starting from 28,000 onion services in 2014, the size of Tor doubles every two years approximately, reaching a peak of 150,000 in 2020 and 143,000 in 2021. The latter figure is more than the five-fold increase in the original number. Outside the scope of the study, it is worth mentioning that these v2 onion services are practically extinct in 2022 in favor of version 3. In February 2023, there are 800,000 active v3 sites, according to Tor official metrics. The current spike is approximately five times the amount of v2 onion services of 2021 and motivates studying in future work on the ecosystem of version 3 [30].

Considering the average discovered onion services, the revised works reach between 2.6% and 20.5% of the entire Tor network, excepting the coverage of 44.8% in 2014. These figures mean

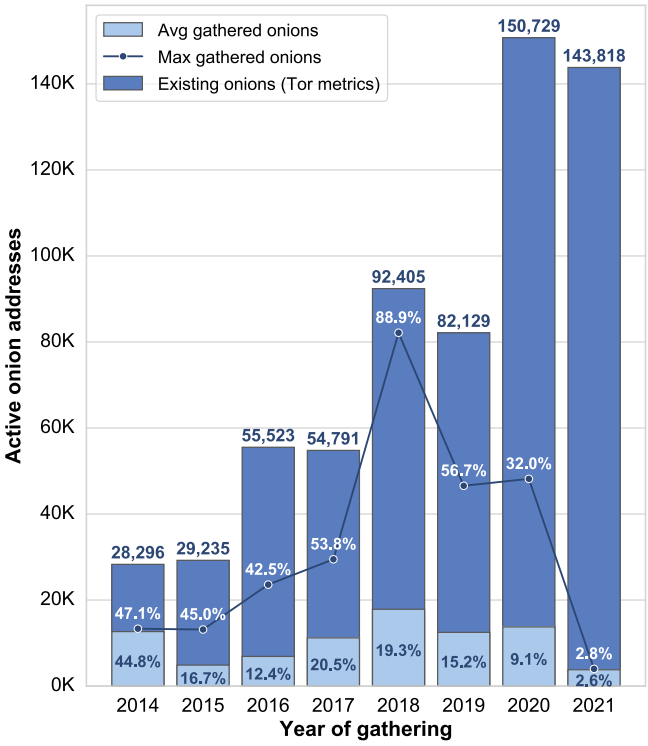


Fig. 10. Obtained number of onion addresses over time.

that, on average, only 17.5% of the landscape is reached in scientific works. These statistics are essential to dimension the mass knowledge of the Tor network: a large portion of it is not explored in scientific works.

In terms of the most representative collections each year (line plot), points indicate those works that reach the maximum number of onion services per year. These individual studies collected from 32% to 88.9% of the existing onion services, considerably more representative than the average discussed above. Three works [2,8,69] reaching more than half of the Tor network.

Lastly, while the increase in the number of onion services in Tor is notable, the representativeness of average samples is questionable. Generally, the number of onion services is doubling every two years, but literature collections do not exceed 20.5% in the last years. Despite having an ascending potential discovery space over time, studies generally employ no more than 20,000 sites. Regarding the most outstanding collections, the absolute figures improve over time, analyzing between 40,000 and 80,000 onion services in recent years. Although samples do not become more significant over time on average, the trend in the line plot indicates that more representative datasets are emerging.

6. Limitations and challenges of onion address gathering

Considering the usage and effectiveness of gathering techniques, and the representativeness of collections in literature, some current limitations and challenges are discussed next to make progress in the identification of new onion domains. It is important to highlight the strength of the Tor protocol, which manage to provide protection for the discovery of new onion services. However, from the research perspective, it is crucial to obtain representative and variety samples to know more about the dark web.

17 https://metrics.torproject.org/hidserv-dir-onions-seen.html?start=2014-01-01&end=2021-12-31 (last access: Feb 9, 2023).

18 The year since official records from Tor of the number of existing onion service are available.

19 Most recent year for which there are evidences through the analyzed articles.

6.1. Current limitations for discovering onion addresses

This study unveils current limitations when sampling the dark web, having a particular impact on the representativeness of research outcomes.

Limitation 1. Lack of gathering techniques
The review does not show specific onion gathering techniques beyond relay injection, which is out of use but the most effective. Therefore, the current usable set of harvesting techniques on which the research on the dark web is composed of traditional methods. No heuristics or search systems designed exclusively for obtaining onion addresses in the wild have been found.

Limitation 2. Bias on collecting procedures
Repositories, Tor crawling, and search engines suffer from bias of precompiled listings, third-party activities, internal functioning, or keyword selections. Research on the Tor network is based on methods that return a limited portion that could be different from onion services that are not found. In this sense, literature results may overfit samples of certain characteristics, needing more generality and representativeness. The relay injection method would be an example of a random way to harvest onion addresses.

Limitation 3. Narrow variety of discovered onion services
Although relay injection or repositories could include different kinds of onion services, those found with crawling or search engines are web-oriented. The latter means that only those that expose web services and are referenced from another page are found, leaving out (i) many onion services that run other types of services (SSH, SMTP, IRC, Bitcoin, XMPP/Jabber, RTSP, IMAP, among others [69]), (ii) those that are not referenced by other pages, or (iii) others that contain crawling defense techniques. Moreover, the hub-and-spoke topology of the Tor network does not help crawling tasks, where 30% of onion sites are only pointed once, and 87% did not reference any other service according to some evidences [29].

Limitation 4. Limited spaces of search
The usage of techniques in the literature shows that they are based on the surface web and the Tor network. However, no studies propose probing the deep web as a way to find onion addresses. This would imply obtaining Tor links that are not indexed in search engines and cannot be reached by crawling or following links. On the other hand, there are also no discoveries of onion addresses in other anonymous networks when these could form a more extensive interconnected network [2].

6.2. Challenges on onion address gathering

The limitations above lead us to consider that the bottleneck may lie in the current collecting techniques. Six challenges to tackle onion address discovery shortcomings are proposed next to improve dark web research and LEA solutions.

Challenge 1. Combination of multiple techniques
The low measured effectiveness of the current discovery gives room for improvement. In this line, although we identified up to five major gathering techniques, collections do not tend to conflate multiple techniques, and 88.9% use one or two of them.

In particular, it would be desirable to configure several techniques to enlarge the number of sources, increase summatively the effectiveness, and reduce the impact of the individual drawbacks of each of them.

Challenge 2. Unify the search surface
This paper distinguishes between out-of-band and in-band discoveries, i.e., searches performed outside and inside the Tor network, respectively. We observe that 53.7% of works monitor both spaces, but the rest lack an essential portion of the overall cyberspace. Collecting workflows can be prepared to follow links and jump continuously between both surface and dark webs if needed.

Some reviewed solutions implement a unique clearnet-to-darknet transition (e.g., collecting onion addresses from surface repositories to seed a Tor crawling process) but limited unidirectional scope. This procedure can be scaled by implementing darknet-to-clearnet transitions (e.g., automatically return to websites identified in the Tor web), generating bidirectional workflows between both spaces. Moreover, discoveries could exploit the darknet-to-darknet relationship (e.g., the connections between Tor and I2P networks) to move between different subtypes of darknets. Therefore, the ideal gathering will run loops through the surface and dark web, monitoring a unique logical surface.

Challenge 3. Constant update of the gathering process
Many discoveries visit datasets, repositories, and websites only once or iterate on non-dynamic sources to aggregate onion addresses or fuel crawling processes. Given the volatility of onion services, discoveries quickly turn into outdated searches based on dead services. Proof of this are the significant differences between the total of onion addresses collected with respect to those that are active and can be analyzed (see Table 1).

Although inactive onion addresses may also be helpful for the discovery, we propose integrating non-static data sources for more satisfactory results. Ideal workflows will continuously consider the status of seeds and intelligently search for new inputs, such as live repositories, social media streaming feeds, fresh forum posts, new datasets just published, or threat intelligence updates. The goal is to guarantee a permanent gathering compliant with the ephemerality of onion services, following the philosophy of the *cat-and-mouse* game to keep track of the most active snapshot possible of the dark web.

Challenge 4. Guide gathering to amplify outcomes
Genuinely, traditional discoveries are launched to obtain the highest quantity of onion addresses. However, methodologies do not employ specific strategies or intelligence to achieve optimal performance. Considering current crawling-based discoveries as brute-force approaches collecting a random sample s during a time window t, it is worth making an effort to design and implement heuristics to guide gathering through spaces with a greater number of addresses s', for the same period t.

For example, one of the lessons learned from the literature is that the Tor network is sparse and contains isolated cores. Theoretically, it is not the best scenario to launch a heuristics-free crawling process within Tor which is likely to go through shallow and enclosed spaces. Alternatively, the discovery could preference hub nodes, i.e., Tor services with many more links to others than the average.

Challenge 5. Exploit alternative search paths
On the one hand, only 13.3% employed generic search engines, one of the most powerful tools for surfing the web. The latter suggests that current collecting methodologies are focused on tor-specific tailored techniques, discarding generic discovery heuristics or traditional search engines that reach an ample search space. On the other hand, most discoveries are based on already existing compilations or/and Tor crawling. These techniques rely on following hyperlinks to resources that are indexed or referenced, but may exist many more addresses in non-crawleable sites (i.e., deep web).

Collecting methodologies may include Open Source Intelligence (OSINT) to collect, analyze and extract onion addresses through publicly available data [5]. For example, inspecting changes in the code of open source projects where scripts or configuration files are usually posted, visit indicator of compromise feeds, threat intelligence sources, analyze malware samples, use dorks in search engines to find pages indexed by gateways between surface and tor (.ly, .tor2web or .foundation, among others). Indeed, OSINT pipelines, which can exploit the fact that users, organizations, and online resources involved in onion services are somehow linked to each other, may enhance the traditional crawling by exploring collateral paths beyond hyperlinks. Analysts could design investigations starting from a specific point (e.g., a website, social network message, or forum post related to Tor or containing onion addresses) and iteratively jump to further correlated online resources through pivots (e.g., IP addresses, emails, hostnames, or domains).

Existing OSINT tools [5] and techniques can automate and facilitate these intelligence tasks, and it is worth adopting them for onion discovery. The concatenation of pivoting techniques may help to reach the deep web (i.e., not indexed resources or not referenced by links), thus identifying new onion addresses on the way.

Challenge 6. Evaluation of gathering effectiveness and representativeness

Our study defines and employs two metrics for evaluating discoveries. The gathering effectiveness measures the number of onion addresses discovered per time unit. This rate evaluates collecting performance and establishes a benchmark for comparing and improving collectors. Relay injection achieved 3900 onion domains per day, although this technique can no longer be adopted. Future techniques could be investigated, tested, and configured to to get closer to a gathering rate of 3000 onion addresses per day, achieved by one work through repositories. This challenge is more accessible today with a much larger number of onion services than in the past.

The gathering representativeness is the proportion of onion addresses collected with respect to official figures. The representativeness is an approximation to quantify the reached portion of the Tor network, important to contextualize the generality of research results and question the real knowledge around the dark web. This metric also helps to estimate the performance of discoveries, compare research outcomes, and quantify improvements over years.

On average, reviewed works obtain approximately 17.5% of the whole bulk of existing onion addresses, so new discoveries may aim to exceed this number. Moreover, with our historical survey in hand, the most representative experiment to this day is 88.9% (82,145 active sites). The latter may be the limit to overcome in future solutions.

7. Conclusions and future work

Tor is a network designed to provide anonymity and privacy, where online resources are defined as onion services (historically, hidden services) accessible through unmeaningful and unmemorizable addresses. Due to the lack of a standard way of obtaining these onion domains, the user needs to manually investigate and obtain existing but unknown Tor links. This gathering procedure becomes challenging when collecting extensive amounts of onion addresses to conduct, e.g., research or situational awareness.

Through a study of 54 academic works published from 2013 to 2022, this paper identified five different gathering techniques employed to this day for collecting v2 onion addresses, namely: repositories, Tor crawling, onion search engines, relay injection, and generic search engines, in order of popularity. All of them are based on the extraction of publicly available data, with special usage of existing repositories (67.8%) and crawling (73.5%). The 88.9% of reviewed solutions tended to employ only one or two techniques simultaneously. In terms of effectiveness, this work estimates the average number of onion addresses discovered per time unit by each technique, being relay injection (772 onions/day), repositories (551 onions/day), and Tor crawling (372 onions/day) the ones with best performance. Regarding the representativeness of literature works, this study extracts the number of discovered onion addresses by each research project and calculates the proportion of the Tor network explored with respect to the total existing v2 onion services (official Tor figures). On average, solutions do not reach beyond 17.5% of the total existing services, which is of limited representativeness compared to the growing number of onion services, doubling every two years.

Considering our historical analysis, three remarkable highlights are obtained: the most effective discovery rate has been 3900 onion addresses per day, the quantity of 82,145 active onions has never been exceeded, and the largest monitored Tor area has been 88.9%.

The limitations of onion address gathering are related to the lack of tailored techniques for harvesting onion addresses, bias on discovery procedures, narrow variety of obtained onion services, and limited spaces of discovery. Six major challenges are discussed to evolve the current gathering techniques and overcome the aforementioned limitations. In short, they seek to broaden the scope of the searches and include more intelligent components.

In future work, it is worth exploring the works collecting and analyzing onion services in version 3, comparing them with the official figures that Tor has been publishing since October 2021, and contrasting our results of version 2. On the other hand, we plan to develop a scalable and active platform for continuous monitoring. We hypothesize that one of the limitations is that collections are not reaching non-indexed, non-crawlable, or non-referenced resources, so we would like to design a pivoting module to reach the deep web where crawling could not access.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the GitHub repository within the article.

Acknowledgments

We thank the Spanish National Institute of Cybersecurity (Instituto Nacional de Ciberseguridad, INCIBE) for feedback on this research.

Funding

This study was funded by the Spanish Ministry of Universities (FPU18/00304), and PASTOR (Platform of Analysis of Services in TOR), Spain project, co-funded by the Instituto para la Competitividad Empresarial of the Junta de Castilla *y* León, Spain and the European Regional Development Fund.

References

[1] M. Simioni, Investigative techniques for the de-anonymization of hidden services, IEEE Secur. Priv. 19 (2) (2021) 60–64, http://dx.doi.org/10.1109/MSEC.2021.3050245.

[2] C. Cilleruelo, L. De-Marcos, J. Junquera-Sánchez, J.-J. Martínez-Herráiz, Interconnection between darknets, IEEE Internet Comput. 25 (3) (2021) 61–70, http://dx.doi.org/10.1109/MIC.2020.3037723.

[3] K. Kirkpatrick, Financing the dark web, Commun. ACM 60 (3) (2017) 21–22, http://dx.doi.org/10.1145/3037386.

[4] D.L. Huete Trujillo, A. Ruiz-Martínez, Tor hidden services: A systematic literature review, J. Cybersecur. Priv. 1 (3) (2021) 496–518, http://dx.doi.org/10.3390/jcp1030025.

[5] J. Pastor-Galindo, P. Nespoli, F. Gómez Mármol, G. Martínez Pérez, The not yet exploited Goldmine of OSINT: Opportunities, open challenges and future trends, IEEE Access 8 (2020) 10282–10304, http://dx.doi.org/10.1109/ACCESS.2020.2965257.

[6] J. Appelbaum, A. Muffett, The ".onion" Special-Use Domain Name, in: Request for Comments 7686, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7686, URL https://www.rfc-editor.org/info/rfc7686.

[7] P. Burda, C. Boot, L. Allodi, Characterizing the redundancy of DarkWeb .Onion services, in: Proceedings of the 14th International Conference on Availability, Reliability and Security, ARES '19, Association for Computing Machinery, New York, NY, USA, 2019, http://dx.doi.org/10.1145/3339252.3339273.

[8] M. Bernaschi, A. Celestini, S. Guarino, F. Lombardi, E. Mastrostefano, Spiders like Onions: On the network of tor hidden services, in: The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019, 2019, pp. 105–115, http://dx.doi.org/10.1145/3308558.3313687.

[9] P. Syverson, M. Finkel, S. Eskandarian, D. Boneh, Attacks on Onion discovery and remedies via self-authenticating traditional addresses, in: WPES 2021 - Proceedings of the 20th Workshop on Privacy in the Electronic Society, Co-Located with CCS 2021, 2021, pp. 45–52, http://dx.doi.org/10.1145/3463676.3485610.

[10] F. Platzer, A. Lux, A synopsis of critical aspects for darknet research, in: ACM International Conference Proceeding Series, Vol. 20, 2022, http://dx.doi.org/10.1145/3538969.3544444.

[11] P. Winter, A. Edmundson, L.M. Roberts, A. Dutkowska-Zuk, M. Chetty, N. Feamster, How do tor users interact with Onion services? in: Proceedings of the 27th USENIX Conference on Security Symposium, SEC '18, USENIX Association, USA, 2018, pp. 411–428.

[12] J. Pastor-Galindo, M. Zago, P. Nespoli, S. López Bernal, A. Huertas Celdrán, M. Gil Pérez, J.A. Ruipérez-Valiente, G. Martínez Pérez, F. Gómez Mármol, Spotting political social bots in Twitter: A use case of the 2019 Spanish general election, IEEE Trans. Netw. Serv. Manag. 17 (4) (2020) 2156–2170, http://dx.doi.org/10.1109/TNSM.2020.3031573.

[13] J. Pastor-Galindo, F. Gómez Mármol, G. Martínez Pérez, Profiling users and bots in Twitter through social media analysis, Inform. Sci. 613 (2022) 161–183, http://dx.doi.org/10.1016/j.ins.2022.09.046, URL https://www.sciencedirect.com/science/article/pii/S0020025522010933.

[14] V. Adewopo, B. Gonen, S. Varlioglu, M. Ozer, Plunge into the underworld: A survey on emergence of darknet, in: 2019 International Conference on Computational Science and Computational Intelligence, CSCI, 2019, pp. 155–159, http://dx.doi.org/10.1109/CSCI49370.2019.00033.

[15] E. Erdin, C. Zachor, M.H. Gunes, How to find hidden users: A survey of attacks on anonymity networks, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2296–2316, http://dx.doi.org/10.1109/COMST.2015.2453434.

[16] H. Zhang, F. Zou, A survey of the dark web and dark market research, in: 2020 IEEE 6th International Conference on Computer and Communications, ICCC, 2020, pp. 1694–1705, http://dx.doi.org/10.1109/ICCC51575.2020.9345271.

[17] B. Conrad, F. Shirazi, A survey on Tor and I2P, in: Ninth International Conference on Internet Monitoring and Protection, ICIMP2014, 2014, pp. 22–28.

[18] R.A. Haraty, B. Zantout, The TOR data communication system: A survey, in: 2014 IEEE Symposium on Computers and Communications, Vol. Workshops, ISCC, 2014, pp. 1–6, http://dx.doi.org/10.1109/ISCC.2014.6912635.

[19] M. Alsabah, I. Goldberg, Performance and security improvements for Tor: A survey, ACM Comput. Surv. 49 (2) (2016) http://dx.doi.org/10.1145/2946802.

[20] F. Yu, R. Zhou, X. Zhai, Y. Qu, G. Fei, Survey on bridge discovery in Tor, in: W. Bao, X. Yuan, L. Gao, T.H. Luan, D.B.J. Choi (Eds.), Ad Hoc Networks and Tools for IT, Springer International Publishing, Cham, 2022, pp. 317–326.

[21] S. Saleh, J. Qadir, M.U. Ilyas, Shedding light on the dark corners of the Internet: A survey of Tor research, J. Netw. Comput. Appl. 114 (March) (2018) 1–28, http://dx.doi.org/10.1016/j.jnca.2018.04.002, arXiv:1803.02816.

[22] J. Saleem, R. Islam, M.A. Kabir, The anonymity of the dark web: A survey, IEEE Access 10 (2022) 33628–33660, http://dx.doi.org/10.1109/ACCESS.2022.3161547.

[23] I. Karunanayake, N. Ahmed, R. Malaney, R. Islam, S.K. Jha, De-anonymisation attacks on Tor: A survey, IEEE Commun. Surv. Tutor. (2021) 1, http://dx.doi.org/10.1109/COMST.2021.3093615.

[24] S. Nepal, S. Dahal, S. Shin, Deanonymizing schemes of hidden services in Tor network: A survey, in: 2015 International Conference on Information Networking, ICOIN, 2015, pp. 468–473, http://dx.doi.org/10.1109/ICOIN.2015.7057949.

[25] L. Basyoni, N. Fetais, A. Erbad, A. Mohamed, M. Guizani, Traffic analysis attacks on Tor: A survey, in: 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICIoT, 2020, pp. 183–188, http://dx.doi.org/10.1109/ICIoT48696.2020.9089497.

[26] Y. Sun, M. Apostolaki, H. Birge-Lee, L. Vanbever, J. Rexford, M. Chiang, P. Mittal, Securing Internet applications from routing attacks, Commun. ACM 64 (6) (2021) 86–96, http://dx.doi.org/10.1145/3429775.

[27] C. Guitton, A review of the available content on Tor hidden services: The case against further development, Comput. Hum. Behav. 29 (6) (2013) 2805–2815, http://dx.doi.org/10.1016/j.chb.2013.07.031.

[28] G. Owenson, S. Cortes, A. Lewman, The darknet's smaller than we thought: The life cycle of Tor hidden services, Digit. Investig. 27 (2018) 17–22, http://dx.doi.org/10.1016/j.diin.2018.09.005.

[29] K.P. O'Keeffe, V. Griffith, Y. Xu, P. Santi, C. Ratti, The darkweb: A social network anomaly, in: D. Braha, M.A.M. de Aguiar, C. Gershenson, A.J. Morales, L. Kaufman, E.N. Naumova, A.A. Minai, Y. Bar-Yam (Eds.), Unifying Themes in Complex Systems X, Springer International Publishing, Cham, 2021, pp. 335–347.

[30] T. Hoeller, M. Roland, R. Mayrhofer, On the state of V3 Onion services, in: Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet, FOCI '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 50–56, http://dx.doi.org/10.1145/3473604.3474565.

[31] A. Biryukov, I. Pustogarov, R.-P. Weinmann, Trawling for Tor hidden services: Detection, measurement, deanonymization, in: 2013 IEEE Symposium on Security and Privacy, 2013, pp. 80–94, http://dx.doi.org/10.1109/SP.2013.15.

[32] C. Guitton, A review of the available content on Tor hidden services: The case against further development, Comput. Hum. Behav. 29 (6) (2013) 2805–2815, http://dx.doi.org/10.1016/j.chb.2013.07.031.

[33] M. Spitters, S. Verbruggen, M. Van Staalduinen, Towards a comprehensive insight into the thematic organization of the Tor hidden services, in: 2014 IEEE Joint Intelligence and Security Informatics Conference, 2014, pp. 220–223, http://dx.doi.org/10.1109/JISIC.2014.40.

[34] G. Owen, N. Savage, Empirical analysis of Tor hidden services, IET Inf. Secur. 10 (3) (2016) 113–118, http://dx.doi.org/10.1049/iet-ifs.2015.0121.

[35] P. Liu, X. Wang, X. He, C. Li, S. Cao, L. He, J. Zhu, A quantitative model for analysis and evaluation of Tor hidden service discovery, in: G. Sun, S. Liu (Eds.), Advanced Hybrid Information Processing, Springer International Publishing, Cham, 2018, pp. 70–77.

[36] A. Kwon, M. AlSabah, D. Lazar, M. Dacier, S. Devadas, Circuit fingerprinting attacks: Passive deanonymization of Tor hidden services, in: 24th USENIX Security Symposium (USENIX Security 15), USENIX Association, Washington, D.C., 2015, pp. 287–302.

[37] A. Panchenko, A. Mitseva, M. Henze, F. Lanze, K. Wehrle, T. Engel, Analysis of fingerprinting techniques for Tor hidden services, in: Proceedings of the 2017 on Workshop on Privacy in the Electronic Society, WPES '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 165–175, http://dx.doi.org/10.1145/3139550.3139564.

[38] S. Matic, P. Kotzias, J. Caballero, CARONTE: Detecting location leaks for deanonymizing Tor hidden services, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 1455–1466, http://dx.doi.org/10.1145/2810103.2813667.

[39] K. Li, P. Liu, Q. Tan, J. Shi, Y. Gao, X. Wang, Out-of-band discovery and evaluation for Tor hidden services, in: Proceedings of the 31st Annual ACM Symposium on Applied Computing, SAC '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 2057–2062, http://dx.doi.org/10.1145/2851613.2851798.

[40] Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, A closer look at eclipse attacks against Tor hidden services, in: 2017 IEEE International Conference on Communications, ICC, 2017, pp. 1–6, http://dx.doi.org/10.1109/ICC.2017.7996832.

[41] H.A. Jawaheri, M.A. Sabah, Y. Boshmaf, A. Erbad, Deanonymizing Tor hidden service users through Bitcoin transactions analysis, Comput. Secur. 89 (2020) 101684, http://dx.doi.org/10.1016/j.cose.2019.101684.

[42] R. Overdorf, M. Juarez, G. Acar, R. Greenstadt, C. Diaz, How unique is your .Onion? An analysis of the fingerprintability of Tor Onion services, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, Association for Computing Machinery, New

York, NY, USA, 2017, pp. 2021–2036, http://dx.doi.org/10.1145/3133956.3134005.

[43] I. Sanchez-Rola, D. Balzarotti, I. Santos, The Onions have eyes: A comprehensive structure and privacy analysis of Tor hidden services, in: Proceedings of the 26th International Conference on World Wide Web, WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017, pp. 1251–1260, http://dx.doi.org/10.1145/3038912.3052657.

[44] A. Mitseva, A. Panchenko, F. Lanze, M. Henze, K. Wehrle, T. Engel, POSTER: Fingerprinting Tor hidden services, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 1766–1768, http://dx.doi.org/10.1145/2976749.2989054.

[45] A. Kinder, K.-K.R. Choo, N.-A. Le-Khac, Towards an automated process to categorise Tor's hidden services, in: N.-A. Le-Khac, K.-K.R. Choo (Eds.), Cyber and Digital Forensic Investigations: A Law Enforcement Practitioner's Perspective, Springer International Publishing, Cham, 2020, pp. 221–246, http://dx.doi.org/10.1007/978-3-030-47131-6_10.

[46] M.W. Al Nabki, E. Fidalgo, E. Alegre, 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, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 35–43.

[47] A.T. Zulkarnine, R. Frank, B. Monk, J. Mitchell, G. Davies, 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, http://dx.doi.org/10.1109/ISI.2016.7745452.

[48] M. Faizan, R.A. Khan, A two-step dimensionality reduction scheme for dark web text classification, in: Y.-C. Hu, S. Tiwari, M.C. Trivedi, K.K. Mishra (Eds.), Ambient Communications and Computer Systems, Springer Singapore, Singapore, 2020, pp. 303–312.

[49] G. Cherubin, J. Hayes, M. Juarez, Website fingerprinting defenses at the application layer, Proc. Priv. Enhan. Technol. 2017 (2) (2017) 186–203, http://dx.doi.org/10.1515/popets-2017-0023.

[50] S. Ghosh, A. Das, P. Porras, V. Yegneswaran, A. Gehani, Automated categorization of Onion sites for analyzing the darkweb ecosystem, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Vol. Part F1296, 2017, pp. 1793–1802, http://dx.doi.org/10.1145/3097983.3098193.

[51] J. Dalins, C. Wilson, M. Carman, Criminal motivation on the dark web: A categorisation model for law enforcement, Digit. Investig. 24 (2018) 62–71.

[52] M. Bernaschi, A. Celestini, S. Guarino, F. Lombardi, Exploring and analyzing the Tor hidden services graph, ACM Trans. Web 11 (4) (2017) http://dx.doi.org/10.1145/3008662.

[53] M. Chen, X. Wang, J. Shi, Y. Gao, C. Zhao, W. Sun, Towards comprehensive security analysis of hidden services using binding guard relays, in: J. Zhou, X. Luo, Q. Shen, Z. Xu (Eds.), Information and Communications Security, Springer International Publishing, Cham, 2020, pp. 521–538.

[54] J. Park, Y. Lee, POSTER: Probing Tor hidden service with dockers, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 2571–2573, http://dx.doi.org/10.1145/3133956.3138849.

[55] M.W. Al-Nabki, E. Fidalgo, E. Alegre, L. Fernández-Robles, ToRank: Identifying the most influential suspicious domains in the Tor network, Expert Syst. Appl. 123 (2019) 212–226, http://dx.doi.org/10.1016/j.eswa.2019.01.029.

[56] X. Zhang, K.P. Chow, A framework for dark web threat intelligence analysis, in: I.R.M. Association (Ed.), Cyber Warfare and Terrorism: Concepts, Methodologies, Tools, and Applications, IGI Global, Hershey, PA, USA, 2020, pp. 266–276, http://dx.doi.org/10.4018/978-1-7998-2466-4.ch017.

[57] C. Yoon, K. Kim, Y. Kim, S. Shin, S. Son, Doppelgängers on the dark web: A large-scale assessment on phishing hidden web services, in: The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019, 2019, pp. 2225–2235, http://dx.doi.org/10.1145/3308558.3313551.

[58] K.P. O'Keeffe, V. Griffith, Y. Xu, P. Santi, C. Ratti, The darkweb: A social network anomaly, in: D. Braha, M.A.M. de Aguiar, C. Gershenson, A.J. Morales, L. Kaufman, E.N. Naumova, A.A. Minai, Y. Bar-Yam (Eds.), Unifying Themes in Complex Systems X, Springer International Publishing, Cham, 2021, pp. 335–347, http://dx.doi.org/10.1007/978-3-030-67318-5_22.

[59] B. Monk, J. Mitchell, R. Frank, G. Davies, Uncovering Tor: An examination of the network structure, in: Z. Yan (Ed.), Secur. Commun. Netw. 2018 (2018) 4231326, http://dx.doi.org/10.1155/2018/4231326.

[60] V. Duddu, D. Samanta, D.V. Rao, Fuzzy graph modelling of anonymous networks, Adv. Intell. Syst. Comput. 1222 AISC (2021) 432–444, http://dx.doi.org/10.1007/978-3-030-52190-5_31.

[61] M. Steinebach, M. Schäfer, A. Karakuz, K. Brandl, Y. Yannikos, Detection and analysis of Tor Onion services, in: Proceedings of the 14th International Conference on Availability, Reliability and Security, ARES '19, Association for Computing Machinery, New York, NY, USA, 2019, http://dx.doi.org/10.1145/3339252.3341486.

[62] M. Faizan, R.A. Khan, Exploring and analyzing the dark web: A new Alchemy, First Monday 24 (5) (2019) http://dx.doi.org/10.5210/fm.v24i5.9473.

[63] J. Park, H. Mun, Y. Lee, Improving Tor hidden service Crawler performance, in: 2018 IEEE Conference on Dependable and Secure Computing, DSC, 2018, pp. 1–8, http://dx.doi.org/10.1109/DESEC.2018.8625103.

[64] A. Mani, T. Wilson-Brown, R. Jansen, A. Johnson, M. Sherr, Understanding Tor usage with privacy-preserving measurement, in: Proceedings of the Internet Measurement Conference 2018, IMC '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 175–187, http://dx.doi.org/10.1145/3278532.3278549.

[65] J. Lee, Y. Hong, H. Kwon, J. Hur, Shedding light on dark Korea: An in-depth analysis and profiling of the dark web in Korea, in: I. You (Ed.), Information Security Applications, Springer International Publishing, Cham, 2020, pp. 357–369, http://dx.doi.org/10.1007/978-3-030-39303-8_27.

[66] S. Takaaki, I. Atsuo, Dark web content analysis and visualization, in: Proceedings of the ACM International Workshop on Security and Privacy Analytics, IWSPA '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 53–59, http://dx.doi.org/10.1145/3309182.3309189.

[67] Y. Kawaguchi, S. Ozawa, Exploring and identifying malicious sites in dark web using machine learning, in: T. Gedeon, K.W. Wong, M. Lee (Eds.), Neural Information Processing, Springer International Publishing, Cham, 2019, pp. 319–327, http://dx.doi.org/10.1007/978-3-030-36718-3_27.

[68] M. Zabihimayvan, R. Sadeghi, D. Doran, M. Allahyari, A broad evaluation of the Tor English content ecosystem, in: Proceedings of the 10th ACM Conference on Web Science, WebSci '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 333–342, http://dx.doi.org/10.1145/3292522.3326031.

[69] M. Steinebach, M. Schäfer, A. Karakuz, K. Brandl, Detection and analysis of Tor Onion services, J. Cyber Secur. Mobil. 9 (2020) 141–174, http://dx.doi.org/10.13052/jcsm2245-1439.915.

[70] F. Barr-Smith, J. Wright, Phishing with a darknet: Imitation of Onion services, in: 2020 APWG Symposium on Electronic Crime Research (ECrime), 2020, pp. 1–13, http://dx.doi.org/10.1109/eCrime51433.2020.9493262.

[71] A. Feal, P. Vallina, J. Gamba, S. Pastrana, A. Nappa, O. Hohlfeld, N. Vallina-Rodriguez, J. Tapiador, Blocklist babel: On the transparency and dynamics of open source blocklisting, IEEE Trans. Netw. Serv. Manag. 18 (2) (2021) 1334–1349, http://dx.doi.org/10.1109/TNSM.2021.3075552.

[72] M. Zabihimayvan, D. Doran, A first look at references from the dark to the surface web world: A case study in Tor, Int. J. Inf. Secur. 21 (4) (2022) 739–755, http://dx.doi.org/10.1007/s10207-022-00580-z.

[73] N. Ferry, T. Hackenheimer, F. Herrmann, A. Tourette, Methodology of dark web monitoring, in: 2019 11th International Conference on Electronics, Computers and Artificial Intelligence, ECAI, 2019, pp. 1–7, http://dx.doi.org/10.1109/ECAI46879.2019.9042072.

[74] A. Alharbi, M. Faizan, W. Alosaimi, H. Alyami, A. Agrawal, R. Kumar, R.A. Khan, 36. Exploring the topological properties of the Tor dark web, IEEE Access 9 (2021) 21746–21758, http://dx.doi.org/10.1109/ACCESS.2021.3055532.

[75] S.M.M. Monterrubio, J.E.A. Naranjo, L.I.B. Lopez, A.L.V. Caraguay, Black widow crawler for TOR network to search for criminal patterns, in: Proceedings - 2021 2nd International Conference on Information Systems and Software Technologies, ICI2ST 2021, 2021, pp. 108–113, http://dx.doi.org/10.1109/ICI2ST51859.2021.00023.

[76] M. Steinebach, S. Zenglein, K. Brandl, Phishing detection on Tor hidden services, Forensic Sci. Int. Digit. Investig. 36 (2021) 301117, http://dx.doi.org/10.1016/j.fsidi.2021.301117.

[77] F. Platzer, A. Lux, A synopsis of critical aspects for darknet research, in: ACM International Conference Proceeding Series, no. 20, 2022, http://dx.doi.org/10.1145/3538969.3544444.

[78] A. Alharbi, M. Faizan, W. Alosaimi, H. Alyami, M. Nadeem, S.A. Khan, A. Agrawal, R.A. Khan, A link analysis algorithm for identification of key hidden services, Comput. Mater. Contin. 68 (1) (2021) 877–886, http://dx.doi.org/10.32604/cmc.2021.016887.

[79] A.H.M. Alaidi, R.M. Alairaji, H.T.H.S. ALRikabi, I.A. Aljazaery, S.H. Abbood, Dark web illegal activities crawling and classifying using data mining techniques, Int. J. Interact. Mob. Technol. 16 (10) (2022) 122–139, http://dx.doi.org/10.3991/ijim.v16i10.30209.

[80] V. Nair, J.M. Kannimoola, A tool to extract onion links from Tor hidden services and identify illegal activities, in: S. Smys, V.E. Balas, R. Palanisamy (Eds.), Inventive Computation and Information Technologies, Springer Na-

ture Singapore, Singapore, 2022, pp. 29–37, http://dx.doi.org/10.1007/978-981-16-6723-7_3.

[81] K. Turk, S. Pastrana, B. Collier, A tight scrape: Methodological approaches to cybercrime research data collection in adversarial environments, in: Proceedings - 5th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2020, 2020, pp. 428–437, http://dx.doi.org/10.1109/EuroSPW51379.2020.00064.

Javier Pastor-Galindo received B.Sc. and M.Sc. degrees in Computer Science from the University of Murcia, Spain. He is currently a Ph.D. student in the same institution with an FPU contract granted by the Spanish Ministry of Universities. His research interests focus on open source intelligence (OSINT), security, and privacy. Contact: javierpg@um.es

Félix Gómez Mármol received a M.Sc. and Ph.D. in Computer Science from the University of Murcia. He is currently Associate Professor in the Department of Information and Communications Engineering at the University of Murcia, Spain. His research interests include cybersecurity, internet of things, machine learning and bio-inspired algorithms. Contact: felixgm@um.es

Gregorio Martínez Pérez received a Ph.D. degree in Computer Science at the University of Murcia, where he is Full Professor since 2014. His scientific activity is mainly devoted to cybersecurity and data science. He is working on different national and European IST research projects related to these topics, being Principal Investigator for UMU in most of them. Contact: gregorio@um.es