AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Approved by AICTE, New Delhi, Recognized by Govt. of Maharashtra,
Affiliated to Savitribai Phule Pune University and recognized 2(f) and 12(B) by UGC
(Id.No. PU / PN/ Engg. / 093 (1992)
(Accredited by NAAC with grade A+)

A PROJECT BASED LEARNING REPORT ON

"Identification of Network Structures through Exploration the Topological Properties of the Dark Web"

S.E. (COMPUTER)

SUBMITTED BY

PRATIK PINGALE (19CO056)
ROHAN DAYAL (19CO060)
SAGNIK ROY (19CO061)
YASH TATIYA (19CO067)

UNDER THE GUIDANCE OF

Prof. Vidya Waykule

(Academic Year: 2020-2021)

Abstract

The web graph can be used to get insight into the internal structure and connectivity of the Tor dark web. This paper analyzes the internal structure of the Tor dark web graph and examines the presence of bow-tie structure as found in the World Wide Web. The web graph is generated from the data collected by the Python crawler customized to scrape data from the Tor dark web. Each of the nodes in the graph represents an individual Tor hidden service, and an edge denotes the hyperlink from one hidden service to the other. Various graph metrics are then computed and analyzed for both directed and undirected graphs using the Python NetworkX package.

It was found that most of the nodes of the graph have in-degree and out-degree less than ten. The presence of power-law in degree distribution could neither be confirmed nor denied. The Tor web graph is sparse with a few connected pairs of nodes. Like the surface web, the dark web can also be decomposed into a bow-tie structure though with small component sizes. Several important and well-known websites on the surface web have incoming links from the dark web. Moreover, the Tor network also shows the characteristics of small-world and scale-free networks.

Contents

Chapter 1
INTRODUCTION

A web graph can be described as a collection of vertices and edges where a single vertex depicts a web page, and an edge between two vertices is a directed hyperlink from a web page to the other. All the vertices and edges belong to the World Wide Web or the Internet. The study of web graph may help in identifying the underlying structure of the Internet of how the different web pages are linked to each other. They help in developing efficient data mining techniques and better crawling strategies. The exponential growth in the size of the World Wide Web has attracted the scientific community to investigate its graph structural properties with various aspects[1][4].

Dark Web is a portion of the Internet that can only be accessed using sophisticated routing techniques. The anonymity provided to the dark web users have been misused to carry out illicit activities online[5]. The law enforcement agencies have expressed concerns regarding the criminal usage of the dark web[6]. Moreover, it also becomes challenging given the ever-changing structure and high churn in the dark web ecosystem[7]. There is a need to study the structure of the dark web to get insight into the criminal activity on the dark web. To the best of our knowledge, only limited work has been done on the graph structure of the dark web as the majority of studies have focused on the nature of content present in the dark web and its legal acceptability.

This report[0] has tried to fill this gap by exploring the directed and undirected Tor web graph at the domain level. The data has been collected from the dark web using the tailored made web crawler. The Python NetworkX package has been used to generate the graphs from the collected data. The web graph consists of 48,174 vertices or nodes and 103526 edges. Various graph metrics have been calculated and analyzed for the graph to identify the importance of different hidden services. The connectivity within graph and its structural shape are also investigated. Also, the connectivity to surface web from the dark web is studied.

1.1 State Of Art

Deep analysis of *topological web structure* of Dark Web have widely aided research organizations world wide along with Cybersecurity to disentangle nodes or central tendency of Dark web that give rise or illicit activities and those that sneak out to fiddle with surface web.

Many institution like Taif University Researchers, Taif, Saudi Arabia have already started showing interest in this field.

1.2 Overview

The dark web is being used for both good and corrupt activities. Users with good intentions can leverage this anonymous platform to express their thoughts and views openly without any censorship[8]. The law enforcement organizations and government bodies may use the dark web to secretly carry out their missions [9].

The anonymity provided to the dark web users have been misused to carry out illicit activities online[5]. There is a need to study the structure of the dark web to get insight into the criminal activity on the dark web[6].

This project has tried to fill this gap by exploring the directed and undirected Tor web graph at the domain level. The data has been collected from the dark web using the tailored made web crawler. The web graph consists of 48,174 vertices or nodes and 103526 edges. Various graph metrics have been calculated and analyzed for the graph to identify the importance of different hidden services. The connectivity within graph and its structural shape are also investigated. Also, the connectivity to surface web from the dark web is studied.

1.2.1 Basics of Topic

The dark web[20] is the World Wide Web content that exists on darknets: overlay networks that use the Internet but require specific software, configurations, or authorization to access. Through the dark web, private computer networks can communicate and conduct business anonymously without divulging identifying information, such as a user's location. The dark web forms a small part of the deep web, the part of the Web not indexed by web search engines, although sometimes the term deep web is mistakenly used to refer specifically to the dark web.

The darknets which constitute the dark web include small, friend-to-friend peer-to-peer networks, as well as large, popular networks such as Tor, Freenet, I2P, and Riffle operated by public organizations and individuals. Users of the dark web refer to the regular web as Clearnet due to its unencrypted nature. The Tor dark web or onionland uses the traffic anonymization technique of onion routing under the network's top-level domain suffix .onion.

The two opposite usage of the dark web platform creates a state of dilemma for law enforcement agencies to undertake a strict action. Hence there is a need to get insight into the internal structure of the dark web, which may help in the better monitoring of activities being carried out there.

The web graph of the dark web may help in identifying the central nodes that hold the network structure. It may also help uncover how the network structure supports the illicit activities.

1.3 Problem Statement

Construct and analyze an idea to track hidden Dark Web services and prepare a report on the same mentioning every detail required to be known.
The topological properties of the dark web graph can be used for analysis.

1.4 Motivation

We found a network impacted (but not dominated) by illicit commerce and money laundering, but almost completely devoid of violence and extremism. In short, criminality on this 'dark web' is based more upon greed and desire, rather than any particular political motivations.

We introduce the Tor-use Motivation Model (TMM), a two-dimensional classification methodology specifically designed for use within a law enforcement context. The TMM achieves greater levels of granularity by explicitly distinguishing site content from motivation, providing a richer labeling schema without introducing inefficient complexity or reliance upon overly broad categories of relevance.

1.4.1 Topic Reasoning

The downside of the dark web platform is that it is being misused by criminals and fraudsters. Many of the illicit trades that were previously carried out on the streets have now been shifted to the dark web.

The sale of illegal drugs being one of the other grey activities carried out on the dark web. Other controversial activities include child abuse, hate, violent and gory content, credit card frauds, pirated software, unethical hacking guides, etc.

Chapter 2
LITERATURE REVIEW

This section begins with the description of related work on the analysis of the Tor dark web graph followed by Basic System Architecture.

A critical study is by Bernaschi 2017[10], where they collected data using the BUbiNG crawler for the analysis of graphs at the page level, host level, and service level. They have reported graph metrics like degree distributions along with the identification of connected components of the graph and their importance in the integration of the entire graph structure. Both the directed and undirected graphs were analyzed. They also performed the semantic analysis of their data to relate it with their findings of the topology of the web graph.

A few of the early studies to determine the topological properties of the web graph were by Kumar 1999[11] and Barabási 2000[12] where the former reports the inverse polynomial distribution being followed by the in-degree and out-degree distributions while the latter claimed the power law. Kleinberg 1999[1] proposed in their paper two algorithms for web graph, which they ran on their data set to obtain measures for better search. Based on their observations, they also propose a family of random graph models that suited their measurements.

Broder 2000[2] analyzed the web graph generated from a large data set of 200 million pages with 1.5 billion links provided by AltaVista crawl. They found an important result that the web graph bears a structure similar to that of a bow tie having six components. According to them, the in-degree and out-degree distributions follow the power-law and affirm it as the underlying property of the Web. They also claim that the measurements of the graph properties do not have any effect due to the change in crawled data.

A similar study by Donato 2007[13] conducted their experiment on the data set provided by the Web Base project. They reconrm the presence of power-law

exhibited by in degree distribution as observed in previous studies. They also found that there exists a small correlation between the page rank and in-degree distribution. The bipartite cliques were also found in their analysis of the web graph.

Lehmberg 2014[3] analyzed the surface web graph aggregated at the page level domain (PLD). Their coverage of more than three billion pages and their hyperlinks was collected in 2012. Once again, the in-degree distribution was following a power law with exponent 2.4 in their findings. They confirm the presence of a bow-tie structure in the PLD graph as set forth by Broder 2000[2], for the page graph. Based on the overall observations, they suggest a hypothetical structure for a PLD graph called the Two-Layer Model which consists of a Low Degree Layer that contains sparsely connected websites and a High Degree Layer containing densely connected websites with high in-degree.

Meusel 2015[4] performed graph analysis at different aggregation levels on the data set collected by the Common Crawl Foundation in 2012. Comparing their results with the previous studies, they reported a significant increase in the average in-degree and the connectivity of the page graph. As the existence of a large strongly connected component in page and PLD graph was confirmed by the previous studies discussed above, its existence in the remaining host graph was also confirmed by them in their work.

Serrano 2007[14] performed a comparative study of different crawl mechanisms and their effects on the various graph-theoretic properties. They put forward the need for a framework for the analysis of sampling biases that occurs in different crawl techniques. According to them, it will help in determining the exact structure of the Web, which varies due to the change in the crawling mechanism. The process of extraction of data from the dark web is similar to that on the surface web as they both are built using a hypertext markup language (HTML).

A survey study has identified two distinct approaches for extracting data from the web[18]. The first approach relies on the structural properties of the web pages that are built upon HTML. Such techniques consider the tree-like structure of the HTML tags and extract required data embedded within the tags. These methods give better performance on the web pages having similar HTML markup but may not be effective on web pages with different HTML structures. The second

approach is based on machine learning techniques where a model is trained to predict the label of a particular web page. In practice, machine learning methods usually need a large amount of training data to produce a good performance. However, such a method may not be able to predict the new instances that were not present in the training set.

To overcome the challenges associated with web data extraction, an extractor is presented based on supervised and unsupervised methods to extract the specific product description from the websites[19]. An unsupervised technique called visual validation was combined with a supervised classifier to produce a versatile extractor that works on a variety of websites. The extractor was implemented on a publicly available corpus. The results on the corpus showed that the visual validation technique can improve the extractor performance when trained on the visual features.

Chapter 3
BASIC SYSTEM ARCHITECTURE

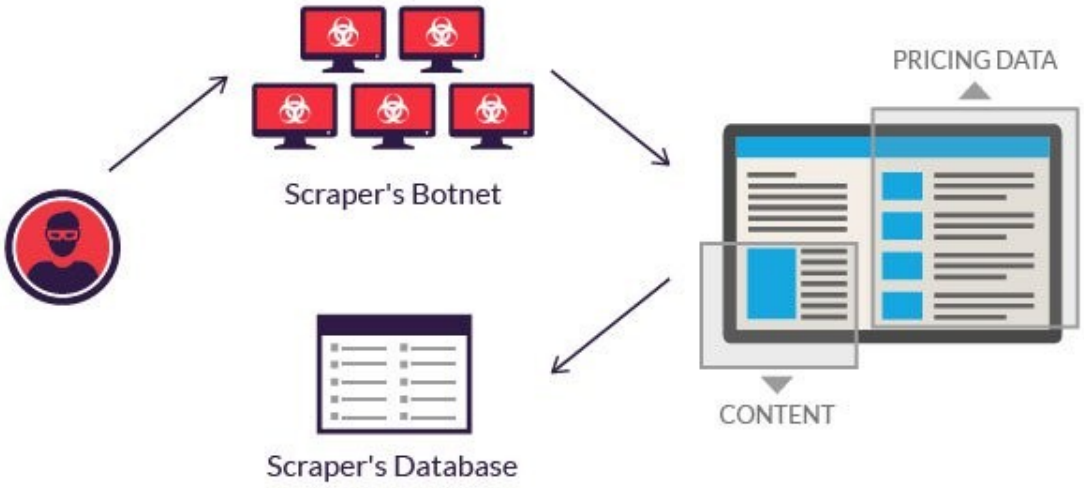

Figure 1: Scraping Onion Links using Web Crawler

We investigate the structure of the dark web at the domain level, so all the sub-domains of a particular domain are aggregated under it and are denoted as a single individual node of a graph.

To be specific, consider a hidden service domain *XYZ.onion* having two sub-domains *en.XYZ.onion* and *ch.XYZ.onion*, both of these sub-domains and their internal web pages are treated as a single node of graph identified by their domain name. Hence any edge from node X to node Y in the graph represents that there is

a hyperlink within the web page of domain X pointing to a web page contained in domain Y.

For the construction of the graph, the collected domain web pages are searched for the links in HTML <a > tag to other pages and saved in a separate file with the help of Python code.

The hyperlinks that point within the web pages of the same domain are discarded. Once the links of all the domains are obtained, an adjacency list for the graph has been generated. This adjacency list is supplied to the Python NetworkX library for the generation of the directed web graph. The undirected version of the graph was also created from the directed graph.

Chapter 4
DESIGN AND ANALYSIS

BOW-TIE DECOMPOSITION

Definition 3. Let $G = (V, A)$ be a digraph and let S be a strongly connected component of G. The bow-tie decomposition of G with respect to S consists of the following sets of nodes:

$$SCC = S$$

$$IN = \{v \epsilon V - S \,|\, S\, is\, reachable\, from\, v\}$$

$$OUT = \{v \epsilon V - S \,|\, v\, is\, reachable\, from\, S\}$$

$$TUBES = \{v \epsilon V - S - IN - OUT \,|\; v\, is\, reachable\, from\, IN\, and\; OUT\, is\, reachable\, from\, v\}$$

$$INTENDRILS = \{v \epsilon V - S \,|\; v\, is\, reachable\, from\, IN\, and\; OUT\, is\, not\, reachable\, from\, v\}$$

$$OUTTENDRILS = \{v \epsilon V - S \,|\; v\, is\, not\, reachable\, from\, IN\, and\; OUT\, is\, reachable\, from\, v\}$$

$$OTHERS = V - S - IN - OUT - TUBES - INTENDRILS - OUTTENDRILS$$

Figure 2: Bow-Tie basic Definitions

Bow-tie Decomposition Algorithm: Let $G = (V, A)$ be a digraph and let S be a strongly connected component of G. Then the bow-tie decomposition of G with respect to S may be computed as follows:

1) Set $SCC = S$.
2) Choose $v \epsilon S$. Then $OUT = DFS_G(v) - S$.
3) Choose $v \epsilon S$. Then $IN = DFS_{G^T}(v) - S$.
4) For each $v \epsilon V - S - IN - OUT$, compute the following two Boolean values:

$$IRV = (IN \cap DFS_{G^T}(v) \neq \phi)$$

$$VRO = (OUT \cap DFS_G(v) \neq \phi)$$

Then, since IRV answers the question of whether IN can reach v and VRO answers the question of whether v can reach OUT:

i) IRV and $VRO \Rightarrow v \epsilon TUBES$;

ii) IRV and not $VRO \Rightarrow v \epsilon INTENDRILS$;

iii) not IRV and $VRO \Rightarrow v \epsilon OUTTENDRILS$;

iv) not IRV and not $VRO \Rightarrow v \epsilon OTHER$.

Figure 3: Bow-Tie Decomposition Algorithm

The bow-tie structure of the World Wide Web as proposed by Broder 2000[2] consists of six mutually disjoint sets of nodes called components which are described as follows:

LSCC: The Largest Strongly Connected Component of the graph and is also called CORE.

IN: The set of nodes excluding those in LSCC and are reachable to CORE.

OUT: The set of nodes excluding those in LSCC and are reachable from CORE.

TUBES: The set of nodes excluding those in LSCC, IN and OUT such that they lie in between the directed path from IN to OUT.

TENDRILS: The set of nodes excluding all the above-listed nodes such that they are reachable from IN or can reach to OUT.

DISCONNECTED: The set of all the remaining nodes.

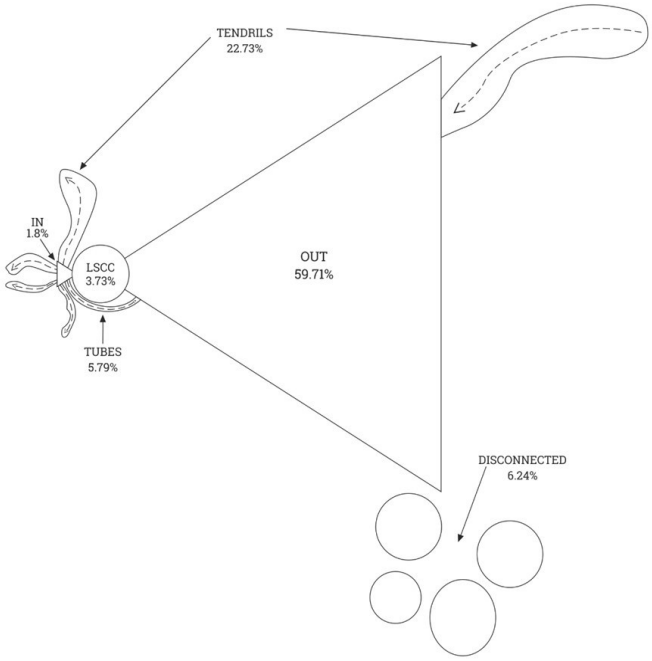

Figure 4: Bow-Tie Decomposition of Dark Web

To examine the presence of the bow-tie structure in the graph, the size of its various components are computed. As already discussed in the preceding sub-section, S1 is the *LSCC* having 1796 nodes; all the other components are calculated using *LSCC*. Fig. 4 shows the results obtained. It can be seen that the OUT is much bigger as compared to the IN. The big size of *OUT* is because there are many of the

largest out-degree nodes in CORE that connect to the majority of non *CORE* nodes. All the nodes of *DISCONNECTED* are the isolated nodes. The *INTENDRILS* is the subset of *TENDRILS* having nodes that are reachable from IN and *OUTTENDRILS* contains nodes from *TENDRILS* that can reach to *OUT*. Of 22.73% of *TENDRILS*, 20.15% belongs to *INTENDRILS* and the remaining 2.58% is *OUTTENDRILS*.

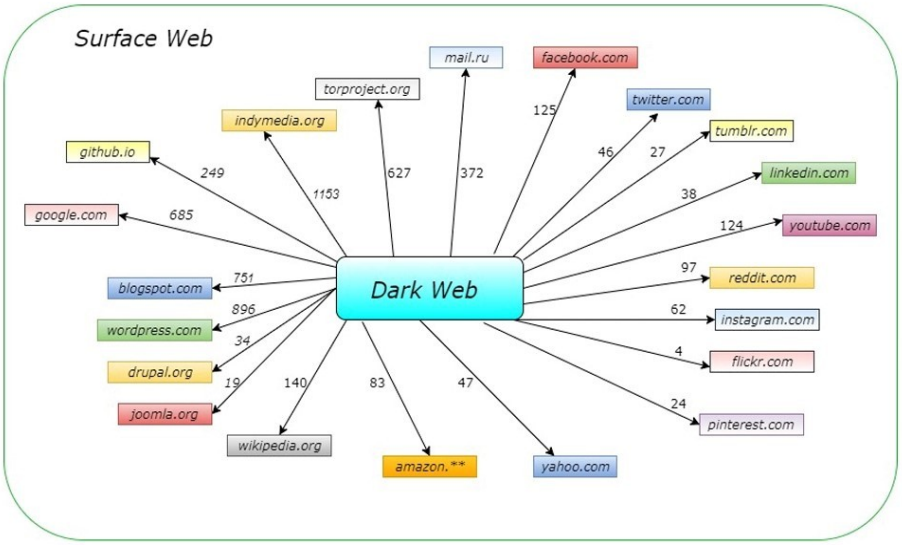

Figure 5: Connectivity of Dark Web to the Surface Web

A total of 19386 outgoing links were obtained from Tor dark web towards surface web and of which, 5406 links were representing URLs of unique websites. Besides these, another 954 URLs were found ending with suffix .i2p that represents dark web sites running through the I2P network.2 Fig. 5 shows the pictorial representation of the data collected. For the sake of simplicity, only the domains having an in-degree greater than ten are shown. All the remaining domains are not included. Note that the weight on the edges denotes the in-degree of the websites from the dark web and is a sum of all the links to each of the sub-domains of that particular website.

The number of links to content management and blogging services like WordPress, Joomla, Blogspot indicates that the Tor hidden services mainly used pre-defined templates to post their content and share their thoughts and expression. Furthermore, there are many links to most of the commonly accessed social networks on the surface web. There were 124 incoming hyperlinks from the dark web pointing to the adult content websites on the surface web.

4.1 Methodology

Revisit to the Graph Terminology

Some basic graph terminology used in this paper is briefly described in this section. Readers familiar with the graph terminology may skip this part.

1) Graph

A graph or an undirected graph G denoted as G = (V, E) is a collection of two sets V and E where V is a finite nonempty set of vertices, also called nodes, and E is a set of edges or arcs or lines that connects the vertices. A directed graph or a digraph is one in which there is a specified direction between vertices as represented by edges. An edge (A, B) represents a link starting from vertex A and ending at vertex B.

2) Path

A path v1 → v2 → v3 → · · · · · · vn is a sequence of edges between two vertices v1 and vn such that vi → vj represents an edge from the vertex vi to vj.

3) Degree

A vertex degree is the total number of edges that are incident on it. For a vertex in the directed graph, the in-degree is the total number of incoming edges to the vertex, and the out-degree is the total number of outgoing edges from the vertex. The degree of a vertex in a directed graph is the sum of its in-degree and out-degree.

4) Connected Components

A connected component S in an undirected graph is a set of vertices such that each vertex in it is reachable from every other vertex in S.

5) Strongly Connected Components

A strongly connected component (SCC) of a directed graph is a set of vertices S such that there is a directed path between any two vertices (x, y) of S. A

weakly connected component (WCC) of a directed graph is a set of vertices S such that there is a path between any two vertices (x, y) of S ignoring the direction.

6) Centrality

A centrality of the vertex is a quantitative measure of the importance of that vertex in the graph. A range of centrality measures exists.

4.1.1 Methods of Data Collection

This section reports the crawling process applied to gather the data, followed by its pre-processing. Then the description of the extraction process of links for graph generation is given.

The data for the study was collected by the custom made web crawler. The crawler was coded in Python, which makes connections to the Tor Hidden Services using SOCKS proxy[15]. The crawler was supplied with the initial onion domains or seeds from the publicly available directory of Tor links[16] from the surface web. It scrapes each of the seeds and stores the new links found in a separate file. The fresh links were again crawled to discover further new links. The process of scraping new links is successively repeated two more times until no new links can be found. The crawler did not scrape the hidden services that require user login or were behind subscription pay-walls. At the end of the crawling session, 48,174 online hidden services were left for further processing.

As per the statistics of the Tor Project Inc., there are more than 100,000 hidden services available online on any given day. However, our crawler could only capture around 48,000 hidden services online. The gap in the numbers of hidden services may be attributed to the chatting platforms on Tor like TorChat[17], where each of the participating users is assigned a unique 16 character .onion address.

Upon completing the process described above, the graph was obtained having 48,174 nodes and 103526 edges. Each node in the graph represents a unique onion domain, and an edge represents a hyperlink between two different domains.

4.1.2 Methods of Data Analysis

The in-degree distribution of nodes (having an in-degree value of less than 50) is presented in Fig. 6. In Fig. 6, it can be seen that ~7% of nodes are source i.e., they have zero in-degree. Followed by this are ~39% of nodes having in-degree 1. On adding the nodes having in-degree up to 10, we get ~97% of total nodes. This result is in line with the previous study[10] about the hidden nature of Tor web pages that are difficult to discover as they have few incoming links. In our analyses, we found that only seven nodes have in-degree higher than 100, of which the maximum in-degree being 184. Fig. 7 shows the in-degree distribution on the log-log scatter plot. There are some spikes present on the plot as the in-degree increases.

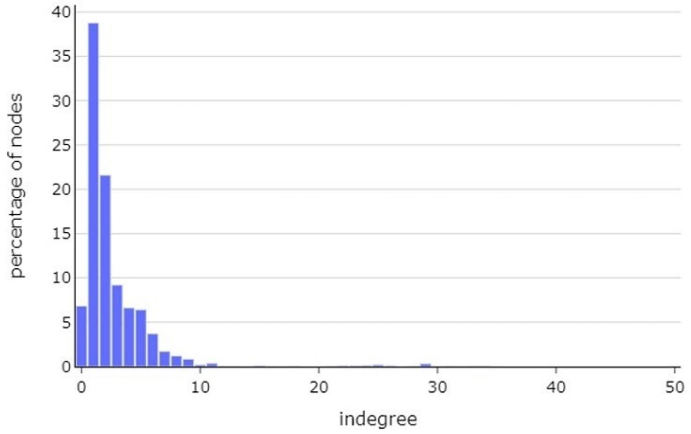

Figure 6: In-degree distribution

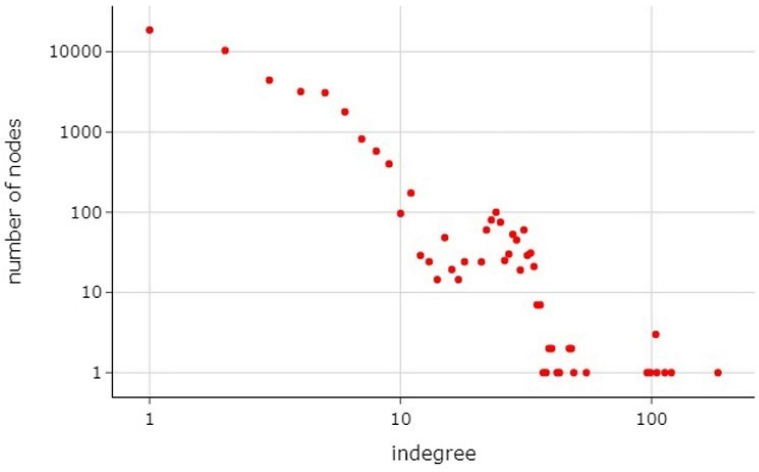

Figure 7: In-degree distribution (log-log scale)

The out-degree distribution of nodes (having an out-degree value of less than 50) is shown in Fig. 8. Around 75% of nodes are sink having zero out-degree and 98% of nodes have out-degree less than or equal to 10. We found that 31 nodes have out-degree greater than 100 and out of them, nine nodes have out-degree greater than 1000. The maximum is 2846 links, thus covering a large portion of the web graph. All of the highest out-degree hubs are the websites offering directory services and Wikis. Although these nodes have a large number of outgoing links, they themselves have less than ten incoming links making them hard to find. The most noticeable feature that came across was the presence of isolated nodes which renders the web graph disconnected. A total of 3006 nodes were disconnected from the entire web graph having no incoming or outgoing links. Fig. 9 shows the out-degree distribution on a log-log scale. The distribution does not seem to be following the power law as the points are scattered haphazardly.

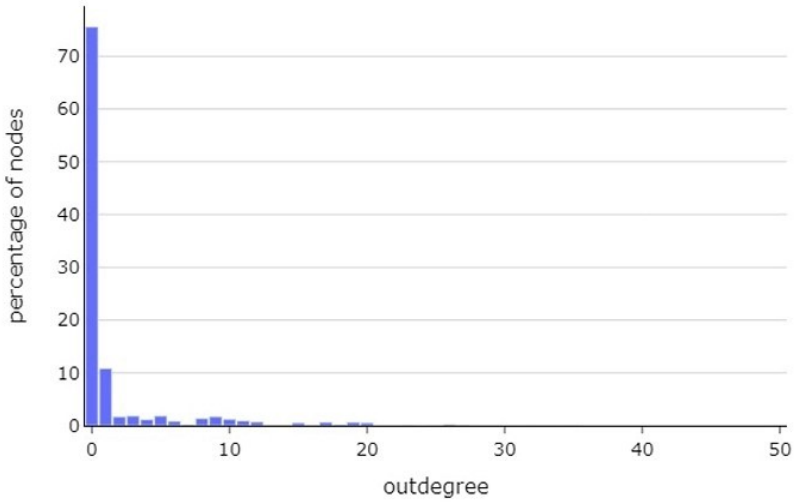

Figure 8: Out-degree distribution

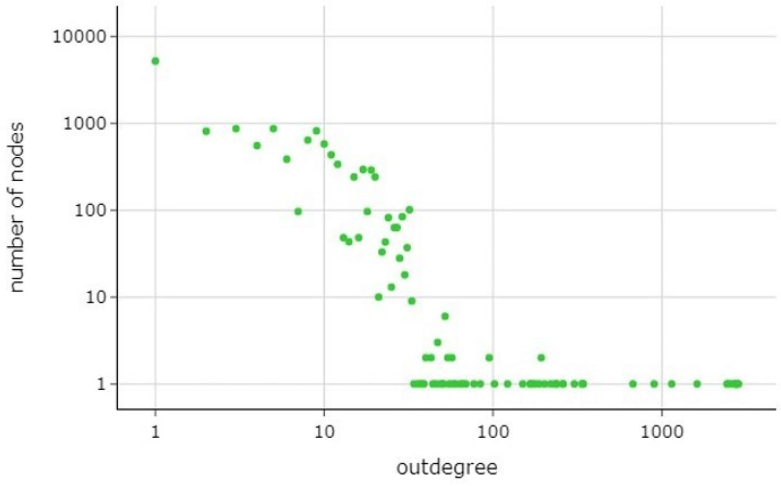

Figure 9: Out-degree distribution (log-log scale)

More than 95% of nodes have their in-degree and out-degree lies between 1 and 10, which makes the web graph a sparse one. Most of the edges are clustered around a few high out-degree hubs.

The largest in-degree and out-degree node were of comparable size in the surface web[4], but the same could not be held in the dark web as evident in the above discussion. The size of the largest out-degree node in the dark web is nearly fifteen times more than the largest in-degree node. Consequently, the in-degree distribution decays slowly than the out-degree distribution.

The low in-degree value of the majority of the nodes suggests that the hidden services did not advertise themselves much. By not revealing the URLs, the hidden services may keep the law enforcement agencies at bay to some extent.

4.2 Software Development Process

- The data for the study was collected by the custom made web crawler.

- The crawler was coded in Python, which makes connections to the Tor Hidden Services using SOCKS proxy. It creates a Transmission Control Protocol (TCP) connection to another server behind the firewall on the client's behalf, then exchanges network packets between the client and the actual server.

- The crawler was supplied with the initial onion domains or seeds from the publicly available directory of Tor links from the surface web.

- It scrapes each of the seeds and stores the new links found in a separate file. The fresh links were again crawled to discover further new links. The process of scraping new links is successively repeated two more times until no new links can be found.

- There is investigation of dark web at the domain level, so all the sub-domains of a particular domain are aggregated under it and are denoted as a single individual node of a graph.

- To be specific, consider a hidden service domain *XYZ.onion* having two sub-domains *en.XYZ.onion* and *ch.XYZ.onion*, both of these sub-domains and their internal web pages are treated as a single node of graph identified by their domain name. Hence any edge from *nodeX* to *nodeY* in the graph represents that there is a hyperlink within the web page of domain *X* pointing to a web page contained in domain *Y*.

- For the construction of the graph, the collected domain web pages are searched for the links in HTML <a> tag to other pages and saved in a separate file with the help of Python code.

- The hyperlinks that point within the web pages of the same domain are discarded. Once the links of all the domains are obtained, an adjacency list for the graph has been generated.

- This adjacency list is supplied to the Python NetworkX library for the generation of the directed web graph. The undirected version of the graph was also created from the directed graph.

- Each node in the graph represents a unique onion domain, and an edge represents a hyperlink between two different domains.

Chapter 5
TECHNICAL DETAILS

Programming Language

PYTHON

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

For this project Python version $\geqslant 3.5$ is recommended. Along with this Python's NetworkX library is used to analyze the Web Graph.

System

LINUX

Linux is a family of open-source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds. Linux is typically packaged in a Linux distribution.

This project is build on Arch Linux which is a Linux distribution. Arch Linux adheres to the KISS principle ("Keep It Simple, Stupid") and is focused on simplicity, modernity, pragmatism, user centrality, and versatility.

Web Browser

TOR BROWSER

The Tor Browser is the flagship product of the Tor Project. It was created as the Tor Browser Bundle by Steven J. Murdoch and announced in January 2008.

The Tor Browser consists of a modified Mozilla Firefox ESR web browser, the TorButton, TorLauncher, NoScript, and HTTPS Everywhere Firefox extensions and the Tor proxy. Users can run the Tor Browser from removable media

SOCKS

SOCKS is an Internet protocol that exchanges network packets between a client and server through a proxy server. SOCKS5 optionally provides authentication so only authorized users may access a server. Practically, a SOCKS server proxies TCP connections to an arbitrary IP address, and provides a means for UDP packets to be forwarded.

SOCKS performs at Layer 5 of the OSI model (the session layer, an intermediate layer between the presentation layer and the transport layer). A SOCKS server accepts incoming client connection on TCP port 1080, as defined in RFC 1928.

Chapter 6
TESTING

Software testing

It is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest.

The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

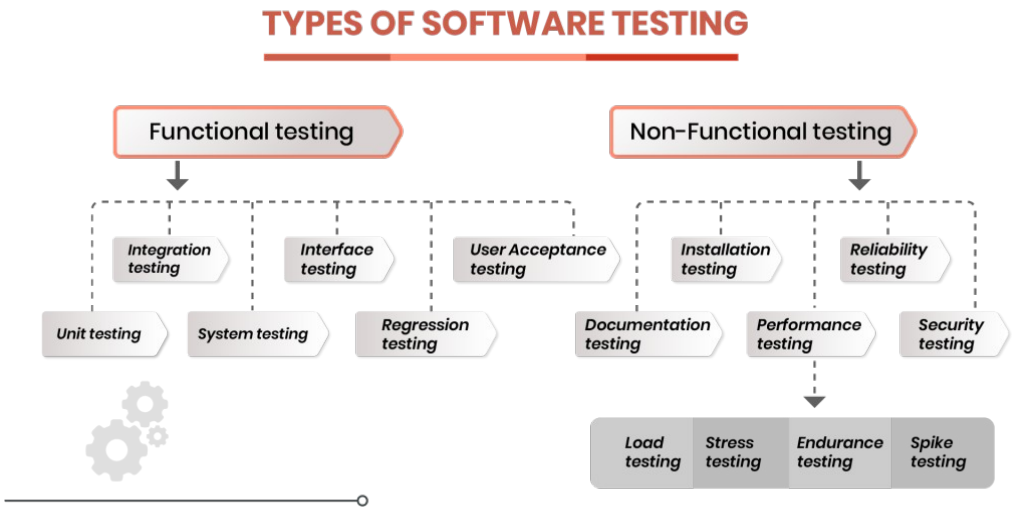

Figure 10: Types of Software Testing

UNIT TESTING

- It is a type of software testing where individual units or components of a software are tested.
- Unit Testing is done during the development (coding phase) of an application by the developers.
- Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

- Unit Testing Tools : Junit, NUnit, JMockit, EMMA, PHPUnit

INTEGRATION TESTING

- It is defined as a type of testing where software modules are integrated logically and tested as a group.
- The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated
- The Integration test procedure:
  1) Prepare the Integration Tests Plan
  2) Design the Test Scenarios, Cases, and Scripts.
  3) Executing the test Cases followed by reporting the defects.
  4) Tracking & re-testing the defects.
- Integration Test Case differs from other test cases in the sense it **focuses mainly on the interfaces & flow of data/information between the modules**. Here priority is to be given for the integrating links rather than the unit functions which are already tested.
- Integration testing is of four types: Top-down, Bottom-up, Sandwich, Big-Bang

SYSTEM TESTING

- It is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications.
- Two Category of Software Testing:
  1) *Black Box Testing* - System test involves the external workings of the software from the user's perspective.
  2) *White Box Testing* - White box testing is the testing of the internal workings or code of a software application.
- Different Types of System Testing:
  1) *Usability Testing* - mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives
  2) *Load Testing* - is necessary to know that a software solution will perform under real-life loads

3) *Regression Testing* - involves testing done to make sure none of the changes made over the course of the development process have caused new bugs. It also makes sure no old bugs appear from the addition of new software modules over time.
4) *Recovery testing* - is done to demonstrate a software solution is reliable, trustworthy and can successfully recoup from possible crashes.
5) *Migration testing*- is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.
6) *Functional Testing* - Also known as functional completeness testing, Functional Testing involves trying to think of any possible missing functions. Testers might make a list of additional functionalities that a product could have to improve it during functional testing.
7) *Hardware/Software Testing* - IBM refers to Hardware/Software testing as "HW/SW Testing". This is when the tester focuses his/her attention on the interactions between the hardware and software during system testing

USER ACCEPTANCE TESTING

- It is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment.
- Performed by – Client and End user.
- Need of User Acceptance Testing arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.
- How to do UAT Testing:
  1) Analysis of Business Requirements
  2) Creation of UAT test plan
  3) Identify Test Scenarios
  4) Create UAT Test Cases
  5) Preparation of Test Data(Production like Data)
  6) Run the Test cases
  7) Record the Results
  8) Confirm business objectives

6.1 Objectives Of Testing

Figure 11: Objectives Of Testing

- To evaluate the work products such as requirements, design, user stories, and code
- To verify the fulfillment of all specified requirements
- To validate if the test object is complete and works as per the expectation of the users and the stakeholders
- To build confidence in the quality level of the test object
- To prevent defects in the software product
- To find defects in the software product
- To provide sufficient information to stakeholders to allow them to make informed decisions, especially regarding the level of quality of the test object
- To reduce the level of risk of insufficient software quality
- To comply with contractual, legal, or regulatory requirements or standards, and to verify the test object’s compliance with such requirements or standards

6.2 Test Plans

Furthermore testing include system compatibility as a whole i.e. making the project system independent. Since Programming language is out-of-the-box system independent we can conclude that working of this project is guaranteed on almost all Operating Systems like Windows, Linux or OSX.

Data mining mostly doesn't include user interface model because what matters the most is the resultant of the process. Since we will conclude our work by analyzing the Topological Structure of the Web, end user will be hardly interacting with the coding part.

To improve accuracy of the model time is necessary as the crawler need to scrape every out-degree node of a current node and this flow of the program will follow a Breadth First Search Algorithm. Scraping thousands of links for external pointing hyperlinks will be time consuming. Moreover user will be anonymous due to the layered encryption system of the Tor Network, so fetching source data itself is time consuming.

Chapter 7
FUTURE WORK

A deeper study of the problem with a relatively large sample size is needed to obtain the properties of the Tor web graph with greater details.

There is a need to expand this study to cover other dark web networks which shall bring out a more clear picture of these encrypted networks.

The introvert nature of the hidden services helps them to sustain a strongly connected component of a similar type. The average distance between any connected pair is also low.

The law enforcement organization may take advantage of this nature to shut down all the services simultaneously.

Future planning also include implementation of encrypted proxy connection to the Tor Server for faster accessing of source data.

Chapter 8
CONCLUSION

In this study, we have performed an analysis of the Tor web graph at the domain level. The data has been obtained by running the web crawler tailored in Python to specifically cover the Tor dark web. The graph constructed from the collected data consists of 48,174 nodes and 1,03,526 edges where a node represents a Tor hidden service and the hyperlink between two hidden services is represented by an edge. The in-degree distribution of the web graph seems to follow the power-law distribution as the log-log plot of in-degree distribution do resemble roughly a straight line. Out-degree distribution does not follow a power law.

The study further discovers the links to the regular Internet from the dark web. A large number of such links were found, which were connecting to many of the popular surface websites like Facebook, Google, Amazon, etc. The majority of the links were to social networks, web content management, news and adult content. The TLD .com and .org were predominantly present while .de and .ru were among the top in country wise TLD.

Chapter 9
REFERENCES

[0] A. Alharbi et al., "Exploring the Topological Properties of the Tor Dark Web", IEEE Access, vol. 9, pp. 21746-21758, 2021.

[1] J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, The Web as a graph: Measurements, models, and methods, in Computing and Combinatorics (Lecture Notes in Computer Science), T. Asano, H. Imai, D. T. Lee, S. Nakano, and T. Tokuyama, Eds. Berlin, Germany: Springer, vol. 1627, 1999, pp. 1#17.

[2] A. Broder, R. Kumar, F. Maghoul, and P. Raghavan, Graph structure in the Web, Comput. Netw., vol. 33, nos. 1#6, pp. 309#320, Jun. 2000.

[3] O. Lehmberg, R. Meusel, and C. Bizer, Graph structure in the Web: Aggregated by pay-level domain, in Proc. ACM Conf. Web Sci., Bloomington, IN, USA, 2014, pp. 119#128.

[4] R. Meusel, The graph structure in the Web#analyzed on different aggregation levels, J. Web Sci., vol. 1, no. 1, pp. 33#47, Aug. 2015.

[5] M. W. Al Nabki, E. Fidalgo, E. Alegre, and I. de Paz, Classifying illegal activities on tor network based on Web textual contents, in Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics, 2017, pp. 35#43.

[6] E. Jardine, The darkWeb dilemma: Tor, anonymity and online policing, SSRN Electron. J., vol. 21, pp. 1#24, Dec. 2015. [Online].

[7] G. Owenson, S. Cortes, and A. Lewman, The Darknet's smaller than we thought: The life cycle of Tor hidden services, Digit. Invest., vol. 27, pp. 17#22, 2018, doi: 10.1016/j.diin 2018.09.005.

[8] E. Jardine, Tor, what is it good for? Political repression and the use of online anonymity-granting technologies, New Media Soc., vol. 20, no. 2, pp. 435#452, Feb. 2018, doi: 10.1177/1461444816639976.

[9] M. Chertoff and T. Simon, The impact of the dark Web on Internet governance and cyber security, Global Commission Inter net Governance, vol. 6, pp. 1#18, May 2015. [Online].

[10] M. Bernaschi, A. Celestini, S. Guarino, and F. Lombardi, Exploring and analyzing the tor hidden services graph, ACM Trans. Web, vol. 11, no. 4, pp. 1#26, Sep. 2017.

[11] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, Trawling the Web for emerging cyber-communities, Comput. Netw., vol. 31, nos. 11#16, pp. 1481#1493, May 1999.

[12] A.-L. Barabási, R. Albert, and H. Jeong, Scale-free characteristics of random networks: The topology of the world-wide Web, Phys. A, Stat. Mech. Appl., vol. 281, nos. 1#4, pp. 69#77, Jun. 2000.

[13] D. Donato, L. Laura, S. Leonardi, and S. Millozzi, The Web as a graph: How far we are, ACM Trans. Internet Technol., vol. 7, no. 1, pp. 1#23, 2007.

[14] M. Serrano, A. Maguitman, M. Boguñá, S. Fortunato, and A. Vespignani, Decoding the structure of the WWW: A comparative analysis of Web crawls, ACM Trans. Web, vol. 1, no. 2, pp. 1#25, 2007.

[15] (2021). SOCKS. [Online]. Available: https://en.wikipedia.org/wiki/SOCKS/

[16] (2021). The Hidden Wiki. [Online]. Available: https://thehiddenwiki.org/

[17] (2021). TorChat. [Online]. Available: https://en.wikipedia.org/wiki/TorChat/

[18] E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, Web data extraction, applications and techniques: A survey, Knowl.-Based Syst., vol. 70, pp. 301#323, Nov. 2014.

[19] B. Potvin and R. Villemaire, Robust Web data extraction based on unsupervised visual validation, in Intelligent Information and Database Systems (Lecture Notes in Computer Science), vol. 11431, N. Nguyen, F. Gaol, T. Hong, and B. Trawiski, Eds. Cham, Switzerland: Springer, 2019, pp. 77#89.

[20] (2021). Dark Web. [Online]. Available: https://en.wikipedia.org/wiki/Dark_web