D-miner: A Framework for Mining, Searching, Visualizing, and Alerting on Darknet Events

Heather Lawrence, Andrew Hughes, Robert Tonic, Cliff Zou
College of Engineering and Computer Science
University of Central Florida
Orlando, Florida 32826

***Abstract*—Darknet resources are mined for their data in order to provide possible cyber threat intelligence to network operators. Network operators, however, often have limited resources with which to search the darknet for threats. Previous work in this area has failed to address this use case instead focusing on sales volumes, vendor characteristics, and identifying the sale of zero day exploits. In this paper we present D-miner: a modular framework designed to mine data from websites, specializing in darknet sites, and parse the data into JSON objects for searching, visualizations, and alerts. This open source solution to darknet mining is intended to make it easier for network shareholders to monitor the darknet for potential threats. We show how D-miner is customizable for multiple use cases and how it can be used to visualize data to aid analysis.**

I. Introduction

Web forums and marketplaces are avenues where computer security enthusiasts can share or purchase ideas, vulnerabilities, or code. While increased sharing has the potential to strengthen our defenses, not all collaboration on the darknet or deepnet is benevolent. Like Nunes, et al. [1] we define 'darknet' as resources that cannot be accessed without Tor, 'deepnet' as resources that are not indexed by search engines, and 'clearnet' as resources that are indexed by search engines. For example, the Mirai botnet code was released on HackForums, a deepnet forum, in late September 2016 right before massive DDoS attacks to DYN and Deutsche Telekom in October and November of the same year, respectively [2]. Like many malware variants, Mirai did not take advantage of zero-day attacks instead leveraging default credentials in order to takeover Internet of Things (IoT) devices. This is a common practice as several variants of malware take advantage of old vulnerabilities that have not yet been patched [3].

Like Mirai, malware is available through the darknet - either through renting established botnets, purchasing malware source code, or purchasing guides to "roll your own" malware as shown in Figure 1. The listing shown in Figure 1 is an example of a vendor leasing time on a botnet consisting of 25,000 zombies - essentially providing DDoS-as-a-service to potential buyers. Increased availability through the darknet allows others to tinker with established malware variants, increasing the likelihood of an attack leveraging a similar vector. Verizon's 2016 Data Breach Investigation Report [4] indicates that "99% of malware hashes are seen for only 58 seconds or less. In fact, most malware was seen only once." This indicates that malware is custom-made to fit a victim's attack vector and as completely new strains of malware take more resources to develop it is more likely that attackers are reusing and obfuscating available code. Darknet listings are one of the avenues that can indicate an increase of a particular malware variant due to its increased availability.

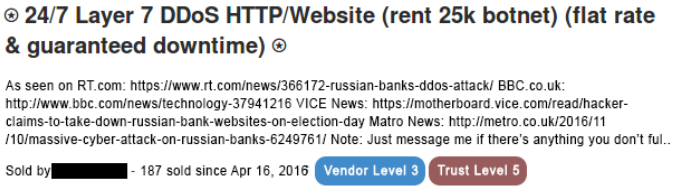

Fig. 1: Example of botnet rental listing

As a network's attack surface increases with the popularity of the IoT and bring-your-own-device (BYOD) policies, network operators are motivated to harden infrastructure through a security operation centers (SOC) using human analysts to monitor positive indications that an attack has occurred or is occurring [5]. Data that can be used to broaden a network operator's visibility in terms of threats is gathered in-network through internal log data and out-of-network through threat intelligence streams. Threat intelligence with respect to computer security is a developing field, but most agree that it is a collection of cyber threat information from various sources intended to provide context vital to understanding attacks, predicting attacks, and providing data for attribution for law enforcement [1], [6], [7]. Several public areas on the Internet are already mined for external network data including open source intelligence sources like Pastebins, honeypots, Internet relay chat (IRC), peer-to-peer (P2P) networks, breach databases, zone registrars, regional Internet registry data, and social media networks such as Twitter and Facebook.

Darknet monitoring is a useful addition to sources already in use. Darknet marketplace listings include hacking-for-hire services, general exploits or exploit kits for sale (often for use against specific service sectors), hacked accounts, counterfeit electronics, spam/phishing/malvertising campaigns, increasing social media followings through the use of fake accounts, and botnet rentals. There are several public projects available for scanning the darknet, including the OnionScan project [8] and the Shadowserver Foundation [9], that provide information about hidden services, honeypot data, and botnet data.

978-1-5386-0683-4/17/$31.00

A. Contributions

In this vein we present D-miner, an open source solution designed for darknet scraping and parsing while providing native support for full text searching, visualization, and alerting for network shareholders. Our approach provides a solution that is configurable for multiple use cases, publicly available, and it is extensible for other data source aggregators. We tested D-miner by scraping over 5,000 pages of listings and parsing them into HTML scrapes resulting in over 500,000 JSON objects, each representing an individual listing available over time that can be visualized and tracked.

In short, our contributions are as follows:

1) Automatable scraper for darkweb that can also be applied to clearnet assets
2) Near-real time scraping frequency utilizing scraping obfuscation techniques
3) Extendable and adaptable framework that allows users to change datastores, parse only, scrape only, or completely pipeline data ingestion
4) Quickly and easily deployable system requiring minimal configuration on behalf of the user
5) Source code that is publicly available under an MIT License

The remainder of the paper continues as follows: Section II covers related work, Section III discusses implementation, Section IV explores a case study, Section V describes limitations and future work, Section VI concludes this paper and indicates where the data and code base can be found.

II. Related work

Scraping the darknet for analysis is not new, but datasets quickly age due to the mercurial nature of darknet sites. This is particularly true for darknet marketplaces (DNMs). The largest public collection available covered 89 DNMs and 37 related forums from 2011-2015 [10]. Only 6 of the 89 DNMs still remain accessible. Several DNMs are no longer available due to a variety of reasons including law enforcement, voluntary shutdowns, and exit scams.[1] As case in point to the mercurial nature of darknet assets, Alphabay was shut down by law enforcement during the course of our data collection [11].

Analysis of the largest collection of DNMs [10] was provided by Soska and Christin [12] who identified sales volumes and vendor characteristics in the largest DNMs at the time. However, not all of the DNM data was parsed and analyzed including Alphabay which was considered one of the leading DNMs available [13] until recently. Soska and Christin provided an overall perspective of a DNM as a whole as they analyzed revenues from drug listings and paraphernalia, discussed cash revenues of marketplaces, and their impact (or lack thereof) on the overall globalization of drug trafficking. They failed, however, to analyze other important listings and services included in marketplaces as applied to cybercrime. Most importantly the method used to gather the archive [10] was grossly inefficient and saved files that provided no value and led to the subsequent banning of the account. For example, on the last day of scraping Alphabay on the 5th of July 2015, the archive contains 413 valid vendor account details after which Alphabay flood warnings are saved in lieu of actual data: "Flood limit, wait 10 seconds. This is strike 1." This method continued to enumerate potential vendor accounts that did not exist and saved the error messages to the archive. On that particular day, only 413 accounts existed despite 70,000 enumerations and out of the 71 MB required to store that data, only 11.2 MB was usable.

OnionScan [8] analyzes hidden services and provides a means to scan for privacy and security problems. It also provides a feature to tag search results that can aid users with correlation. The Shadowserver Foundation [9] scans internet services and provides network statistics on ASNs, botnets, malware, and viruses. Both of these projects provide valuable information on the health of the internet and hidden services, but provide no insight on data hosted within hidden services. Scanning traffic and services provide a means to monitor attacks as they are occurring but force defenders to take a reactive approach. Knowledge of the frequency of purchase of different malware families provides a forward indication of what attack vectors are growing in popularity.

Nunes et al. [1] made a large contribution in this area by creating a similar scraper/parser system that analyzed deepnet and darknet forums and marketplace listings. Their machine learning algorithm was able to correctly identify zero day exploits for sale. Unfortunately their work was described at a high level in that they did not discuss how they overcame challenges to gather their data, did not discuss analysis of other listings involved with cybercrime save zero day exploits, and their scraper was not made public as they are transitioning their system to a commercial partner. Additionally their work did not mention any metrics of their scraping process, how they defeated captchas and DDoS protections, or if their project supported extendability beyond detecting new zero day exploits. Lastly their dataset is gathered over a period of 4 weeks while our data was gathered over a period of six months.

As cyber threat intelligence (TI) is a developing area, there have been growing pains associated with choosing the correct intelligence such that it provides actionable data. Sillaber et al. [5] surveyed analysts from prominent security operation centers to determine how threat intelligence was used and if it was useful. They released several findings noting that users do not find information from disparate service sectors useful, that TI tools often limit data accessibility (i.e., filters) to actionable data, and that automated integration of external resources can improve data quality over manual entry. They also included recommendations to threat intelligence providers such as educating users about the data quality of the intelligence they are ingesting and focusing on current threat intelligence while still making stale data available for historical analysis.

These gaps were addressed by creating a framework which is modular and extensible such that, when sites no longer operate or new sites are created, it can be easily adapted

[1] An 'exit scam' is a colloquial term to describe when a DNM operator purloins all Bitcoin in escrow from vendors and users of that DNM.

to new assets or assets that have changed. Our analysis focuses specifically on cybercrime in general vice a single, but important, facet (i.e., zero day exploits). Lastly, we ensured that findings and recommendations provided by Sillaber et al. [5] were adhered to in order to provide effective and practical data.

III. Design methodology

Darknet crawling is made notoriously complicated via usage of captcha and DDoS protections, DNM availability, DNM banning, and handling illegal material. Previous research in this area did not provide enough accessible information with which to build an automatable scraper for DNMs, did not provide adequate analysis of listings with regards to general cybercrime, and did not provide an extensible or native means to visualize or alert on data. We assumed limited resources of analysts with regard to the installation of this framework and designed the framework to meet the needs of various use cases as described further in Section IV.

D-miner was designed with several use cases in mind. It was designed to be useful to analysts as mentioned above as well as entities interested in darknet data including researchers studying the darkweb domain, law enforcement officials, and antivirus/antimalware organizations as the framework provides metrics of interest customizable to meet the needs to the aforementioned users. D-miner is constructed such that researchers studying other facets of darkweb data (e.g., drug market globalization, weapons purchases) can also use the framework, not just those studying the effects of the darknet on computer security. For example, law enforcement can use the framework to obtain additional points of attribution. Finally, malware statistics from the darknet can be used by antivirus/antimalware organizations.

In order to minimize pain points felt by current users of threat intelligence, we kept the findings and recommendations made by Sillaber et al. [5] in mind. More specifically, as users of threat intelligence did not find data from other sectors useful, we needed a way to filter hits based on areas of interest. We solve this technical challenge by providing native support for Elasticsearch [14] to provide full text searching. In their research under finding 3, Sillaber et al. discussed a shortcoming with current TI sharing tools limiting data accessibility. With D-miner data is fully accessible (it can be grouped as needed by the user) and the framework is modular to allow changes as needed to customize installations and extend functionality. In addition, they found that TI analysts must trust the data they are using to make informed decisions. Unfortunately, data scraped from darknet spaces is temporal in nature. That is, data is hard to verify as DNMs are in constant flux, so while we cannot verify these transactions took place we have taken the best effort approach, via programatically and manual verification, to verify that the data scraped was effectively parsed into accurate JSON objects for analysis. This process is detailed more in the *Parsing data* subsection that follows.

Source	Dates Acquired	Sections Gathered
Alphabay	12/5/16-7/4/17	Counterfeit Electronics, Botnets & Malware, Exploits, Exploit Kits, Security Software, Other1, Other2
Dream Market	12/5/16-7/20/17	Counterfeit Electronics, Hacking Services, Software, Hacking, Data
Hansa	2/3/17-current	Electronics, Leaks & Databases, Malware & Botnets, Hacking Services, Hacking Guides, Security Guides
Valhalla	2/3/17-current	Digital Guides, Hacking Services

TABLE I: Darknet Marketplace Data Sources

Death by Captcha (DBC) allows the framework to cheaply solve captcha mechanisms through human labor. Programmatic approaches to captcha-solving are out of the scope of this research and an open research problem as many current implementations are largely ineffective and slow, increasing the chance of detection, the load on the DNM, and the length of time needed to scrape. In future work we would like to integrate a new module to solve this programmatically.

The user can control D-miner through the command line interface via terminal. Arguments such as which scraper or parser to use, DNM and Death By Captcha (DBC) [15] credentials, and where to save data to disk are parsed in order to prepare the data pipeline of the framework. If a scraper is chosen, D-miner will launch Firefox if there is a login with a captcha to bypass. Otherwise headless scraping will occur. The scraper either visits each page listed in a configuration file or scrapes URLs dynamically by section and optionally saves the raw HTML scrapes to a directory chosen by the user. If a parser is chosen, data saved to disk is converted to JSON objects for ingestion a database with an implemented datastore interface. A Kibana [16] dashboard then provides a visual representation of the objects indexed. As it is fully modular, the framework can be used to perform standalone tasks and be extended or used by third parties. That is, the scraper and parsers can be used by themselves or in conjunction with other features of the daemon as well as interfaced by other projects. A visual representation of the flowchart of the framework is shown in Figure 2.

A. Data sources

Gathering data was required to test the functionality and robustness of the framework. Listings available on darknet marketplace are of particular interest as they detail items or services for sale. The dataset used for this project contained 4 DNMs as shown in Table I. We reviewed adding the Gwern [10] dataset to our analysis, but as actionable threat intelligence is temporal in nature, the data archived by Gwern was reviewed and mostly discarded due to the age of the data. The Gwern archive ranges in dates, but is mostly 2 years old or older. While this framework does not yet encompass all DNMs and all forums, D-miner is expandable to additional sources, both clearnet and darknet, and this is discussed further in the following section, *Scraping marketplaces*.

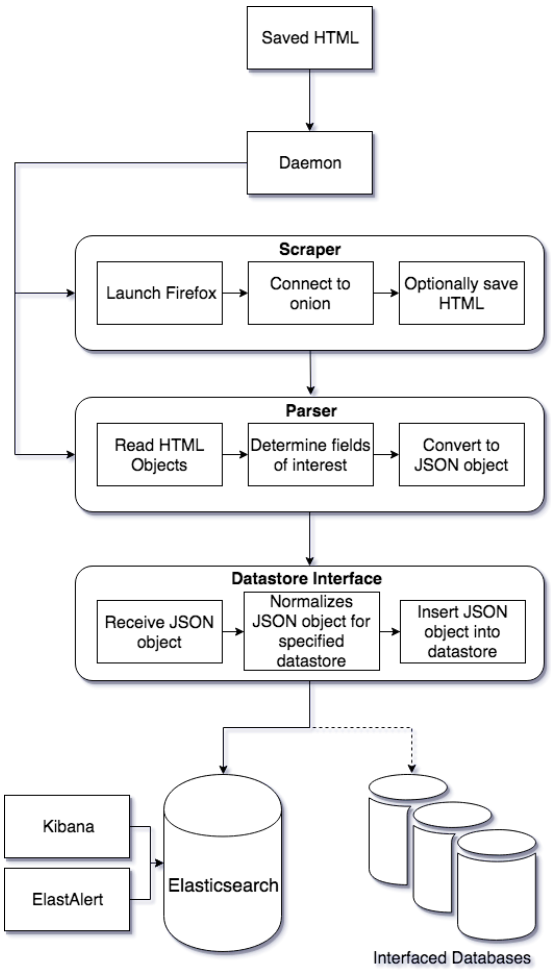

Fig. 2: Flowchart for D-miner

B. Scraping marketplaces

Scraping on a regular basis was subject to DNM availability. That is, if a DNM was undergoing DDoS, maintenance, or the Tor exit node was very slow, it was much slower to load and save pages than usual or it was outright unavailable. If a DNM was unavailable scraping occurred at the next opportunity.

Selenium WebDriver [17] controls Firefox through the use of geckodriver [18] and facilitates the completion of login and DDoS prevention prompts. Once the DDoS prevention and login prompts are bypassed, the session in Selenium is cloned into a Python Requests session so that the scraping can be done in a headless browser. This decreased the amount of time needed to scrape each page from 2-3 seconds per page down to 0.5-1.5 seconds per page. Slow scraping occurs because extra requests are made through Selenium to retrieve images, Javascript, and Cascading Style Sheets (CSS).

In the event that the CSS are randomized, the use of Selenium allows D-miner to find required CSS fields to enter text for login. Dream Market was one such example as the CSS was randomized on the login page. This allows D-miner to bypass attempts to block automated access that would be otherwise be difficult to implement and hard to adapt to new markets using libraries such as urllib or Requests.

Captchas are a common problem with web scraping and the darknet is no exception. DDoS attacks have been used in the past as a means of eliminating competitors [19]. Selenium [17] is able to take screenshots of assets for captcha-solving and preserving a site for posterity. Screenshots make captcha-solving feasible and easier to implement. We used Death by Captcha (DBC) [15] to automate the captcha-solving process. DBC is a captcha-solve service that employs personnel to solve captchas much like Amazon's Mechanical Turk [20]. This method adds negligible cost to fully automate the framework at $1.39 for 1000 successfully decoded captchas at the time of writing.

Captcha-solving is a significant road block to scraping in near-real time and automation. Sources make use of different captcha styles requiring a different captcha-solving approach for each individual source. Dream Market, for example, displays seven to eight alphanumeric digits and a box outlines four of the digits required to solve. When sent to DBC in its entirety the error rate of solving increased significantly and required more attempts to solve successfully. The error rate decreased after using the OpenCV [21] library to find the contours of the box outline and cropping to the edges of the box before sending the captcha to DBC. Alphabay, in contrast, did not use a box to indicate which parts of the image were required.

It is important to maintain anonymity while browsing marketplaces as it is possible for DNM operators to detect browser settings unique to individuals in order to fingerprint users [22]–[24]. The Tor Browser is a standard web browser maintained by the Tor Foundation [25] for the purpose of browsing onion websites available over the Tor network. The Tor Browser attempts to mitigate fingerprinting by setting a static resolution of the browser and adjusting multiple settings upon starting. Similarly, D-miner uses Selenium [17] to programmatically adjust its fingerprint to appear as a normal Firefox browser. D-miner then authenticates to the darknet site and hands the session to the Requests library thus utilizing the headers and authentication data from Selenium. Therefore, even after scraping is handed off from Selenium to Requests, D-miner has the same fingerprint as Firefox.

Additionally, the consistency and frequency of HTML requests can be an indicator of a bot scraping a page. This can set off alerts to operators and be a drain on web hosting resources if the pages are requested in a rapid succession. D-miner uses random wait times to access and manipulate elements on a page providing another layer of obfuscation and decreasing the chance of DNM detection.

D-miner has reduced the time it takes to implement scrapers for new sources by providing and abstracting most of the features required to scrape a source. Captcha solving, login detection, dynamic URL gathering, means to avoid detection and subsequent banning, and saving the data are all functions that users do not have to implement. However this process is not completely automatable as every asset implements their source code differently. Login and captcha fields are not given

the same IDs. Not every source requires login to scrape (e.g. Hansa and Valhalla) and not every source implements DDoS protection. These must be determined manually from the HTML of each new source. Additionally, threat intelligence data must be useful and not every darknet hidden service can be scraped for useful threat intelligence. Automatically locating and scraping these sources adds to network background noise and wastes space to save useless data.

C. Ethical data scraping

Much of the data available on the darknet is of interest to law enforcement due to illicit content. The possession of this illicit content often results in legal ramifications. It was thus important to minimize risk to researchers interested in investigating this domain. We minimize this risk programmatically by pipelining raw HTML scrapes, without images, to Elasticsearch. To the best of our knowledge this has minimized the possibility of retaining illicit content.

Data was scraped from sections of darknet marketplaces that claimed association with hacking or computer crimes. That is, sections involving drug trafficking, weapons, and pornographic material were not included in the scraping process. This minimizes the amount of data that we need to scrape from the DNM thus reducing usage of Tor and the DNM. Scrapes are provided for public use both to further reduce usage of the Tor network for scraping and provide data for reproducibility and posterity.

In order to protect operators, vendors, and buyers, darknet scrapes were pulled from 'public areas' - either pulled from public archives or DNMs with thousands of users where vendors operate under the assumption that the contents of their listings are under review by law enforcement [12], [26]. A best effort approach was used to ensure data was free of personal details, like phone numbers, that could be used to directly identify market operators or participants.

Lastly, the use of Death by Captcha (DBC) is ethically questionable due to its association with scammers as it provides a means to reliably defeat captcha services to generate fake accounts en masse [27]. We further address this limitation in Section V.

D. Parsing data

Parsing is a process that transforms the various raw data sources into JSON objects. Beautiful Soup [28] is a Python library for parsing data encapsulated by HTML and XML tags. It is employed to easily parse the HTML scrapes. While Selenium WebDriver [17] also has the ability to natively filter HTML through the use of xpaths,[2] Beautiful Soup allows us to select elements more intuitively by enabling us to search for Cascading Style Sheet (CSS) selectors. Pseudocode for algorithm is shown in Algorithm 1. Beautiful Soup increases the modularity of the scraper by allowing us to rapidly prototype parsers for new assets using similar code to parse elements.

[2]xpath is a query language for selecting nodes from a XML document

Algorithm 1 Parsing

```
procedure PARSING(Darknet marketplace scrape)
    Open HTML document
    Convert to BeautifulSoup4 object or
    Convert passed HTML obj to BS4 obj
    while there is HTML data do:
        Iterate through listing blocks via CSS attributes
        Store listing in datastore via interface
    end while
end procedure
```

Source	Fields Extracted
Alphabay	Listing name, Listing ID, Listing date, Price in USD, price in BTC, Category, Vendor name, Vendor ID, Views, Bids, Quantity, Timestamp
Dream Market	Listing name, Price in BTC, Category, Vendor name, Vendor successful transactions, Vendor rating, Timestamp
Hansa	Listing name, Price in USD, Price in BTC, Category, Vendor name, Vendor rating, Views, Timestamp
Valhalla	Listing name, Price in EUR, Vendor name, Vendor feedback, Timestamp

TABLE II: Fields Extracted

It is important to parse fields that encapsulate each listing from each DNM properly. For example, the code of an encoded image was not considered data that could be used to plot DNM development over time or provide a point of attribution. Similar to Soska and Christin [12], care was given to identify fields that related to vendors. That is, fields that can help correlate vendors across aliases like join date, stylometry, types of products or services sold, shipping policies, and user reviews. As mentioned in future work, these fields could help investigators determine possible points of attribution for vendors across marketplaces. A table of the fields parsed into JSON objects is described in Table II.

We took care to track areas of interest across a wide range of sources. Several DNMs do not categorize data in the same way, i.e., a section titled the same may have different listings and vice versa or may not exist. For example, in Alphabay malware listings categorized under 'Malware & Botnets' whilst in Dream Market malware listings mostly fall under 'Security Software.' Thus, while the section where the listing was found was recorded for posterity, the contents of the listing was considered more important.

In order to optimize our ability to visualize data, we converted the values pulled from the HTML file scrapes into a JSON file structure. Where possible, fields were converted to represent their native types (string, integer, double, etc). These operations allow us to optimize our dataset to extend our query abilities, such as performing numeric aggregations and text-field searches, and have more precise visualizations due to less layers of datatype casting. Storing data as JSON objects makes the dataset more human-readable and it is trivial to export data into other storage backends that support JSON documents or structures.

E. Validating data completeness

In order to reduce data quality problems related with traditional datasets [5], manual inspections of our data against the scraped data is performed to ensure the integrity of our data with respect to original data available on the DNM (i.e., each scrape was checked against the actual listing). This was important in the event a DNM was unavailable for part of the scraping process which could result in data loss. In addition, pages were checked against actual links as some DNMs will serve the last page available in a section if the page requested does not exist. This results in storing duplicate listings. To mitigate this we compared resulting objects for similarities (e.g., same amount of data, and duplicate fields) and established a threshold for duplicate content.

F. Potential uses of platform

Once the data is parsed into JSON objects, the data can be imported into a supported database, like Elasticsearch. JSON objects are preferable as they are easy to export for use in other platforms. Elasticsearch is supported natively as we found other databases (e.g., SQL databases and columnar databases) to either have poor JSON support, slow exporting, or restrictive APIs causing the framework to perform at a diminished level as the amount of data indexed increases. Kibana compliments Elasticsearch to allow for the visualization of full text searches. They are both open source, have no cost, and are well documented further decreasing possible overhead of network shareholders.

A visual representation of data makes it easier to understand as the data set grows. Visual representations, however, have to be uncluttered and relevant in order to be effective. Visualizations provided were designed to provide a quick reference to keywords as defined by the user. This aided us in answering questions we had about the data set. For example, what types of malware were sold on a particular market and how much of a market share did they hold? Similarly, which vendors dominate each DNM in terms of selling botnet services? These findings are discussed in detail in the following section.

The framework was designed to meet the needs of researchers, analysts, and law enforcement in gathering dark web data and curating the raw data into actionable data. Interdisciplinary researchers interested in studying the dark web are at a disadvantage when it comes to obtaining a recent dataset large enough to study. In order to open this area to researchers we designed the framework to be adaptable to different fields of interest. Instead of studying cybercrime, for example, researchers can apply D-miner to study the darkweb for weapons trafficking.

We considered the needs of law enforcement to use darkweb data as additional points of attribution for cybercriminal activity. As D-miner natively supports tools to search and visualize data, it is designed to reduce the amount of time an agent needs to comb through data in order to find correlations they are looking for.

IV. Case Study

We use this section to investigate areas that are of interest to different network shareholders like researchers, analysts, and law enforcement. We surmised that researchers would be interested in a general view of DNMs before pursuing inquiries about their domain. Likewise analysts would be most interested in listings mentioning their organizational name (e.g., an analyst for Netflix would be interested in listings mentioning related keywords) and listings selling prominent malware families. Finally, law enforcement would be interested in drilling into details about specific vendors and the goods they sell.

A. General data

We initially investigated unique listings across Alphabay and Dream Market. The data gathered from Dream Market includes 322,812 unique listings compared to Alphabay's 21,355 unique listings. We mapped Alphabay's unique listings over time and found individual sellers were causing spikes in unique listings on February 1, 2016 and September 26, 2016. After reviewing darknet news articles published around that time, we surmised that these particular users were migrating their goods over to Alphabay as there was no mention of exit scams or other similar market environmental influences such as a DNM takedown. Spikes in unique listings were seen again once Alphabay was subject to law enforcement takedown [11] on July 4th 2017 as Dream Market listings spiked in the sections gathered from an average of 4,400 listings to 11,934 listings in the sections sampled.

B. Malware detection & brand protection

Several types of malware keywords such as 'rat' and 'botnet' were searched to determine the top listings as determined by interest from buyers. Notice that several prominent malware families in Table IV are frequently visited by users like the Bleeding Life exploit kit at 13,000 views, BlackShades RAT at 54,000 views, and BTC Stealer at almost 50,000 views.

Organizations can be proactive about threats by searching for and monitoring keywords related to their brand. Looking for spikes in views can also indicate a serious compromise or vulnerability in services. Brand monitoring can also help combat fraudulent activity such as carding campaigns and large scale unauthorized account use. For example, after searching for 'Netflix' we determined the most popular listings that pertain to Netflix deal with account fraud as shown in Table III.

C. Vendor listings

We wanted to study an event that might be of use to law enforcement. When looking for anomalies in new listings we found a spike occurring on January 2, 2017. We then found the source of the anomaly to be a vendor named "theshadowbrokers". This vendor had listed 11 new items on the second of January, and 15 additional items the next day. Exploits were

Listing Name	Listing Views
[DDI] Social-Crack-Pack January 2017 Updated Programs for HACKING into Facebook, Instagram, Netflix, Twitter Hack ANY kind of social account	4,906
Best Account Generator Netflix & Minecraft	1,117
[FREE]Netflix Account Hacker	700
SCAM PAGES Phishing NETFLIX-HQ Phishing Pages && Letter ++ Netflix Checker	514

TABLE III: Netflix-specific brand hits

Search Term	Listing Name	Listing views
Rat	BlackShades RAT 5.5.1 + User Guide	54,125
Malware	BTC Stealer 4.3 and Mass Address Generator 1.2+How to steal bitcoin user guide	49,842
Ransomware	Stampado 2.2 - ONLY $39 - FULL LIFETIME LICENSE	22,522
Botnet	24/7 Layer 7 DDoS HTTP/Website (rent 25k botnet) (flat rate & guaranteed downtime)	26,126
Exploit Kit	[Exploit Kit] Bleeding Life (2.0)	13,675
Keylogger	Spytech SpyAgent Keylogger	4,108

TABLE IV: Most visited listings per keyword search

listed for sale at $7,500 USD per exploit and implants were listed for ten times that amount, or $75,000 USD each. The most viewed item theshadowbrokers had for sale on Alphabay was the Skimcountry implant, an alleged NSA implant released with their leak of purported NSA Tailored Operations tools, with 666 views at the time of writing. Of interest is a remote access tool called Nopen that is sold at the same price as the firewall implants. We theorize this price is due to its use of encryption, a wide range of supported architectures, and applicable operating systems [29]. Unlike many long term vendors, theshadowbrokers provided no additional detail about preferences and had no other transactions. This indicates they were only using the DNM to spread the word about the zero days and exploits they had for sale.

While their listings lead the marketplace in worth, theshadowbrokers are not leading the marketplaces with the amount of listings they have for sale. We indexed the vendors with the highest market share of unique listings across Alphabay and Dream Market as shown in Figure 3. Leading the pack with 535 unique listings is the vendor 'color' selling only on Dream Market with listings titled like *Make heavy Money with your RAT Keylogger Stealer* and *What is a Hacker*. This vendor is considered well trusted with a vendor rating of 4.78 out of 5 and 4,700 transactions. 'Color' is closely followed by 'HappyEyes' with listings that provide general knowledge like *Mastering The Windows XP Registry*. Both of these vendors sell their knowledge for approximately $1 USD each listing. This indicates that these listings are not primary sources of revenue and more intended to spread knowledge. This is further indicated in their profiles with personal entries saying 'Em (sic) Selling my products just for knowledge base.'

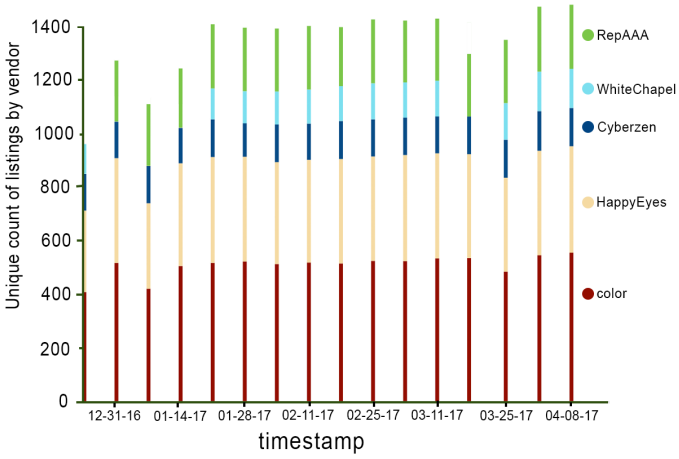

Fig. 3: Vendors with highest market share of listings

V. Limitations and Future Work

Due to how we implemented the framework to scrape assets, how we approached solving captchas, how we tracked listings over time, and gathering additional data sets to analyze for attribution there were several considerations we had to make. Scaling this framework efficiently required that most of the framework be automatable. Additional automation can occur by auto-detecting and scraping assets similar to Onionscan [8]. This approach weakens the possible data aggregated as gathered assets still need to be verified to contain usable data by humans and as such was avoided.

The ability to map listings over time deterministically depends on the DNM tracking the date the listing is created. Visualizations makes it easier to determine how a listing or DNM develops over time. While there were several DNMs that did not track the date a listing was posted, we made use of a timestamp when the asset was scraped to be able to track the listing over time. This is not optimal, as you must rely on the timestamp to determine when a listing originated which may not be as accurate as desired.

Attribution is a difficult process for both network operators and law enforcement. Different levels of evidence are required to provide attribution on a per-case basis [30]. That is, attributing an attack to a hacktivist or a nation state requires different levels of evidence to prove cause to different interested parties be they small and medium businesses, the public, or a government. Data points that could be used for this purpose include times of postings, alias names and meanings, language used in descriptions, and stylometry, but these fields are subject to the data included in each DNM listing. We took the best effort approach to glean as many attributable fields from a post as possible, but additional sources of data would be required to trace activity across both the darknet and the clearnet.

VI. Conclusion

In this paper we have presented D-miner, a modular framework for exploring darknet resources that allows for the

scraping, parsing, visualization, and alerting on darknet events. We show how D-miner fills gaps left by prior work in this area and discuss how the framework facilitates the scraping, parsing, and analyzing of data. Future features of D-miner will apply machine learning to solve captchas without DBC, increase the coverage across assets, and expand obfuscation options. Future work is planned to determine reliable points of attribution across data sources and explore automating correlation between clearnet and darknet identities.

We show that D-miner can fulfill the needs of many parties to gain insight into what takes place on the darknet. D-miner allows users to process information as they require through the use of third party tools. It also provides an interface to facilitate automation of operations pertaining to darknet data. D-miner is cross-compatible, tested on both Linux and Windows operating systems, completely open source under an MIT License, and ready to be deployed by analysts, security researchers, and law enforcement to further discover the secrets of the dark web.

ACKNOWLEDGMENTS

This work is supported by the Seed grant from Florida Center for Cybersecurity (FC2). The authors would like to thank Dr. Yier Jin and the Security in Silicon Lab for their guidance that directly led to the success of this project.

AVAILABILITY

Source code, documentation, and pre-configured Kibana dashboards are available through Github:

`https://github.com/infosecanon/dminer/`

REFERENCES

[1] E. Nunes, A. Diab, A. Gunn, E. Marin, V. Mishra, V. Paliath, J. Robertson, J. Shakarian, A. Thart, and P. Shakarian, "Darknet and deepnet mining for proactive cybersecurity threat intelligence," *CoRR*, vol. abs/1607.08583, 2016. [Online]. Available: http://arxiv.org/abs/1607.08583

[2] B. Krebs, "Source code for iot botnet 'mirai' released," Oct 2016. [Online]. Available: https://krebsonsecurity.com/2016/10/source-code-for-iot-botnet-mirai-released/

[3] "Iot devices being increasingly used for ddos attacks," Symantec Security Response, September 2016. [Online]. Available: https://www.symantec.com/connect/blogs/iot-devices-being-increasingly-used-ddos-attacks

[4] Verizon, "Verizon 2016 data breach investigations report," April 2016, pp. 48. Accessed 20 Jan 2017.

[5] C. Sillaber, C. Sauerwein, A. Mussmann, and R. Breu, "Data quality challenges and future research directions in threat intelligence sharing practice," in *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, ser. WISCS '16. New York, NY, USA: ACM, 2016, pp. 65–70. [Online]. Available: http://doi.acm.org/10.1145/2994539.2994546

[6] S. Brown, J. Gommers, and O. Serrano, "From cyber security information sharing to threat management," in *Proceedings of the 2Nd ACM Workshop on Information Sharing and Collaborative Security*, ser. WISCS '15. New York, NY, USA: ACM, 2015, pp. 43–49. [Online]. Available: http://doi.acm.org/10.1145/2808128.2808133

[7] C. Wheelus, E. Bou-Harb, and X. Zhu, "Towards a big data architecture for facilitating cyber threat intelligence," in *8th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2016, Larnaca, Cyprus, November 21-23, 2016*, 2016, pp. 1–5. [Online]. Available: http://dx.doi.org/10.1109/NTMS.2016.7792484

[8] S. J. Lewis, "Onionscan," October 2016. [Online]. Available: https://github.com/s-rah/onionscan

[9] The Shadowserver Foundation, 2004. [Online]. Available: https://www.shadowserver.org/

[10] G. Branwen, N. Christin, D. Dcary-Htu, R. M. Andersen, StExo, E. Presidente, Anonymous, D. Lau, Sohhlz, D. Kratunov, V. Cakic, V. Buskirk, and Whom, "Dark net market archives 2011-2015," July 2015, accessed 12 Dec 2016. www.gwern.net/Black-market%20archives.

[11] A. Greenberg, "The biggest dark web takedown yet sends black markets reeling," 2017. [Online]. Available: https://www.wired.com/story/alphabay-takedown-dark-web-chaos/

[12] K. Soska and N. Christin, "Measuring the longitudinal evolution of the online anonymous marketplace ecosystem," in *Proceedings of the 24th USENIX Conference on Security Symposium*, ser. SEC'15. Berkeley, CA, USA: USENIX Association, 2015, pp. 33–48. [Online]. Available: http://dl.acm.org/citation.cfm?id=2831143.2831146

[13] DeepDotWeb, "Updated: List of dark net markets (tor & i2p)," 2017. [Online]. Available: https://www.deepdotweb.com/2013/10/28/updated-llist-of-hidden-marketplaces-tor-i2p/

[14] "Elasticsearch," 2016. [Online]. Available: https://github.com/elasticsearch/elasticsearch

[15] "Death by captcha — best and cheapest captcha service!" August 2009. [Online]. Available: http://www.deathbycaptcha.com

[16] Elastic.co, "Kibana analytics and search dashboard for elasticsearch," 2016. [Online]. Available: https://github.com/elastic/kibana

[17] S. Kasatani, J. Huggins, P. Hanrigou, H.-B. Chai, S. Badle, and P. Lightbody, "Selenium project: Browser automation," March 2004. [Online]. Available: http://www.seleniumhq.org/

[18] J. Graham, E. Garrido, and A. Tolfsen, "Proxy for using w3c webdriver-compatible clients to interact with gecko-based browsers," January 2017. [Online]. Available: https://github.com/mozilla/geckodriver/

[19] DeepDotWeb, "Meet the market admin who was responsible for the ddos attacks," May 2015. [Online]. Available: https://www.deepdotweb.com/2015/05/31/meet-the-market-admin-who-was-responsible-for-the-ddos-attacks/

[20] P. G. Ipeirotis, "Analyzing the amazon mechanical turk marketplace," *XRDS*, vol. 17, no. 2, pp. 16–21, Dec. 2010. [Online]. Available: http://doi.acm.org/10.1145/1869086.1869094

[21] G. Bradski, *Dr. Dobb's Journal of Software Tools*, 2000.

[22] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a distance: Website fingerprinting attacks and defenses," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: ACM, 2012, pp. 605–616. [Online]. Available: http://doi.acm.org/10.1145/2382196.2382260

[23] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, ser. WPES '11. New York, NY, USA: ACM, 2011, pp. 103–114. [Online]. Available: http://doi.acm.org/10.1145/2046556.2046570

[24] T. Wang and I. Goldberg, "Improved website fingerprinting on tor," in *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, ser. WPES '13. New York, NY, USA: ACM, 2013, pp. 201–212. [Online]. Available: http://doi.acm.org/10.1145/2517840.2517851

[25] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, ser. SSYM'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 21–21. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251375.1251396

[26] J. Martin and N. Christin, "Ethics in cryptomarket research," *International Journal of Drug Policy*, vol. 25, pp. 84–91, 2016.

[27] A. Pathak, "An analysis of various tools, methods and systems to generate fake accounts for social media," 2014.

[28] L. Richardson, "Beautiful soup: We called him tortoise because he taught us," December 2004. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/

[29] C. Cimpanu, "Nopen is the equation group's backdoor for unix systems," Sep 2016. [Online]. Available: http://news.softpedia.com/news/nopen-is-the-equation-group-s-backdoor-for-unix-systems-508257.shtml

[30] B. Schneier, "Attack attribution and cyber conflict," September 2015. [Online]. Available: https://www.schneier.com/blog/archives/2015/03/attack_attribut_1.html