ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Ingegneria e Architettura

Corso di Laurea in Ingegneria e Scienze Informatiche

INTRODUCING CARONTE:
A CRAWLER FOR ADVERSARIAL RESOURCES
OVER NON TRUSTED ENVIRONMENTS

Relatore

Prof. MARCO PRANDINI

Co-relatori

Dott. LUCA ALLODI

Prof. GABRIELE D'ANGELO

Presentata da

MICHELE CAMPOBASSO

Terza Sessione di Laurea

Anno Accademico 2017 – 2018

Guai a voi, anime prave!
Non isperate mai veder lo cielo:
i' vegno per menarvi a l'altra riva
ne le tenebre etterne, in caldo e 'n gelo.

This work has been submitted to:

WACCO 2019: 1st Workshop on Attackers and Cyber-Crime Operations

Held Jointly with IEEE EuroS&P 2019

Stockholm, Sweden, June 20, 2019

Abstract

The monitoring of underground criminal activities is often automated to maximize the data collection and to train ML models to automatically adapt data collection tools to different communities. On the other hand, sophisticated adversaries may adopt crawling-detection capabilities that may significantly jeopardize researchers' opportunities to perform the data collection, for example by putting their accounts under the spotlight and being expelled from the community. This is particularly undesirable in prominent and high-profile criminal communities where entry costs are significant (either monetarily or for example for background checking or other trust-building mechanisms). This work presents `CARONTE`, a tool to semi-automatically learn virtually any forum structure for parsing and data-extraction, while maintaining a low profile for the data collection and avoiding the requirement of collecting massive datasets to maintain tool scalability. We showcase `CARONTE` against four underground forum communities, and show that from the adversary's perspective `CARONTE` maintains a profile similar to humans, whereas state-of-the-art crawling tools show clearly distinct and easy to detect patterns of automated activity.

Contents

Introduction

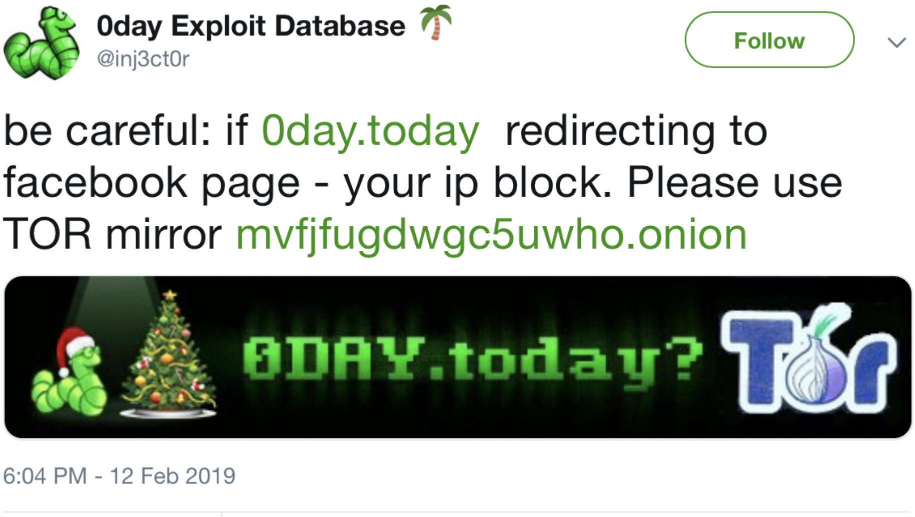
The monitoring of underground activities is a core capability enabling law enforcement actions, (academic) research, malware and criminal profiling, among many other activities. Currently, monitoring activities focus on the rapid collection of massive amounts of data [40], that can then be used to train machine learning (ML) models to, for example, extend available parsing capabilities to different forums or underground communities. Indeed, the proliferation of underground criminal communities makes the scalability of monitoring capabilities an essential aspect of an effective, and extensive, data collection, and ML has been the clear go-to solution to enable this. However, this comes at the high price of having to collect large volumes of data for training, raising the visibility of the researcher's activity and interest in the criminal community.

Indeed, the scientific literature showed that not all communities are born the same [23]; on the contrary, the majority of underground communities appear largely uninteresting (even when generating massive amounts of data about alleged artifacts [43]), both in terms of economics and social aspects [8, 45], as well as in terms of (negative) externalities for society at large [34, 40]. Whereas there are only a limited number of 'interesting' communities to monitor, gaining access to these may be less than trivial in many cases, particularly for forum-based communities and markets [7, 45]: high entry costs in terms of entry fees, background checks, interviews, or pull-in mechanisms are becoming more and more adopted in the underground as a means to control or limit the influence of 'untrusted' players in the community [7,45]. Under these circumstances, researchers and LE infiltrating underground communities may face significant opportunity costs whereby increasing monitoring activity may also jeopardize their ability to monitor the very community(-ies) in which they wish to remain undercover: network logs and navigation patterns of crawling tools (authenticated in the communities using the researcher's credentials) can easily put the real nature of that user's visits under the spotlight, and lead to blacklisting or banning from the community. This is particularly undesirable in high-profile communities where the cost of re-entry can be very high.

Anecdotal evidence shows that monitoring incoming traffic, for example for robot detection or source-IP checking, is a countermeasure that underground communities may employ to limit undesired behaviour. Some communities explicitly acknowledge the adopted countermeasures (see for example Figure **??**), others explicitly state that they are aware of the monitoring operations of LE and other 'undesirable' users; for example, the administrator of one prominent underground forum for malware and cyber-attacks that the authors are monitoring, states explicitly:

> *Forums like this are being parsed by special services and automatically transfer requests to social network accounts and e-mails.*

This significantly inhibits our ability to build scalable, reusable parsing modules, as the collection of large amounts of data to train the associated ML algorithms may be slow or carry significant risks of exclusions from the monitored communities. Pastrana et al. [36] lead the way in identifying *stealthiness* as a requirement for systematic underground resource crawlers, with many recent works not explicitly mentioning these aspects [30, 37].



0day Exploit Database 🌴
@inj3ct0r

Follow

be careful: if 0day.today redirecting to facebook page - your ip block. Please use TOR mirror mvfjfugdwgc5uwho.onion

6:04 PM - 12 Feb 2019

Figure 1: Example of inbound traffic monitoring from criminal communities

In this work we present `CARONTE`, a tool to monitor underground forums that: (1) can be configured to semi-automatically learn virtually any forum structure, without the need of writing ad-hoc parsers or regexps or collecting and manually classify large volumes of data; (2) implements a simple user model to mimic human behaviour on a webpage, to maintain a low profile while performing the data-collection. We showcase the tool against 4 underground forums, and compare the network traffic it generates (as seen from the adversary's position, i.e. the underground community's server) against state-of-the-art tools for web-crawling. Our results clearly show that both `CARONTE`'s temporal characteristic while accessing to multiple resources and the completeness of the downloaded resources to render linked to a page are significantly similar to humans when compared to other SOA tools. Good results have been achieved also when comparing the number of requests triggered inside of a thread.

Problem statement and scope of contribution

The collection of large amounts of information from cybercriminal high-profile communities depends on the fast and reliable scalability of the available crawling tools. However, as criminals become more aware of monitoring activities, crawling becomes adversarial from the perspective of the monitored community; account detection and banning can have severe costs for researchers as re-gaining access to high-profile communities can be an expensive and lengthy process. CARONTE addresses this problem by:

- Providing a semi-automatic method to learn (criminal) forum structures without relying on ad-hoc collections of large datasets that could expose the researcher's monitoring activities;
- Emulate user behaviour during the data-collection phase to minimize the differences between the crawling activity and legitimate user behaviour.

The essay proceeds as follows. In Section 1 we discuss relevant background and related work; Section 2 presents the tool, whereas Section 3 presents the experimental validation and results. A discussion of the impact and limitations of CARONTE are given in Section 4, and Section 4 concludes the essay.

Chapter 1

Background

In this chapter, we will analyze some aspects of cybercrime monitoring, related challenges and propose a solution.

1.1 Cybercrime monitoring

Cybercrime is a phenomenon that can be traced back tho the birth of Internet itself. The pioneers in that sense were mostly individuals that were discovering the effects related to the misuse of a software, often for joke [2] or as a proof of concept for vulnerabilities [41]. The first reported cybercriminal groups were reported during the 80s, when Masters of Deception and Legion of Doom started attacking mainframes of phone companies. From that moment, the hacking phenomenon started to assume relevance and was no longer treated like a cyberpunk wave acted by bored teenagers [33]. Besides channels like IRC, bullettin board systems started to rise, offering hackers a place where to meet, exchange information and tools. Some studies have highlighted that hacker communities are organized as meritocracies where participants are stratified into high-skilled hackers and low-skilled ones. [24, 26]. They cooperate between them sharing information, exchanging tools and organizing to pursue their goals. Lately, sophistication to prevent everyone to see the content on their platforms made the criminal move from the surface web to the Dark Net, thanks to the birth of The Onion Router - TOR, which gives participants communication anonymity properties far superior to the alternative solutions (e.g. anonymous hosting). Recent studies on the black market dynamics have portrayed how criminals operate in the space of drugs, digital goods and weapons retailing [40]. For what concerns more the aspect of malware, carding, malware production and retailing and botnet renting, researchers have created *ad-hoc* tools for scraping adversarial platforms that don't scale up with the number of sources and the variety of the content; nonetheless, most of them were more concerned on developing techniques that enable underground economy discovery, key hacker identification [6] and threat detection [12], disregard stealthiness in favour of parsing volumes [30, 37], with few notable exceptions [36].

1.2 Adversary models and their evolution

Adversaries are increasingly aware of the mounting interest from scientific and nation state sponsored investigations, pushing them to start developing techniques to avoid unwanted actors and data gathering in their communities [35]. Part of these techniques are centered on the identification of a member at the act of registration on these platforms; these strategies are possible to circumvent, since creation and development of fake profiles on the Internet for going undercover is a relatively easy task to accomplish. Nonetheless, this task requires a lot of time and effort. Another set of approaches adopted by our adversary is based on auditing the traffic on the web servers, researching for patterns and anomalies, in order to detect the use of web crawlers and, if necessary, shutdown the relative account [19,38]. Filters to community participation are also often employed by high-profile markets that rigorously monitor and assess the inclusion of new members to their community [7], therefore increasing the bar and the associated cost for reserachers to evaluate underground activities.

1.3 Robot/Crawler Detection

With the birth of the big data society, crawling has become a conspicuous portion of the Internet traffic [10] and an unwanted practice from website owners, due both to network resource consumption and to the lack of an explicit permission to a third-party to massively download all the website content for unknown goals, often resulting in a privacy violation [20, 46]. If it's true that ethical crawlers do obey to the webmaster's will, on the other hand robots that ignore these rules have increased as well. In order to tackle this issue, several anti-crawling techniques have been developed; a great number of these are based on minimal traffic patterns analysis, often focused on monitoring characteristics of HTTP traffic observable from logs. The monitored characteristics include the rate of requests, the length of browsing sessions, lack of cookie management, presence of bogus user agent, JavaScript execution, access to *robots.txt* file, usage of HEAD HTTP requests, cherry-picking of requested resources and the lack of a referrer in HTTP requests [11, 16, 25, 29, 39, 42, 46]. These strategies are quite simplistic and don't provide consistent and reliable crawler detection, ignoring the chance that a focused and stealthier crawler can act in disguise, tampering with information in the requests. Since our adversary could legitimately have advanced skills in computer matters, these requisites remain relevant, but aren't sufficient to keep a robot undercover. Additional efforts have been made for creating more reliable methods; the state-of-art techniques for detecting automated activity on a website include pattern recognition, like loopholes detection and breadth first or depth first strategies, JavaScript fingerprinting and tracking, and Turing tests, on top of other strategies [15, 28, 44]. Turing tests as CAPTCHAs can be outsourced at extremely low prices [3] or solved via OCR [27] and the production of not suspicious traffic can be obtained with a focused crawler that acts with some precautions. Moreover, the context of our studies brings us to platforms in the dark-web where Turing tests that require JavaScript enabled are almost non existent, due to linked risks (e.g. allowing in the past to bypass completely the anonymization of Tor [1, 4]).

Zhang et al. [46] propose a dynamic blocker for crawlers analyzing some traffic patterns, such as the complete exploration of the resources linked to a page (attachments, links, ...), the nonacceptance of cookies, bogus user agents in HTTP requests and high fetch rates. Sardar et al. have developed a framework that statically analyzes logs, identifying some features such as robots.txt file access, source IP addresses, user agent and counting HEAD HTTP requests with undefined referrer [39]. Stevanovic et al. introduces some additional checks compared to the previous analyzed works, such as the *HTML/image ratio*, which tends to be very high for crawlers, the *number of PDF/PS file requests*, the percentage on total requests of answers with 4xx error codes and unassigned referrers, which show high scores for robots [42]. Doran et al. propose to recognize crawlers in real-time [15]. In particular, their work they provide a 4-tier analysis based on *Syntactical log analysis*, *Traffic pattern analysis*, *Analytical learning techniques*, and *Turing test systems*. For what concerns more behavioral patterns, Kwon et al. have studied how crawlers generally have a *monotonous behavior* in the type of requested resources. Their attempt therefore is to classify crawlers based on the *"switching factor"* between text and multimedia contents [28, 29]. Other studies regarding behavioral patterns are by Balla et al., who analyze the time between one request and another and when these are issued (like during night time, making it more suspicious) [11].

Crawler detection patterns aside, data gathering, ready to use and well structured, is not an easy task to accomplish. Forum structures may vary a lot, depending on the forum paltform adopted and their configuration. In particular, the goal is not to scrape the entire pages to dump them on disk, but to extract and structure data for further analysis; For this reason, the crawler need to be instructed on what resources are required to be collected, how to reach them and what do they mean. Therefore, a knowledge base should be created for the crawler in a reliable way, enabling the forum traversal in the required areas through the identification of the existing resources of each page of interest.

1.4 Modelling user behaviour

Several studies evaluated models of user browsing behaviour, broadly distinguished between **click patterns** and **time patterns**.

Click patterns. Click models are used to evaluate user decisions in considering a topic or hyperlink relevant to the specific purpose of their navigation or query [18]. Derived approaches consider *single-browsing* and *multi-browsing* models to infer user behaviour as a function of the purpose of the navigation, in particular distinguishing between *navigational* and *informational* queries, whereby the user wants to reach a specific resource (likely producing one click at a time), or is interested in exploring new information (likely producing multiple clicks at a time) [18, 21]. These models show that past behaviour or user interest are useful predictors of which clicks will happen in the future [18]. In our context, forum-browsing clearly covers both dimensions, depending on whether the user aims at retrieving specific information (e.g. updates in a thread of previous interest to the user), or to explore the content of a forum section.

Time patterns. More broadly, these dynamics are explained in the information retrieval literature as dependent on the user's task [9]. The decision of a user to click on a specific resource depends on its perceived and intrinsic relevance w.r.t. the user's goal, and is bounded by how many topics need be opened to find the answer the information the user is interested in [17]. Post-click user behaviour (i.e. what the user does one he or she reaches the clicked resource) has been shown to be directly related to the relevance of the document [22]. Post-click behaviour includes variables associated with mouse movements, scrolling, and eye-tracking [18, 22], clearly showing that what the user does, and how much time the user spends on a webpage, varies as a function of the relevance of the webpage. Indeed, a user's *inaction* on a webpage has been shown to be relevant to model the quality of dynamic systems such as recommendation systems [48]. Part of that behaviour can be quantified by considering how quickly users can be expected to process the relevant information [32]. Data around this subject is scarce and quite diverse; some sources refer the average reading speed to be around 200-250WPM (Words Per Minute) with a comprehension rate of 50/60% [32], others report that for reading some technical content with a good proficiency, the speed can be around 50-60WPM.

Chapter 2

CARONTE

2.1 Design

From the literature analysis in the previous section, we derive a set of desiderata for `CARONTE`.

2.1.1 Functional and behavioural requirements

First and foremost, `CARONTE` must be able to semi-automatically learn forum structures without the need for extensive pre-collected datasets on which to train automated models [37]. This should be a one-time only process, employed for each new forum structure that has not already been learned. Further, `CARONTE` must have the ability to diverge from crawler behavior and, where possible, to mimic human behavior. In this regard, as emerged from the time patterns paragraph, keeping in mind that one significant aspect of crawlers is their greed in resources, `CARONTE` shouldn't exhibit high fetch rates and mimic as much as possible human's time to browse and read resources, whether the content is appealing for it or not. Therefore, we model `CARONTE` to mimic interest to a specified subset of the forum, exploring only certain sections of it, accordingly to the hypothetical goals of our modelled actor. Then, CARONTE will be able to receive instructions about which areas are valuable to crawl and which to skip. The forum contents will be explored both through *navigational* and *informational* queries; in particular, CARONTE will access quickly resources, like posts in threads already read and the resources related to path traversing that occur from the landing page to the section of interest, while it will take more time and produce less frequent clicks while staying on pages with new content from the section of interest. To improve its stealthiness on this aspect, we design a navigation schedule on a forum like an actual human being having in mind varaibles such as time of day and stochastic interruptions.

Therefore, we model a navigation schedule for the tool that is comparable to human being's real-life needs like work, out-of-routine events and physiological needs. The time slots in which the tool will work vary at every run. The tool will have the opportunity to skip a navigation slot during the day or take some pauses during the browsing activity. Another aspect emerged in literature is that crawlers tend to download any possible resource they find on the analyzed page. CARONTE instead will collect the entire pages but will focus on the textual content of them, avoiding to download files provided in the threads. For the final part of the human-like behavior, Further, we consider that an actual user will never explore the forum in the whole, but will focus only on certain sections that are relevant to his interests. Therefore, CARONTE will be able to receive instructions about which areas are valuable to crawl and what to skip.

2.1.2 Technical requirements

To avoid detection at the network level, CARONTE will have to act indistinguishably to a regular browser in terms of generated traffic and differing to regular crawlers. The primary aspect is to produce not suspicious HTTP requests against the webserver; crawlers' traffic is characterized by the adoption of HEAD HTTP requests to determine whether the resource to download is of their interest or not, non-filling of referral link field in requests and by the usage of a bogus user agent [11, 15, 25, 39, 42]. Also, depending on the goal of the crawler, they might be interested in scraping content without rendering and providing the opportunity to browse it, thus missing the support for a proper browser engine that will allow to consistently handle cache, manage cookies and execute JavaScript. Further, crawlers might be interested on fetching only text content, refusing to download styles, images and JavaScript, (e.g. to minimize network footprint) or won't actively execute client-side code such as Javascript, handle sessions and cache as a 'regular' web browser would do.

With this in mind, we identify the need of a fully functional browser that by design covers all these aspects coherently with a legitimate one, but that offers the possibility to be maneuvered programmatically. Table 2.1 provides an overview of the identified requirements for CARONTE.

2.2 Architecture and implementation

CARONTE aims to get information regarding the structure of each forum, enabling the crawling process to be the most focused possible, reducing the amount of traffic generated, avoiding to collect not relevant data and resources that are redundant and that may represent canaries in crawler detection (like following link that rearrange content in a page, for each page). Therefore, before crawling, it is necessary to instruct the tool how to traverse the forum, what are the required resources, what data is valuable to collect and what is

Requirement	Description	Implementation
Learning forum structures	Understanding forum structure, how to browse it and where valuable information is	Creation of a supervised learning module that identifies needed resources
Regular browser behaviour	Realistic user agents, caching behaviour, referral handling	Exploration of required sections only, throttling requests accordingly to text volume of the page, mimicking reading time. Confining crawling activity in semi-random time slots during the day and suspending it for random amounts of time during the day.
Realistic browser configuration	JavaScript engine, pages download feature	NoScript and changing default to refuse all active content, preparation of the browser to support shortcuts for downloading a page
Anonymity	Browsing session needs to be anonymous	TOR Browser adoption and JavaScript disabled

Table 2.1: Summary of identified requirements for `CARONTE`

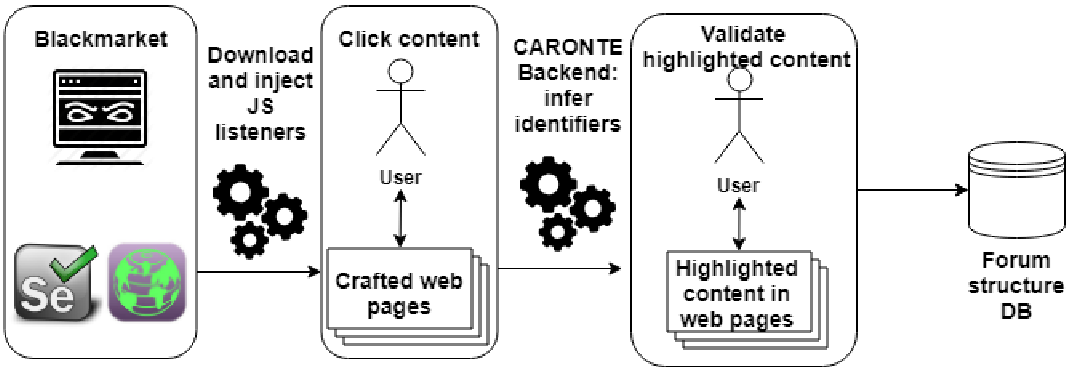

Blackmarket
Download and inject JS listeners
Click content
User
Crafted web pages
CARONTE Backend: infer identifiers
Validate highlighted content
User
Highlighted content in web pages
Forum structure DB

Figure 2.1: CARONTE trainer module structure.

the meaning of the collected information. For this reason, we propose adopts a two-tier architecture, separating the *training* from the *crawling* operations.

2.2.1 Trainer module

Base mechanism

The trainer module has the task to build a knowledge base for traversing the forum structure 2.1. For each page where relevant content or fields are present, the trainer will load, save and render a modified copy of it to the user. For each of them, the operator will be asked to click on the desired resources inside of the rendered page. Before being rendered, pages are preprocessed; in particular, we inject JavaScript scripts to allow `CARONTE` to gather the events triggered by the human operator. With different combinations of `onclick()` and `addEventListener()`, we control these interactions and generate AJAX requests against `CARONTE`'s backend. The payload of these requests is a resource identifier (see "Resource identifiers" paragraph in this section) that will allow the crawler module to access to the required information or interact with it, where necessary. Subsequently, it then proceeds to render again the saved page, but highlighting the previously identified content, allowing the user to confirm if the identifiers for the resources have been inferred correctly by the tool or not (Figure 2.3).

In some cases user-generated clicks are not possible or we aim to identify a group of resources. For example, this is the case for identifying multiple posts inside of a thread; for this kind of resources, our goal is to infer a resource identifier that can operate like a regular expression, enabling the tool to resolve all the required elements on the page. Our strategy here is based on the collection of multiple snippets of text contained in each of these resources (Figure 2.5). For each of the received fragments, `CARONTE` will query the JavaScript engine embedded in the browser handled by Selenium in order to resolve their identifiers and, through syntactical similarity, generate a matching one. Text content will be gathered with the help of the human operator in a special page (here referred to as *content collector page*) that is presented to the user together with the original page. As instance, for the case of post content, the content collector page will ask the user to put five snippets of posts to be sent to `CARONTE`'s backend. After having calculated the generic identifier, the page will be reloaded, highlighting the content inferred like in the previous case.

In each of these steps, the trainer module will instruct and assist the user to achieve this goal through a wizard procedure.

Resource identifiers

The desired resources can be identified through two different approaches: **XPath** or **HTML common classes**. XPath is a standardized query language that identifies elements inside of a XML-like document; it supports regular expressions for matching several elements. HTML classes instead are attributes assigned to nodes of an HTML file for which different styles are assigned. Even

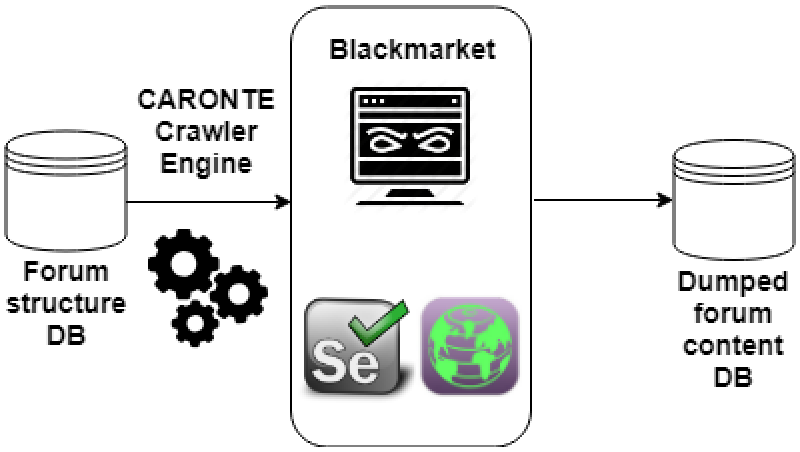


Figure 2.2: CARONTE crawler module structure.

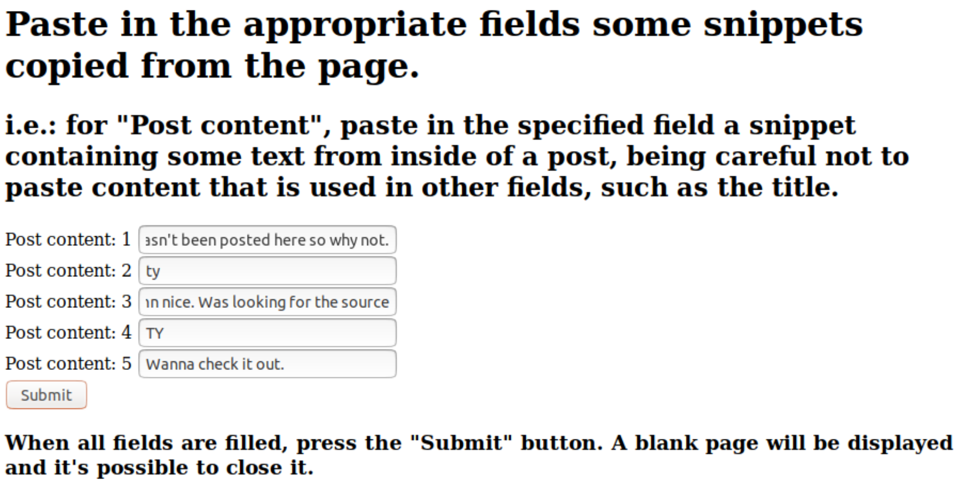
though XPath is an *ad-hoc* technique for identifying elements in a HTML page, sometimes inferring HTML classes is easier than XPaths. During the training phase, when a resource is clicked, the loaded page will identify the associated identifier through a series of heuristics, and send it to the backend. If the resources are multiple, the *content-collector page* will be rendered with the downloaded page and the user will fill the fields with the required data. The process to identify the most likely resource identifiers depend on the data structure and the number of classes associated with that resource. CARONTE supports the following four cases:

- **Technique 1**. Extract the XPath of the exact resource. If the resources are multiple, the most frequent XPath will be the candidate;

- **Technique 2**. Extract XPath of the exact resource, but the last node is truncated. The XPath approach may fail due to the presence of extra HTML tags (e.g. due to text formatting), that can then be disregarded. If the resources are multiple, the most frequent XPath will be the candidate after removing the last node;

- **Technique 3**. The class of the exact resource. If the resources are multiple, the most frequent class will be the candidate. This approach solves the problem of calculating an XPath in a page where the content is dynamic, resulting in a non predictable XPath for a certain resource, depending on the loaded content in the page. If the resources are not assigned to a class, the element will be replaced with its parent, which will act as a wrapper for it;

- **Technique 4**. Two classes of the exact resource. If the resources are multiple, the two most frequent classes will be the candidates. This approach is adopted to handle elements in a page that exhibit the same class of the desired content, resulting in a misclassification. Therefore, this strategy allows to have a stricter condition on the searching criteria for the required resource. If the resource(s) has no class, the element will be replaced with its parent, which will act as a wrapper for it.



Paste in the appropriate fields some snippets copied from the page.

i.e.: for "Post content", paste in the specified field a snippet containing some text from inside of a post, being careful not to paste content that is used in other fields, such as the title.

Post content: 1 ɜsn't been posted here so why not.
Post content: 2 ty
Post content: 3 ın nice. Was looking for the source
Post content: 4 TY
Post content: 5 Wanna check it out.

When all fields are filled, press the "Submit" button. A blank page will be displayed and it's possible to close it.

Figure 2.4: Gathering of text snippets from the saved page (in next tab).

2.2.2 Crawler module

Based on the structural details collected with the trainer module, the crawler module will traverse the forum to reach the required resources, explore threads and collect all the required data. The crawler will also embody the requirements of being compliant with the traffic generated from a regular browser while camouflaging its nature adopting low fetch rates for pages. How time is calculated before accessing to the next resource is deepened in the Reading Time paragraph under Implementation section.

After creating a knowledge base about the current forum to crawl, it is possible to gather the information required from it. `CARONTE` further keeps track of updated threads and selects those opportunistically for visiting. Threads that have not been updated are not traversed a second time.

2.3 Behavioral aspects

In this section, tools, technologies and technical solutions will be discussed in relation to their purpose.

2.3.1 Legitimate browser traffic - Browser

In the previous section, we've identified the need of a programmatic browser that can impersonate our modelled human, capable of browsing and exploring autonomously the resources in a forum while handling all the traffic aspects typical of a regular browser. After some research online, we identified some tools that could have been the candidate for our studies. We needed a solution that could interact actively with elements in a page while generating legitimate browser traffic. It turned out that solutions for testing web applications could fit our needs. **Spynner** was a candidate; based on WebKit and PyQT, it exploits JQuery injection in the page to interact with them. This represents a problem because we want to keep JavaScript disabled inside the browser engine and for therefore it has been ignored; **Splinter** is another automated web application tester. It has been rejected since seems an almost abandoned project and there's no support for Tor Browser integration, despite the appreciation of a good portion of users.

To implement `CARONTE`'s browser functionalities we adopt **Tor Browser Selenium**, or *tbselenium* for short. *tbselenium* accesses *geckodriver*, the browser engine branded Mozilla that allows to maneuverer the browser's behavior and UI. Moreover, *tbselenium* exposes an interface for customizing the environment and, finally, produces traffic identical to Tor Browser.

2.3.2 Element identification and verification

After downloading a page, the downloaded artifact passes through different steps of processing. First, it is purged from all the contained JavaScript in it, in order to avoid any possible unwanted execution outside of the Tor Browser sandbox. Then, depending on the goal, one of two procedures are adopted:

1) If the resources to be identified are clickable fields or links, the downloaded page will be injected with JavaScript that allows to locate the clicked elements through their XPath. Instead, if the resources are not clickable or are multiple and concur to find a common rule for identifying them, a *content-collector page* will be created alongside the downloaded one; this second page will ask the user to collect the required information from the mirrored page and submit them through one or more AJAX calls against `CARONTE`'s backend. The *content-collector page* is a static HTML page that instructs the operator what is the information required for the current step of the training. In particular, the user is demanded to paste in it some snippets of text copied from the

required fields in the downloaded page (e.g. post content as in fig. 5). In this case, it is required to fill 5 snippets from 5 different posts; this is needed both because the post content is a non-clickable resource and because is necessary to have different contents to infer a XPath regular expression.

After the submission of the information, when the page is closed, the backend will receive notice of the completion of the data collection and will stop listening. The user is then prompted with a replica of the webpage, highlighting the content identified in the previous steps, and is asked if the content is correctly identified. If it is, the tool will move to the next resource to be identified, downloading next needed page and so on, else it will try another identification technique. The learning process ends when all the identifiers for the needed resources have been calculated.

2.3.3 Mimicking legitimate human traffic

Work schedule

In order to not produce suspicious network traffic on the forum, we model a potential human actor. In particular, we projected crawling time slots compatible with the alleged possibilities of an employee, from Monday to Friday, 9-17. The possible sessions are morning, afternoon and evening. Our fake user can be configured to work within pre-defined timeslots during the week or in the weekends, late afternoons and evenings during the week and all three sessions on weekend. Between each session, a randomized time of inactivity simulates short pauses (between 5 minutes and half an hour) and longer ones at pre-defined times (e.g. 2 hours around dinner time). These can be configured.

Each session has a start time and an end time; each of them can vary of up to 25% of the total duration of the crawling session randomly. Each session has the 20% of chance to be skipped. Nonetheless, we would avoid to have 24 hours of inactivity, so if there's no sessions scheduled in the next 24 hours, a compatible one with the default schedule will be executed. Start and end times are shifted accordingly to the timezone of the geographical location of our forum user profile.

Moreover, we have given our modelled threat actor also a nationality; start and end times are shifted accordingly to the timezone of the nation or place we want to give to the robot. As instance, if we crawl a Russian-speaking forum, we want to adopt the timezone of a city in, e.g., the Russian Federation, therefore shifting our start and end time accordingly.

Finally, we have also modelled the chance of taking a break for biological needs or whatsoever. The crawler in fact every minute can have a 2% chance of taking a break that lasts an amount of time between 5 and 30 minutes.

Reading time

The time spent between two requests is calculated according to two main criteria:

- If the current page doesn't show significant content to be read (e.g. pressing login button, reaching the section of interest of a forum, moving to page 2 of a forum section, ...) or the content has been already read (a thread may contain new messages, therefore old will be skipped), the time spent before going to the next page is a random number of seconds between 3 and 7. This decision is based on the fact that the information on the page is more essential and visual. This enables our fake actor to skim rapidly and choose what to read, resulting to fulfill the expectation of having a *navigational queries* pattern;
- If the current page is the body of a thread, the tool will wait, for each

unread post, an arbitrary amount of seconds calculated as the time to read the post at a speed in the range of 120-180 WPM. This behavior validates the expectation of producing *informational* queries.

User event generation

`CARONTE`'s modeled user goal is to reach the threads of interest and iterate them to extract their content. When starting the crawling process, `CARONTE` loads the forum homepage, as it was typed on the address bar, then reaches the login page. Once logged in, it reaches one of the sections of interest expressed during the training and opens a thread per time (if it has been never read or has new replies) and browses each page until the thread has been read in the whole. The click patterns generated match the purpose of our fake user, which considers relevant the content of pages with a significant quantity of text like a thread instead of a login page.

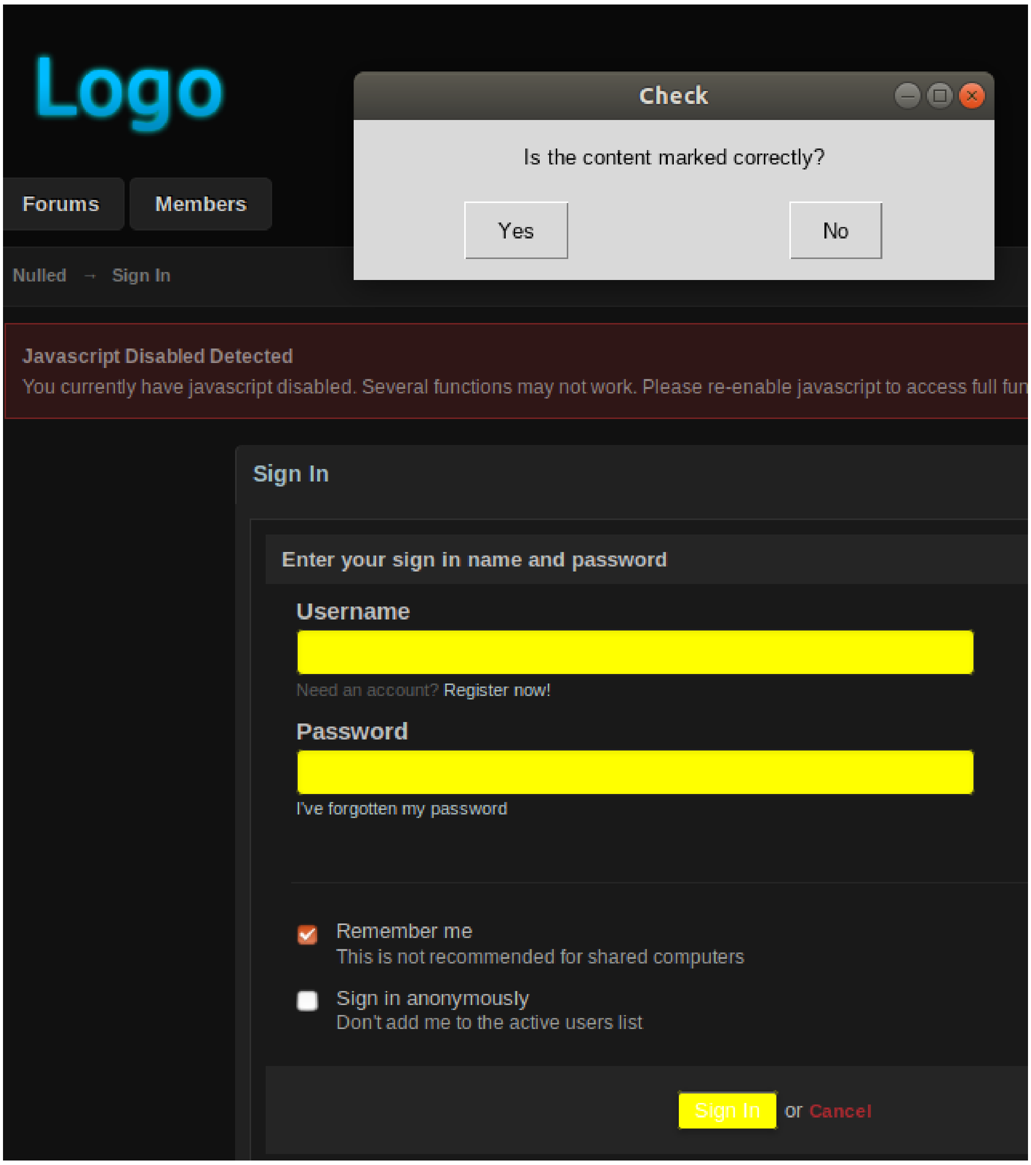
2.4 Collected data

As mentioned, the final goal of `CARONTE` is to collect relevant data and label it properly. Data is mostly posts, each of which is structured as follows:

- *Hash*: it is a unique identifier for the whole tuple that allows `CARONTE` to recognize if a post has been already dumped, skipping it without waiting;
- *Website ID*: a reference to the website where the post comes from;
- *Thread URL*;
- *Thread name*: the title of a thread usually reports some information, like an item on sale etc;
- *Author name*: relevant for profiling buyers and sellers;

- *Author's post count*: degree of activity in the marketplace;
- *Post date*;
- *Post content*: the core of the analysis.

This data could be used for training LDA modules for accomplishing Sentiment Analysis and Topic extraction.



Check

Is the content marked correctly?

Javascript Disabled Detected

You currently have javascript disabled. Several functions may not work. Please re-enable javascript to access full fun

Sign In

Enter your sign in name and password

Username

Need an account? Register now!

Password

I've forgotten my password

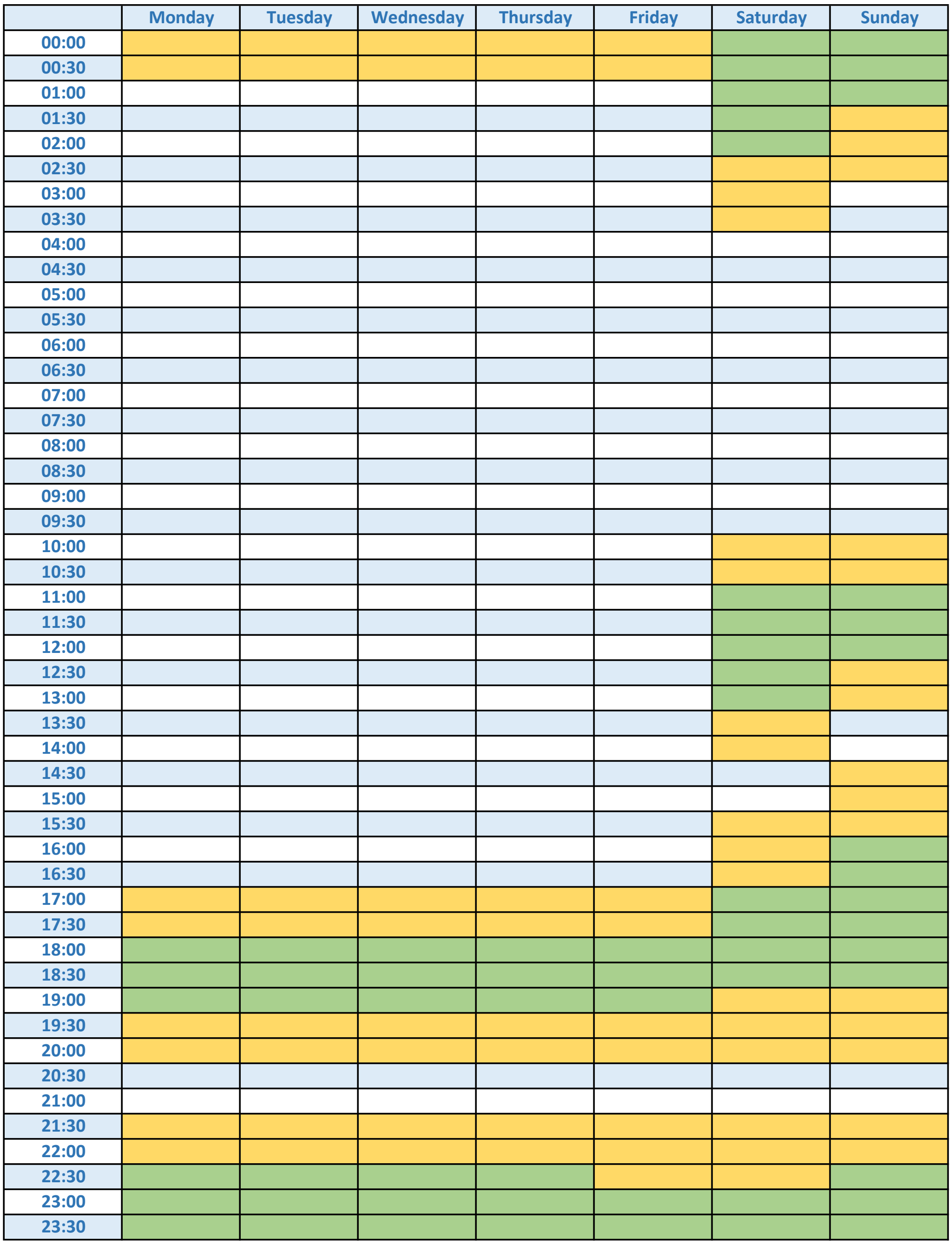
☑ Remember me

This is not recommended for shared computers

□ Sign in anonymously

Don't add me to the active users list

Figure 2.3: Validation of identifiers inferred.



	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
00:00	🟨	🟨	🟨	🟨	🟨	🟩	🟩
00:30	🟨	🟨	🟨	🟨	🟨	🟩	🟩
01:00						🟩	🟩
01:30						🟩	🟨
02:00						🟩	🟨
02:30						🟨	🟨
03:00						🟨	
03:30						🟨	
04:00							
04:30							
05:00							
05:30							
06:00							
06:30							
07:00							
07:30							
08:00							
08:30							
09:00							
09:30							
10:00						🟨	🟨
10:30						🟨	🟨
11:00						🟩	🟩
11:30						🟩	🟩
12:00						🟩	🟩
12:30						🟩	🟨
13:00						🟩	🟨
13:30						🟨	
14:00						🟨	
14:30							🟨
15:00							🟨
15:30						🟨	🟨
16:00						🟨	🟩
16:30						🟨	🟩
17:00	🟨	🟨	🟨	🟨	🟨	🟩	🟩
17:30	🟨	🟨	🟨	🟨	🟨	🟩	🟩
18:00	🟩	🟩	🟩	🟩	🟩	🟩	🟩
18:30	🟩	🟩	🟩	🟩	🟩	🟩	🟩
19:00	🟩	🟩	🟩	🟩	🟩	🟨	🟨
19:30	🟨	🟨	🟨	🟨	🟨	🟨	🟨
20:00	🟨	🟨	🟨	🟨	🟨	🟨	🟨
20:30							
21:00							
21:30	🟨	🟨	🟨	🟨	🟨	🟨	🟨
22:00	🟨	🟨	🟨	🟨	🟨	🟨	🟨
22:30	🟩	🟩	🟩	🟩	🟨	🟨	🟩
23:00	🟩	🟩	🟩	🟩	🟩	🟩	🟩
23:30	🟩	🟩	🟩	🟩	🟩	🟩	🟩

Figure 2.5: Time schedule of our modelled agent. In green, the time that will be always covered if that session takes place; in yellow, the time slots that represent the floating margin.

Chapter 3

Experimental validation

3.1 Forum selection

In order to proof `CARONTE`'s capabilities against different forums, we selected four real-world criminal forums built on top of different platforms. The candidates (Table 3.1) correspond to a consistent representation of the most common forum platforms wildly adopted on the Web [5, 6, 12, 37, 40].

We first reproduced four live hacker forums by scraping them and hosting their content on a local server at our Institution. Before reproducing the content on our systems we inspected the source code and scanned it with `VirusTotal.com` to assure malicious links or code was not present. Forum mirrors include multimedia content, styles and JavaScript. To avoid provoking misservice on the server side while scraping the forums, we avoided aggressive scraping. As our interest is to have an appropriate test-bed to evaluate `CARONTE`'s overall performance, the nature (or quality) of the content of the forums is irrelevant for our purposes.

Forum	Time span	Forum software	Obtained with
https://nulled.io	Jan 2015 to 6 May 2016	IP Board 3.4.4	Online dump
http://offensivecommunity.net	Jun 2012 to 6 Feb 2019	MyBB (unknown version)	HTTrack 3.49.2
http://darkwebmafias.net	Jun 2017 to 7 Feb 2019	XenForo 1.5	A1 Website Downloader 9
http://garage4hackers.com	Jul 2010 to 4 Feb 2019	vBullettin 4.2.1	A1 Website Downloader 9

Table 3.1: Scraped forums for our testbed.

3.2 State-of-art tools selection

To provide a comparison of CARONTE's capabilities against other tools, we select three among the available ones:

- *A1 Website Download*: shareware crawler specialized in downloading forum content. Through a fine-grained customization wizard, it is possible to use configuration presets that fit better the crawling process against a certain forum software;

- *HTTrack*: probably the most famous tool for downloading websites, HTTrack provides several tweaking features through regular expressions for downloading a generic website;

- *grab-site*: fully open-source, grab-site is a regular crawler for downloading large portions of the web, powered by the Archive Team.

3.3 Training phase

The approach adopted by CARONTE to discover the structure of a forum has proven effectiveness over our tests. In order to get the structure of a forum, we rely on the predictability of the structure of a forum in the future in terms of XPath and HTML classes. This is true in the majority of the cases; from the literature analysis and empirical evaluations of the most common forum structures [14,31,47], we found no evidence of dynamically-loaded forum structures that would alter the DOM structure at each visit or while being on a page. This seems well in line with environments like the Dark Web, where platform simplicity and functionality, as well as predictability, are desirable [45].

3.3.1 Problems and solutions

Post details mismatch avoidance. We've found out that seldom post details like authors and date have different structural identifiers or are displaced differently, causing the crawler module to miss them and associate a post's details to another, due to the different cardinality of the identified ones (8 author names missed in 12854 posts for IP Board 3.4.4). Even though CARONTE is not able to recover them, we model the post as an *unique container* (which we call **post wrapper**) where details are anchored to it. By doing so, the identifiers are calculated relatively to the post and we thus avoid accidental post assignment to wrong ids.

Inconsistent reference to navigation button in forums. Depending on the forum platform and on the adopted configuration (e.g. forum skins or themes), HTML tags may have different names and usages. For example, vBullettin 4.2.1 and XenForo 1.5 adopt the same HTML tag id or class for both the forward navigation button and back inside of a thread or section of a forum. During the training stage of CARONTE inferring the class of this element leads to

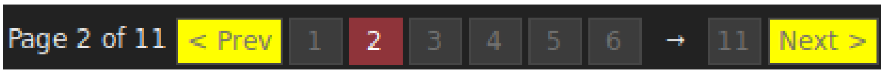


Figure 3.1: Resource name collision.

Forum	Exact XPath	Parent XPath	Single Class	Double Class
https://nulled.io	10	1	2	0
http://offensivecommunity.net	9	2	2	0
http://darkwebmafias.net	9	2	2	0
http://garage4hackers.com	8	1	3	0

Table 3.2: Treatment combination and experiments.

the unwanted result of identifying both buttons with the same rule (Figure 3.1). This would result in moving back and forth between the first and the second page. CARONTE manages this issue in the training phase by loading the second page and asking the user if the highlighted part of the DOM corresponds to one element only, or more. In the second case, CARONTE keeps record of the conflict and accesses to the second retrieved element when this case occurs.

3.3.2 Training evaluation

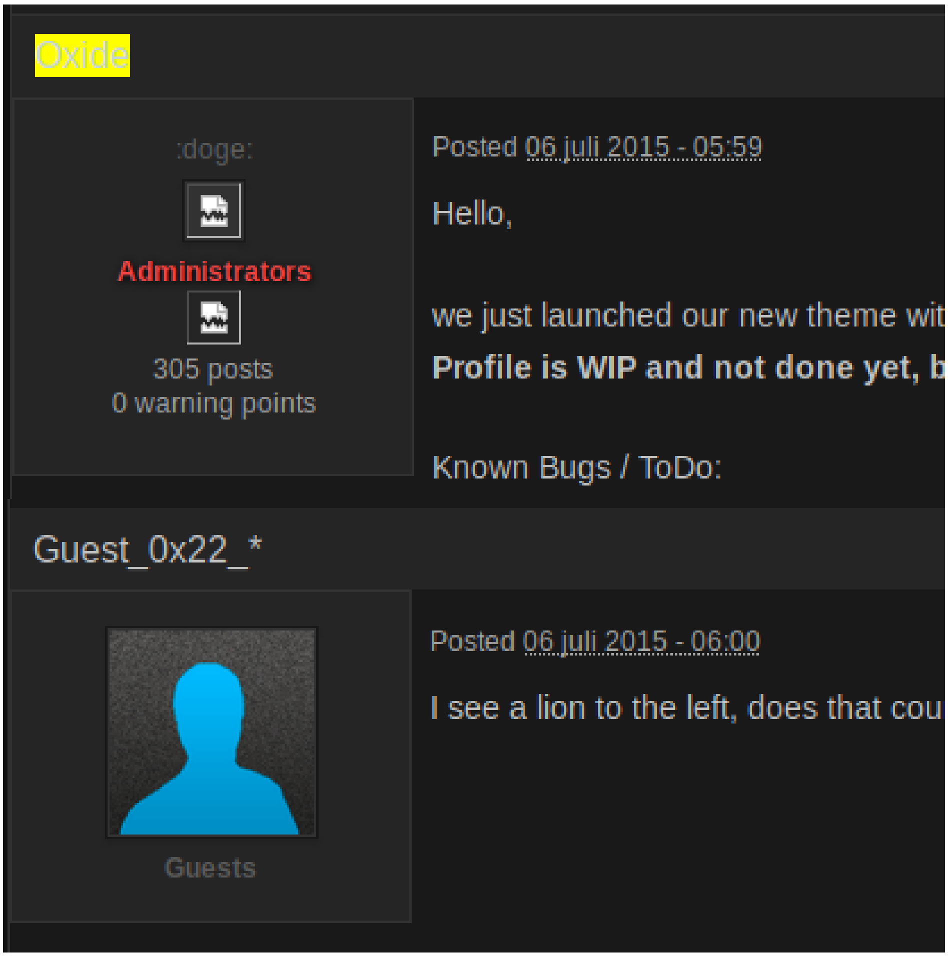
Depending on the peculiarities of the forum against which CARONTE has been trained, different strategies have been adopted to determine the resource identifiers ("Resource identifiers" paragraph). A summary about the identification strategies used per forum is available in Table 3.2. In greater detail, for *OffensiveCommunity.net* and *DarkWebMafias.net*, the first post of a thread has some differences in the HTML structure compared to other posts. In particular, the field of the post author is wrapped around some extra nodes that provide a special style to it. With the adoption of the parent XPath (**technique #2**), it has been possible to infer a rule that works for every post's author.

For *Garage4Hackers.com* the XPath regex was not a sufficient approach to find all the post wrappers in a page. This is due to the fact that, when multiple resources are meant to be identified through an XPath regex, the XPath `//*[starts-with(@id, 'something')]` selector is used, which returns an array of nodes. Specifically, we were interested in nodes with id `post_XXXXX`, but on the same page were also present nodes with id `post_message_XXXXX`, which caused the resolution of both content types. To overcome to that, identifying the required resources through a class was sufficient to solve the problem (**technique #3**).

For all the forums, identifying the regex for the post wrapper is a special process that uses a **variant of technique #2**, where the container is identified by incremental steps. Starting from snippets of text from different posts of the page, a first XPath is calculated. Subsequently, with the user interaction, it is refined removing the unnecessary child until the whole post is correctly classified with the XPath calculated.

Moreover, for all the forums, inferring a stable XPath identifier for the next page buttons is not possible. This depends from the number of buttons inside of the navigation wrapper (Figure 3.2), which changes depending on the number of available pages or even when moving to the second page. To circumvent this problem, again a class comes in help **(technique #3)**.

The double class (technique #4) has never been used with the 4 forums analyzed. Nonetheless, it was proven to be necessary for another forum, a XenForo board, which was used benchmark for some first experiments.



Oxide

:doge:

Administrators

305 posts
0 warning points

Posted 06 juli 2015 - 05:59

Hello,

we just launched our new theme wit

Profile is WIP and not done yet, b

Known Bugs / ToDo:

Guest_0x22_*

Guests

Posted 06 juli 2015 - 06:00

I see a lion to the left, does that cou

Figure 3.2: Misclassification of a post author.

3.4 Network patterns and behaviour

In order to evaluate how network traffic generated by CARONTE compares w.r.t to network traffic generated by humans (i.e. legitimate users) and state-of-the-art crawlers, we performed an experiment employing the Amazon Mechanical Turk platform. This enables us to compare CARONTE against both ‘undesirable’ and ‘desirable’ traffic from the perspective of the criminal forum administrator.

3.4.1 Experiment methodology

To evaluate `CARONTE`'s network behaviour, we perform a set of experiments to compare the network traffic patters generated by `CARONTE` against those of state-of-art crawlers and humans. We parse the HTTP logs produced by IIS 10 server with a Python script which identifies the single sessions and analyzes different features of the traffic.

Human navigation experiment

For our experiment, we require a sufficiently wide amount of human sessions to browse the different scraped forums, enabling us to compare robots with a reasonably true portrait of web surfers' behavior. To generate human traffic towards our forums, we rely on Amazon Mechanical Turk (MTurk). From the literature review, we identify three main experimental variables characterizing the habits of a regular user on the Internet:

- **Var1**: The interest raised in the reader by the content may lead him to read carefully all the content of a certain thread or not, resulting in skimming and moving quickly to a next resource [17, 22];
- **Var2**: The desire of privacy of the user, which may be high or low, resulting in the adoption of solutions that prevent JavaScript to be executed or not to avoid fingerprinting techniques [1, 7, 36];
- **Var3**: The propensity of an user to open several resources in parallel before actually browsing them or opening one per time, reading their content first before moving to a next resource [18].

To control for possible interdependencies between these dimensions, we create a 2^{3-1} *fractional factorial experimental design*, that allows us to reduce

#	Exp. variable	Treatment A	Treatment B
Var1	The reader is interested in the content or skims a few posts	Read all the content inside of the thread	Skim thread or read first post
Var2	The user enables or disables JavaScript on Tor Browser	Enabled	Disabled
Var3	Opening resources in parallel or sequentially	Sequential	Parallel

Table 3.3: Experimental features and treatments

	Exp1		Exp2		Exp3		Exp4	
	A	B	A	B	A	B	A	B
Var1	-	+	-	+	+	-	+	-
Var2	+	-	-	+	+	-	-	+
Var3	+	-	-	+	-	+	+	-

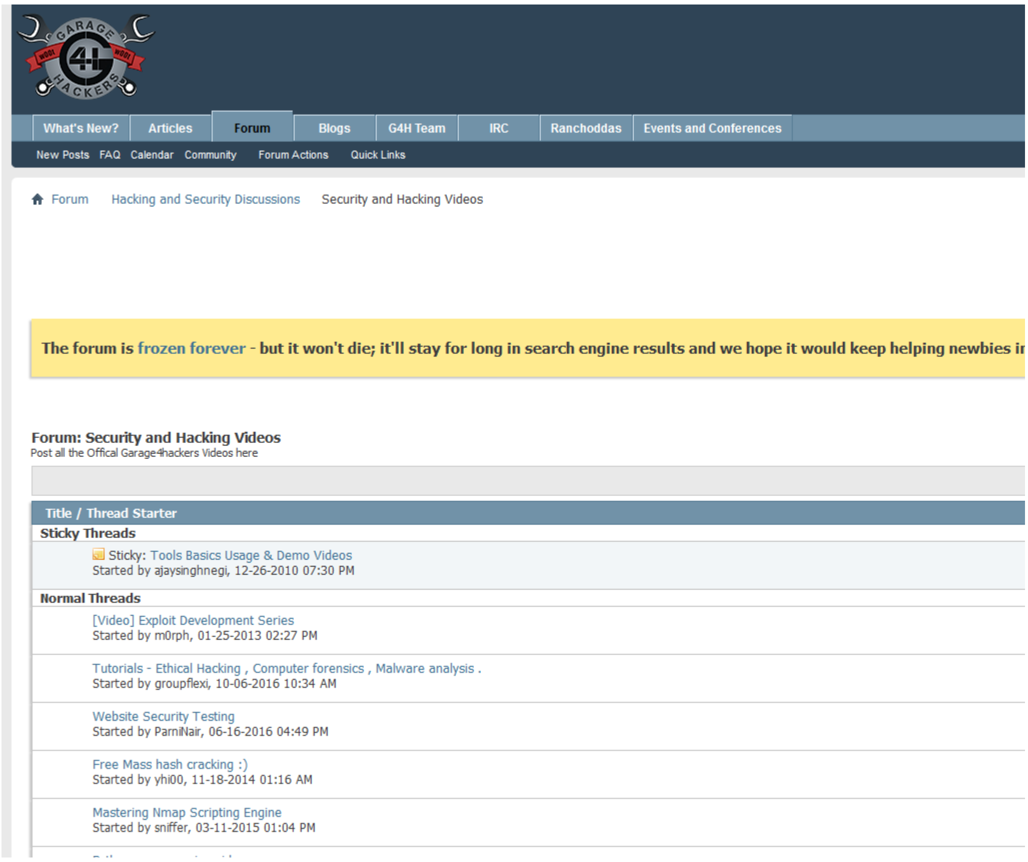
Table 3.4: Treatment combination and experiments.

the number of experimental conditions from eight to four [13], where the eight are resulting from a *fully factorial* experimental design. The subset of the possible experiments is chosen to exploit the sparsity-of-effects principle and to gather the information we need from comparing each experiment to the other, avoiding redundancy on the results from their comparison. The experimental treatments are reported in Table 3.3, and the experiment design in Table 3.4.

3.4.2 Experimental design and setup

An overview of the experimental setup is shown in Figure 3.5. The setup implementation has been carried out in three stages:

The selected web forums (ref. Table 3.1) are hosted on an IIS web server (vers. 10) where access logging is enabled. We prepare an Amazon Mechanial Turk task reflecting the experimental design (ref. Table 3.4). The task includes eight questions based on the content of the forum webpages (two multiple-choice questions per forum).



The forum is frozen forever - but it won't die; it'll stay for long in search engine results and we hope it would keep helping newbies ir

Forum: Security and Hacking Videos
Post all the Offical Garage4hackers Videos here

Title / Thread Starter

Sticky Threads

Sticky: Tools Basics Usage & Demo Videos
Started by ajaysinghnegi, 12-26-2010 07:30 PM

Normal Threads

[Video] Exploit Development Series
Started by m0rph, 01-25-2013 02:27 PM

Tutorials - Ethical Hacking , Computer forensics , Malware analysis .
Started by groupflexi, 10-06-2016 10:34 AM

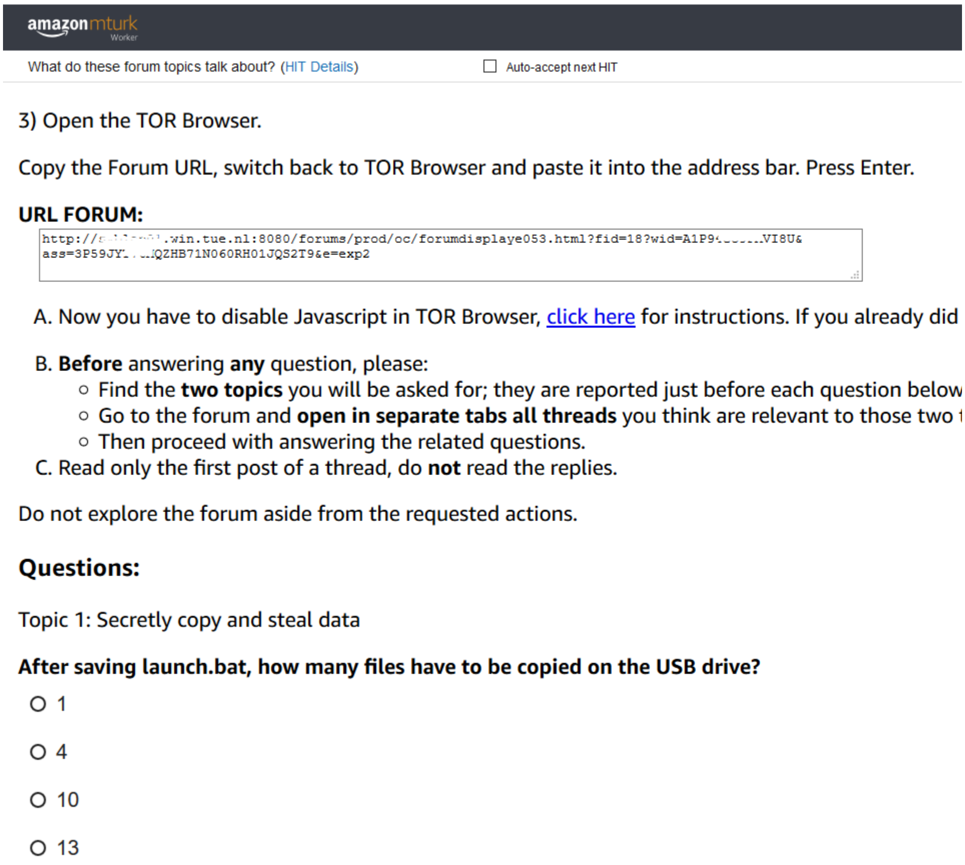
Website Security Testing
Started by ParniNair, 06-16-2016 04:49 PM

Free Mass hash cracking :)
Started by yhi00, 11-18-2014 01:16 AM

Mastering Nmap Scripting Engine
Started by sniffer, 03-11-2015 01:04 PM

Figure 3.3: Forum section screenshot (g4h)

The task included detailed step-by-step instructions that respondents had to follow. Such instructions serve the purpose of enforcing the treatment in



What do these forum topics talk about? (HIT Details)

☐ Auto-accept next HIT

3) Open the TOR Browser.

Copy the Forum URL, switch back to TOR Browser and paste it into the address bar. Press Enter.

URL FORUM:

```
http://[redacted].win.tue.nl:8080/forums/prod/oc/forumdisplaye053.html?fid=18?wid=A1P9[redacted]VI8U&
ass=3P59JY[redacted]QZHB71N060RH01JQS2T9&e=exp2
```

A. Now you have to disable Javascript in TOR Browser, click here for instructions. If you already did

B. **Before** answering **any** question, please:
- Find the **two topics** you will be asked for; they are reported just before each question below
- Go to the forum and **open in separate tabs all threads** you think are relevant to those two t
- Then proceed with answering the related questions.

C. Read only the first post of a thread, do **not** read the replies.

Do not explore the forum aside from the requested actions.

Questions:

Topic 1: Secretly copy and steal data

After saving launch.bat, how many files have to be copied on the USB drive?

○ 1

○ 4

○ 10

○ 13

Figure 3.4: Survey example for **Var2**

the experiment; for example, *Exp3* requires users to read all content of a thread (**Var1, A**), have Javascript enabled (**Var2, A**), and open forum tabs in parallel (**Var3, B**):

> *[...] open in separate tabs all threads you think are relevant to those two topics (**Var3, B**).*
>
> *While reading the forum threads, please also skim through to at least the second thread page (**Var1, A**), if present, and even if you already found the answer.*

Notice that, as JS is enabled by default in TOR browser, there is no instruction for **Var2, A**. When a respondent accesses the task on AMT, he or she is assigned randomly to an experimental condition. Further, each instance of the experiment randomizes the forum order to minimize cross-over effects. The reasoning behind this design is that, on one hand, we need the respondents to navigate on our server to generate traces of their behavior, on the other, the respondents need to visit the websites to find the answers, thus generating useful traffic. In the end, we also need to validate their work and reward the respondents that have complied with the assignment, i.e. at least half of the answers are correct. The survey was written with the help of Amazon's MTurk libraries and published on the same crowd-sourcing platform. From there, users can accept our HIT for an economic reward (this is the incentive mechanism to ensure correct behavior by workers) and afterwards they are asked to install TOR Browser before being forwarded to the forum copies. We also crosscheck their IPs with the public TOR exit nodes list.

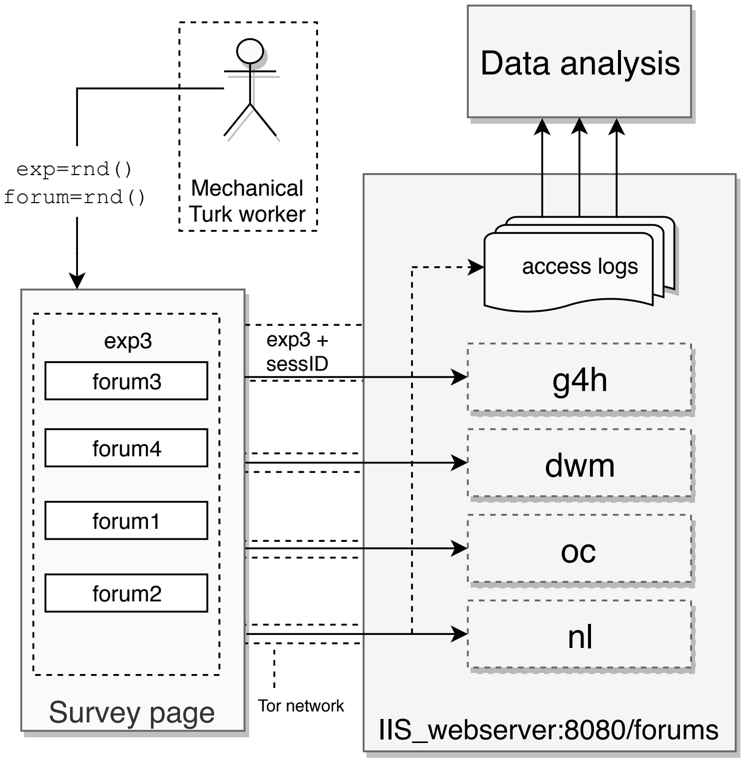
The last step consists of enabling us to collect the generated network requests. To avoid limitations imposed by the TOR circuit refresh mechanism[1] that may change the IP address of users every 10 minutes, we set a cookie on the user's browser with a unique session ID. We use the same strategy to track the experimental condition to which the user has been randomly assigned to at access time.

3.4.3 Obtaining activity data from logs

To measure user navigation patterns, we employ the following metrics:

- To understand the interest and to monitor how humans behave while browsing content, we will measure the time that elapses between requests;

[1] `MaxCircuitDirtiness` - https://www.torproject.org/docs/tor-manual-dev.html.en



The forums are deployed on an internal system at the University. Resources are accessed by industry standard automated tools (scrapers), `CARONTE`, and MTurks. All tools access the local resources through the TOR network. Each MTurk is randomly assigned to an experiment setup with different conditions (see Table 3.4). Internal network logs allow us to backtrack user requests to specific experimental setups.

Figure 3.5: Experimental setup

- We can check if the MTurk has disabled JavaScript by not finding any request to download `.js` files for his session;

- We can detect if several threads have been opened in parallel by watching if the subsequent page opened is the second page of the same thread or not.

Similarity to browser traffic

To measure specific network requests generated by the crawlers, the legitimate TOR browser, and CARONTE, we do the following:

- Compute the media/text ratio of the requests. This ratio is expressed as the number of multimedia content requests against the number of clicked links;
- Check the active use of cache;
- Check the presence of the referrer field in requests;
- Check whether the requester has downloaded any style.

The lack of these features in the logs may represent a detectable crawler session, therefore they are significant aspects to be monitored as well.

3.4.4 Results

Figure 3.6 reports the network analysis for CARONTE compared to the state of the art tools and the MTurks. From the comparisons, emerges that the time elapsed between two different requests[2] produced by humans is comparable to CARONTE's and HTTrack, while the others perform more aggressively. For what concerns the media/text ratio of the sessions, CARONTE together with grab-site, perform quite close to humans. Finally, we've compared the number of requests issued per thread: CARONTE and grab-site perform again better when compared to humans than the other two tools, but their behavior slightly differs from MTurks. Overall, we observe that CARONTE network trace is consistently very similar to human-generated network traffic, whereas other

[2] A **request** refers to all the calls to a page of a thread, without considering all the linked content downloaded.

tools are clearly different over one or more dimensions. This is probably caused to the fact that humans may have skipped some pages in the threads. In fact, in multiple cases, the downloaded forums have a plenty of useless replies to threads to allow them to see some hidden content inside of it, resulting in a decrease of the interest from the reader and leading him to skip the following pages. Instead, the difference between `CARONTE` and `grab-site` and the other two tools is caused to the fact that they follow also non relevant links, such as content re-displacement in the page. In particular, this last behavior represent a well-known traffic feature of a crawler.

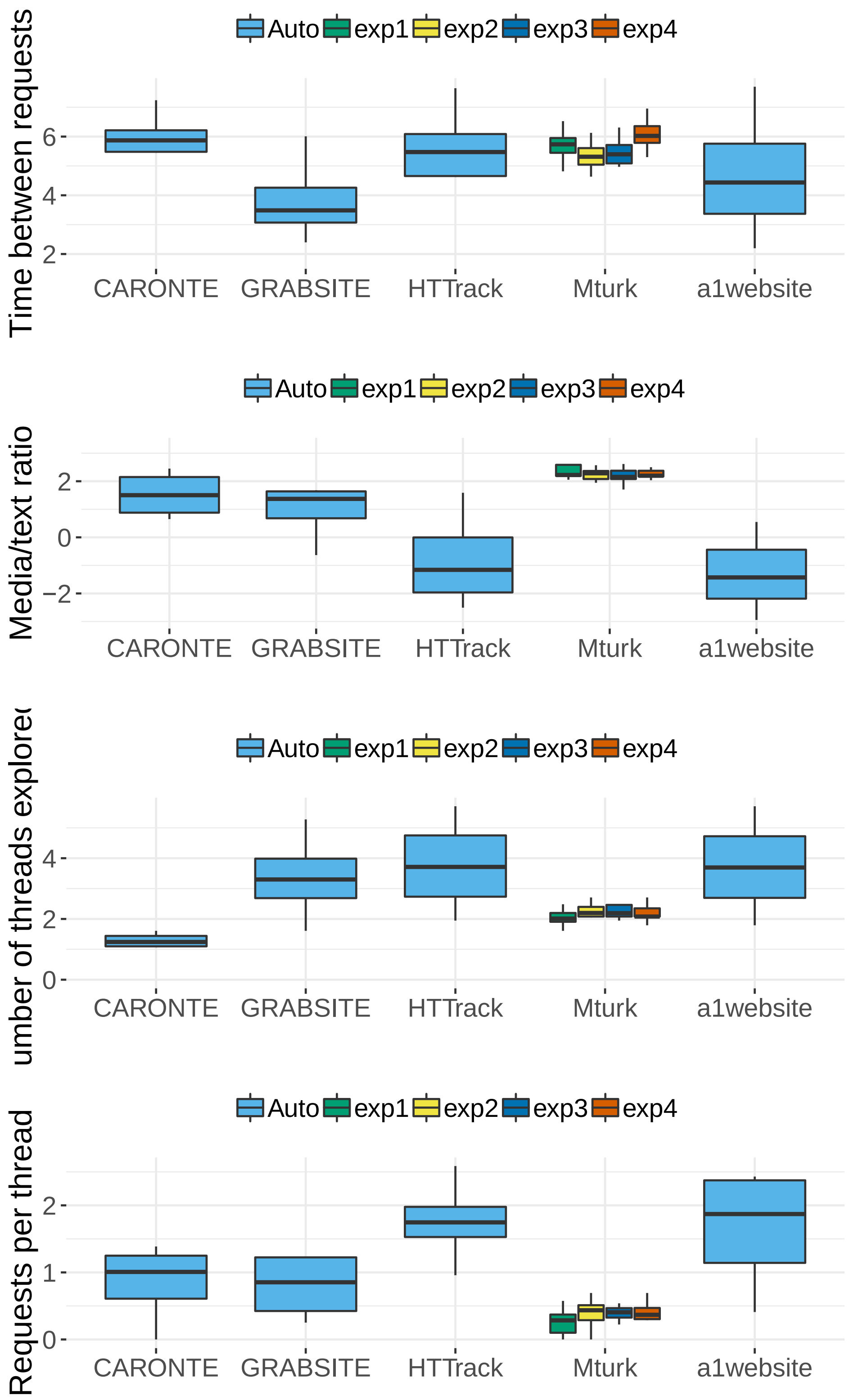


Figure 3.6: Evaluation of `CARONTE` against state-of-art tools and MTurks

Chapter 4

Discussion

CARONTE's training module proved effective in flexibly learning diverse forum structures. Differently from ML-based systems, the adopted semi-automated procedure allows the tool to reliably identify relevant structures in the DOM page, while avoiding entirely the need to collect massive amounts of preexistent data (for the training and validation) that might jeopardize the researcher activity. Whereas this does come at the price of additional human-sourced work w.r.t. fully-automated procedures, CARONTE is meant to be employed over (the few) highly-prominent underground communities where the threat model CARONTE addresses is realistic. The presented proof-of-concept has been tested over four diverse forum structures, and can be expanded in future work across others as well as beyond the 'forum' domain (e.g. e-commerce criminal websites).

From the network analysis it emerges that CARONTE reproduces coherently the three investigated features when compared to humans and performs better, on average, than the other tools. Our tool produces the multimedia traffic of a regular human actor, together with grab-site, while the other two tools diverge from this behavior; we suspect that this is to be traced back to some optimization mechanisms which avoid to reissue requests for the same resource, without even issuing an HEAD HTTP request. A regular browser instead will always reissue the request while loading another page, if not explicitly instructed by a server-side caching policy. Nonetheless, we have found no confirmation in the documentation of these tools. With regards to the number of requests generated per thread, there's a noticeable difference when compared to humans. This is probably caused to the fact that humans may have skipped some pages in the threads. In fact, in multiple cases, the downloaded forums have plenty of useless replies to threads to allow them to see some hidden content inside of it, resulting in a decrease of the interest from the reader and leading them to skip the following pages. Instead, the difference between `CARONTE` and `grab-site` and the other two tools is caused by the fact that they follow also non relevant links, such as content re-displacement in the page. In particular, this last behavior represent a well-known traffic feature of a crawler. To improve this result, it could be possible to instruct our tool to leave threads where content gets redundant and extremely short.

As mentioned in the *network patterns and behavior* subsection, we have monitored some extra features that may represent a red-flag in crawler detection. Nonetheless, they're not part of the experiment since are enforced conditions (like the filling of the referral field) by design of our tool. Results are shown in table 4.1. As expected, `CARONTE` explores threads one at a time, sequentially, while other crawlers tend to open multiple resources in parallel. `A1 Website Download` has never filled the referrer URL in the HTTP requests, highlighting the fact that this request has not been sent from a browser. In the last analysis, for these monitored aspects, we can say that `HTTrack` performs better than the others in terms of browser features exhibited.

Tool	JS	Styles	Cache	Seq. vs Par.	Referrals
CARONTE	✗	✓	✓	Sequential	✓
grab-site	✓	✓	✗	Parallel	✓
HTTrack	✓	✓	✓	Parallel	✓
A1Website	✓	✓	✗	Parallel	✗

Table 4.1: Extra features monitored.

Conclusions and Future Work

Automated tools that gather data in a stealthy way from high-profile forums are a growing need for researchers and LE alike, due to the increased relevance of these communities for society at large. CARONTE is a proof-of-concept tool aimed at creating a baseline for the mitigation of adversarial monitoring capabilities in these communities against researchers. Future work aims at tuning and testing CARONTE capabilities of collecting usably large amount of data, performance testing, and extension of capabilities (e.g. CAPTCHA solving, non-forum communities).

Credits

E' la seconda volta che mi ritrovo a meno di ventiquattr'ore dalla consegna della tesi con questa parte vuota, indubbiamente la più bella, che richiede sempre un po' di sforzo per essere scritta. Non vorrei dimenticare nessuno: di gente ce tanta e più di prima, e non vogliatemene se ciò accade. Al massimo, ci si ritrova stasera al Cucchiaio d'Oro e appariamo i conti. Siete troppi, quindi l'ordine d'apparizione non coincide sempre con quello d'importanza.

Comincerei col ringraziare i miei Genitori, che mi hanno sempre supportato, anche quando ho detto *"a Pasqua non scendo"* e *"vado più lontano di prima"*. La serenità d'animo che hanno potuto trasmettermi è un

Is the second time that I find myself less than twentyfour hours before submitting my final work with this section still empty. For sure, it is the most pleasant to do, but requires some effort to be written. I don't want to forget anyone: I've met a lot of important people during the last two years and don't blame me if I do forget someone. Worst case, tonight we meet at Cucchiaio d'Oro and we solve this issue. You're too many, so the order is not relevant in terms of importance.

I'd start first to thank my Parents, who've always supported me, also when I said "I won't come home at Easter" and "I'm moving further than before". The peace of mind they've gave me is an invaluable gift dono immenso che mi ha permesso di arrivare a questo giorno, soddisfatto del mio percorso e consapevole di averli resi orgogliosi; ringrazio mio Fratello e mia Sorella, coi quali condivido il sogno di una realizzazione personale e le fatiche connesse ad un percorso di studi che presenta insidie tutte sue. A voi, il migliore degli auguri, sia a chi sta remando nella corrente e a chi sta per spiccare il volo.

Ringrazio coloro che ci sono sempre stati: Claudio, Pierpaolo e Giovanni. Intere giornate in aule studio, birre quando si smonta, serate e nottate a parlare di cazzate e preoccupazioni, paste (e cani) in quantità industriale, memetica all'ennesima potenza, dove movimenti facciali minimi e citazioni di video rustici governano il timbro delle nostre conversazioni. Non da ultimo, temo cosa ne sarà di me stasera; le carogne sono ingegnose.

Ringrazio anche chi è entrato solo ad un certo punto del mio percorso: la mia coinquilina Carlotta, amica

that allowed me to arrive to this day, satisfied of my journey and sure I've made them proud; thanks my Brother and my Sister, with whom I share the sparkle of the dream of the personal fulfillment and the endeavours connected to a study path with its snares. Best of luck to you, both to whom is rowing in the stream and to whom is just spreading the wings.

I wish to say thanks also to who has been always there: Claudio, Pierpaolo e Giovanni. Whole days in library, beers when getting out from there, evenings and nights talking about bullshit and concerns, pasta (e cani) like crazy, memetics to its deep essence, where a slight facial movement and citations of rustic videos rule the tone of our conversations. Last but not the least, I'm worried about my life tonight; the bastards are clever.

I'm grateful as well also to who joined my path later: my flatmate Carlotta, friend and confidant, per-

e confidente, persona con la quale ho speso uno degli anni più divertenti di sempre qui a Bologna; il mio (seppur per poco) coinquilino Federico, compagno di zingarate a bordo di una Punto *non* assetto corse, portatore sano di alcolismo ed eterno secondo a Crash Team Racing. Come ha detto Mika Hakkinen in un'intervista, parlando di Michael Schumacher: *"...quando gareggiavo contro di lui, era un piacere vederlo negli specchietti."*

Ringrazio ancora una volta Ivan, una tra le prime persone che ho conosciuto qui ed amico che continuo a portarmi dietro in tutti questi anni. Bevute, giornate a rifiutare la luce del sole davanti ad un computer, un'esperienza in Goliardia e angoscia per gli studi sono le cose che abbiamo condiviso, ma il motivo alla radice di tale amicizia resta ancora un mistero. Ma ho smesso di chiedermelo.

A coloro che mi hanno permesso di riscoprire interessi che credevo perduti: Jacopo, Natale e Daniele,

son with whom I've spent one of the best years of all times there in Bologna; my flatmate (also if just for a while) Federico, the gipsy things companion with the Punto without racing pack, carrier of alcoholism and eternal runner-up at Crash Team Racing. As Mika Hakkinen said in an interview about Michael Schumacher: "...when I was racing against him, it was a pleasure to have him in the mirror."

I thank again Ivan, one among the first people I've met here in Bologna and friend that I still bring back with me, also after all these years. Drinking, days refusing to see the light spent in front of a computer, an experience in Goliardia and study distress are the things that we've shared, nonetheless the reason why of this friendship is still a mistery. But I've stopped questioning myself.

To the ones that allowd me to rediscover passions that I thought I lost: Jacopo, Natale and Daniele, coi quali ho avuto il piacere di spendere giornate e nottate al computer per gustare la soddisfazione di una sudatissima flag, andare in giro a fare *cose divertenti*, insultarci su pensieri politici opinabili e a flammare il feeder di turno in squadra.

Un pensiero va anche alla gente che ho conosciuto a Cesena. Ad Enrico ed Alessio, terroni espatriati con la voglia di scoprire e vedere posti nuovi, oltre che portatori quasi sani di dialetto e cultura del sud in questa terra romagnola. A proposito di romagnoli: grazie anche ad Alberto, Lisa, Gabo, Cevo e Dodo; grazie per avermi accolto nel vostro gruppo sin da subito e di avermi istruito su crescioni e piadine.

Un ringraziamento speciale è a tutti coloro che ho avuto il piacere e l'onore di incontrare durante questo Erasmus ad Eindhoven. In particolare, desidero ringraziare Abhishek, per la sua curiosità infinita, le conversazioni ed i confronti filosofici sca-

with whom I had the pleasure to spend whole days and nights in front of a computer to reaching the satisfaction of a hard-earned flag, going around making funny things, insulting each other for questionable political beliefs and flaming the feeder in the team.

A thought flies also to the people I've met in Cesena. To Enrico and Alessio, expats with the desire to discover and see new places, besides being carriers of dialect and southern culture in this land. Talking about Romagna: thanks to Alberto, Lisa, Gabo, Cevo and Dodo; thanks for having me in your group from the beginning and for introducing me to crescioni and piadine.

Particular thanks fly to everyone I've had the pleasure and the honor to meed during this Erasmus in Eindhoven. In particular, I wish to say thanks to Abhishek, for his limitless curiosity, the philosophical conversations happened, both before and af-

turiti sia prima che dopo la terza birra e la sua mancanza di capacità nel mangiare spaghetti. E ancora, Simone, Francesco e Gaia, trio di terroni (Simò, lo sei pure tu, *deal with it*) coi quali ho speso giornate di studio, cene improbabili, litri di Bavaria (che Dio ce ne salvi) e videogames nelle studycells che vanno contro le deadlines degli assignments.

Ed ancora: Elsa e Kitti, due coinquiline che hanno reso una anonima student house molto più di un semplice dormitorio, ma un posto franco dove spendere piacevoli serate in compagnia; siete nell'Albo d'Oro dei miei coinquilini.

Infine ringrazio il prof. D'Angelo, il quale con le sue lezioni e dibattiti, ha contribuito sensibilmente al riappassionarmi al mondo della Sicurezza Informatica. Un sentito ringraziamento va anche al Prof. Prandini, che mi ha permesso di accedere a questa esperienza internazionale incredibile, agli sforzi fatti per fondare l'ULISSE Lab assieme al Rev-

ter the third beer, and his lack of skills in eating spaghetti. And going on, Simone, Francesco e Gaia, trio of terroni (Simò, you're a terrone as well, deal with it) with whom I've spent study days, last minute dinners, liters of Bavaria (God forbid) and playing videogames in studycells against the deadlines of assignments.

And still: Elsa and Kitti, two amazing flatmates that transformed a simple dorm in a safe place where to spend pleasant evenings in a good company; you are in the Hall of Fame of my flatmates.

Finally, I'd like to say thanks to Prof. D'Angelo that, with his lessons and forums, has significantly contributed to enlight again my passion on cybersecurity. I would like to express sincere thanks to Prof. Prandini, who gave me the opportunity to join this awesome international experience, to his efforts to give birth to the ULISSE Lab with the help of erendo Melis (bella Andrè!) e alla sua grande disponibilità; stessa gratitudine va al Dott. Luca Allodi, il quale mi ha guidato e supportato durante questo periodo di preparazione di tesi.

Grazie di cuore a tutti per esserci stati, per aver contribuito a modo vostro a rendere questi anni indimenticabili, con l'augurio che possiate continuare ad esserci e di realizzare i vostri sogni. *Ad Maiora!*

Reverend Melis (yo Andrè!) and to his great willingness; same thoughts go to Dott. Luca Allodi, whom guided and supported me during this thesis.

To you all, thanks for having contributed in your way to make these years memorable. I wish that you could still be alongside me in the future and to fulfill your dreams. Ad Maiora!

Bibliography

[1] Firefox zero-day exploit used by fbi to shutdown child porn on tor network hosting; tor mail compromised, Aug 2013.

[2] Elk cloner, Jun 2018.

[3] Top 10 captcha solving services compared, 2018.

[4] Zerodium discloses flaw that allows code execution in tor browser, Sep 2018.

[5] Blacktds: an infrastructure for massive scale malware and phishing distribution on demand. forums linked in the page: https://blacktds.com/.

[6] Abbasi, A., Li, W., Benjamin, V., Hu, S., and Chen, H. Descriptive analytics: Examining expert hackers in web forums. In *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint* (2014), IEEE, pp. 56–63.

[7] Allodi, L. Economic factors of vulnerability trade and exploitation. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), ACM, pp. 1483–1499.

[8] Allodi, L., Corradin, M., and Massacci, F. Then and now: On the maturity of the cybercrime markets. *IEEE Transactions on Emerging Topics in Computing* (2015).

[9] Baeza-Yates, R., Ribeiro, B. d. A. N., et al. *Modern information retrieval.* New York: ACM Press; Harlow, England: Addison-Wesley,, 2011.

[10] Bai, Q., Xiong, G., Zhao, Y., and He, L. Analysis and detection of bogus behavior in web crawler measurement. *Procedia Computer Science 31* (2014), 1084–1091.

[11] Balla, A., Stassopoulou, A., and Dikaiakos, M. D. Real-time web crawler detection. In *Telecommunications (ICT), 2011 18th International Conference on* (2011), IEEE, pp. 428–432.

[12] Benjamin, V., Li, W., Holt, T., and Chen, H. Exploring threats and vulnerabilities in hacker web: Forums, irc and carding shops. In *Intelligence and Security Informatics (ISI), 2015 IEEE International Conference on* (2015), IEEE, pp. 85–90.

[13] Box, G. E., and Hunter, J. S. The 2 k—p fractional factorial designs. *Technometrics 3*, 3 (1961), 311–351.

[14] Cai, R., Yang, J.-M., Lai, W., Wang, Y., and Zhang, L. irobot: An intelligent crawler for web forums. In *Proceedings of the 17th international conference on World Wide Web* (2008), ACM, pp. 447–456.

[15] Doran, D., and Gokhale, S. S. Web robot detection techniques: overview and limitations. *Data Mining and Knowledge Discovery 22*, 1-2 (2011), 183–210.

[16] Doran, D., Morillo, K., and Gokhale, S. S. A comparison of web robot and human requests. In *Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining* (2013), ACM, pp. 1374–1380.

[17] Dupret, G., and Liao, C. A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine. In *Proceedings of the third ACM international conference on Web search and data mining* (2010), ACM, pp. 181–190.

[18] Dupret, G. E., and Piwowarski, B. A user browsing model to predict search engine click data from past observations. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (2008), ACM, pp. 331–338.

[19] Fallmann, H., Wondracek, G., and Platzer, C. Covertly probing underground economy marketplaces. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (2010), Springer, pp. 101–110.

[20] Ford, R., and Ray, H. Googling for gold: Web crawlers, hacking and defense explained. *Network Security 2004*, 1 (2004), 10–13.

[21] Guo, F., Liu, C., and Wang, Y. M. Efficient multiple-click models in web search. In *Proceedings of the second acm international conference on web search and data mining* (2009), ACM, pp. 124–131.

[22] Guo, Q., and Agichtein, E. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *Proceedings of the 21st international conference on World Wide Web* (2012), ACM, pp. 569–578.

[23] Herley, C., and Florencio, D. Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy.

[24] Holt, T. J., Strumsky, D., Smirnova, O., and Kilger, M. Examining the social networks of malware writers and hackers. *International Journal of Cyber Criminology 6*, 1 (2012).

[25] Jacob, G., Kirda, E., Kruegel, C., and Vigna, G. Pubcrawl: Protecting users and businesses from crawlers. In *USENIX Security Symposium* (2012), pp. 507–522.

[26] Jordan, T., and Taylor, P. A sociology of hackers. *The Sociological Review 46*, 4 (1998), 757–780.

[27] Korakakis, M., Magkos, E., and Mylonas, P. Automated captcha solving: An empirical comparison of selected techniques. In *SMAP* (2014), pp. 44–47.

[28] Kwon, S., Kim, Y.-G., and Cha, S. Web robot detection based on pattern-matching technique. *Journal of Information Science 38*, 2 (2012), 118–126.

[29] Kwon, S., Oh, M., Kim, D., Lee, J., Kim, Y.-G., and Cha, S. Web robot detection based on monotonous behavior. *Proceedings of the information science and industrial applications 4* (2012), 43–48.

[30] Lai, Y.-M., Zheng, X., Chow, K., Hui, L. C., and Yiu, S.-M. Automatic online monitoring and data-mining internet forums. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011 Seventh International Conference on* (2011), IEEE, pp. 384–387.

[31] Lim, W.-Y., Raja, V., and Thing, V. L. Generalized and lightweight algorithms for automated web forum content extraction. In *2013 IEEE International Conference on Computational Intelligence and Computing Research* (2013), IEEE, pp. 1–8.

[32] McNair, J. What is the average reading speed and the best rate of reading? *http://ezinearticles.com/?What-is-the-Average-Reading-Speed-and-the-Best-Rate-of-Reading?&id=2298503* (2009).

[33] Meyer, G. R. The social organization of the computer underground. Tech. rep., Northern Illinois Univ De Kalb, 1989.

[34] Nayak, K., Marino, D., Efstathopoulos, P., and Dumitraş, T. Some vulnerabilities are different than others. Springer, 2014, pp. 426–446.

[35] Oerlemans, J.-J., et al. *Investigating cybercrime.* PhD thesis, 2017.

[36] Pastrana, S., Thomas, D. R., Hutchings, A., and Clayton, R. Crimebb: Enabling cybercrime research on underground forums at scale. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web* (2018), International World Wide Web Conferences Steering Committee, pp. 1845–1854.

[37] Portnoff, R. S., Afroz, S., Durrett, G., Kummerfeld, J. K., Berg-Kirkpatrick, T., McCoy, D., Levchenko, K., and Paxson, V. Tools for automated analysis of cybercriminal markets. In *Proceedings of the 26th International Conference on World Wide Web* (2017), International World Wide Web Conferences Steering Committee, pp. 657–666.

[38] Qassrawi, M. T., and Zhang, H. Client honeypots: Approaches and challenges. In *New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on* (2010), IEEE, pp. 19–25.

[39] Sardar, T. H., and Ansari, Z. Detection and confirmation of web robot requests for cleaning the voluminous web log data. In *IMpact of*

E-Technology on US (IMPETUS), 2014 International Conference on the (2014), IEEE, pp. 13–19.

[40] Soska, K., and Christin, N. Measuring the longitudinal evolution of the online anonymous marketplace ecosystem. In *USENIX Security* (2015), vol. 15.

[41] Spafford, E. H. The internet worm incident technical report csd-tr-933.

[42] Stevanovic, D., An, A., and Vlajic, N. Feature evaluation for web crawler detection with data mining techniques. *Expert Systems with Applications 39*, 10 (2012), 8707–8717.

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

[44] Von Ahn, L., Blum, M., Hopper, N. J., and Langford, J. Captcha: Using hard ai problems for security. In *International Conference on the Theory and Applications of Cryptographic Techniques* (2003), Springer, pp. 294–311.

[45] Yip, M., Shadbolt, N., and Webber, C. Why forums? an empirical analysis into the facilitating factors of carding forums.

[46] Zhang, D., Zhang, D., and Liu, X. A novel malicious web crawler detector: Performance and evaluation. *International Journal of Computer Science Issues (IJCSI) 10*, 1 (2013), 121.

[47] Zhang, Y., Fan, Y., Hou, S., Liu, J., Ye, Y., and Bourlai, T. idetector: Automate underground forum analysis based on heterogeneous information network. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2018), IEEE, pp. 1071–1078.

[48] Zhao, Q., Willemsen, M. C., Adomavicius, G., Harper, F. M., and Konstan, J. A. Interpreting user inaction in recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems* (New York, NY, USA, 2018), RecSys '18, ACM, pp. 40–48.