Forensic Science International: Digital Investigation 33 (2020) 300943

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

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

A new web forensic framework for bot crime investigation

Rizwan Ur Rahman [a, b, *], Deepak Singh Tomar [a]

[a] Department of CSE and IT, MANIT, Bhopal, 462003, India
[b] Department of CSE and IT, JUIT, Solan, 173234, India

ARTICLE INFO

Article history:
Received 10 October 2019
Received in revised form
28 February 2020
Accepted 4 March 2020
Available online xxx

Keywords:
Web forensic
Forensic framework
Web bot
Spam bot
Web scrapping
Cyber crime

ABSTRACT

Bots are automated programs that robotically navigate the website, upload the data on servers and scrape the data from websites. According to numerous bot traffic reports nearly fifty percent of the website traffic is coming from automated programs. In recent years we have seen a rise in cyber crimes such as illegal web scraping using automated bots. Facebook filed and won a case against Power.com for illegally scraping the Facebook data. Recently in one of the biggest online ticketing scams, a man was arrested for illegally booking tickets using automated bots. While mitigating cyber crime, web forensic investigators face numerous challenges and issues dealing with bot crimes. Most of the existing research is based on web access logs which contain very basic and limited information.

In this paper, we propose four phase web forensic framework to guide forensic examiners in their expedition to verify if the crime is done using automated bots. In order to evaluate the proposed framework, we applied it to the real web application and experimental case scenario. For this case study, a bot crime scenario is developed in an investigation environment. Subsequently, we present in depth forensic procedures and technical reports for bot crime investigation.

1. Introduction

At the moment, the human society is much dependent on the Internet which is very important source of digital communication. The impact of the Internet on human society is currently making educational, economic, societal, and political changes around the world. For example, social media is giving a platform for the people to have a voice in democratic government to raise issues, and hold politicians accountable. Therefore, the ease of access to the Internet is very important for the development of the civilized society. When we connect to the Internet whether from personnel computer, laptop or smart phone, a web browser is generally required. Web browsers facilitate Internet users to access e-commerce websites, buy online tickets, pay bills, and job searching. It also facilitates Internet users to access social networking sites, educational portals and online blogs. For some of the Internet users, this web traffic is sensitive by nature and for others such as attackers it gives the opportunity to attack users. For example, automated program known as bad bot is a class of web security attack that creates a substantial threat to the security of web application (Jin et al., 2013). Basically, the bot is a script or program which is created to carry out fully automated and repetitive task on websites. The term bot is also used interchangeably with the term software agent as described by Franklin and Graesser in agent based taxonomy (Franklin and Graesser, 1996).

The purpose of the creation of bots could either be good or bad and the characteristics or the behavior categorize the bot to be good or bad (Gilani et al., 2016). The most widely used good bot is web crawler such as spider and its main purpose is to index websites in search engines (Thelwall, 2001). In contrast, bad bots are created to perform a variety of malicious tasks.

Bad bots are the main cause of security attacks on websites. These website attacks can be as small as price scraping and click fraud. On the other hand, it can be as big as stealing credit card information and credential stuffing (Wilbur and Zhu, 2009; Rahman and Tomar, 2018; Wang et al., 2014). In general bad bots attack each protection target such as confidentiality, integrity and availability of web applications. For example, price scarping bot which illegally steals the prices of product targets the confidentiality of e-commerce applications. Form spamming bots which are exploited to automatically book the online tickets target the availability of services.

According to latest bot traffic report, bots traffic comprises about forty percent of the total web traffic (Bad Bot Report (2019) htt, 2019). The distribution of web traffic is shown in Fig. 1. From the figure it is clear that the twenty one percent has come from bad bots in the year 2018.

* Corresponding author. Department of CSE and IT, MANIT, Bhopal, 462003, India.
E-mail addresses: rizwan.rahman12@gmail.com, rizwan.urrehman@juit.ac.in (R.U. Rahman).

https://doi.org/10.1016/j.fsidi.2020.300943
2666-2817/

The initial development of general purpose automated programs or Bots was considered as a legal grey area in which bots were used for automatic form filling, purchasing online tickets and scraping the prices from e-commerce websites. It was remained a grey area until on April 14, 2000 when eBay filed a case against Bidders Edge. The eBay stated that the use of automated web bots without eBay's written authorization on the site is against the will of the company (eBay Inc and Bidder's edge Inc, 2000). In the year 2001, a travel company filed a petition against a competitor who had automatically scraped the prices with bots to facilitate the competitor (United States Court of Ap, 2001).

In the year 2009 Facebook won the first copyright suits against Power.com (Non-Infringing Content Be, 2009). The Power.com combined the different social networking accounts in one application. Facebook filed a case against the company with all kinds of complaints, including copyright infringement, illegal competition and computer fraud and abuse act. In 2012 Andrew Auernheimer was convicted of identity fraud to access a website using automated Bots. He harvested around 114,000 emails of iPad customers using web bot (Convicted AT T hacker, 2012).

IRCTC (Indian Railway Catering and Tourism Corporation) is an Indian Railways website that handles tourism and online ticketing, with around half million bookings every day. It is the world's second busiest online ticketing website. In one of the largest IRCTC ticketing scams, the Indian Railways on May 2, 2018 arrested a man from the Mumbai city (Railways to fine Rs. 2 and 0, 2018). He allegedly exploited automated software to facilitate touts for purchasing IRCTC tickets in just a fraction of a second. With the help of automated software, he earned more than fifty thousand USD in a month.

In the most recent case (February, 2020), the Indian Railway Protection Force (RPF) has arrested sixty persons who exploited automated bots to block railway tickets. Officials have described that banned software for instance, ANMS (Tatkal Ticket Software), and Jaguar will evade the application's login CAPTCHA and OTP while a legitimate human user has to undergo all these steps (Now you can have more cha, 2020).

In the year 2016, the United States Congress has passed the first legislation particularly to target the bad bots known as BOTS (Better Online Ticket Sales) Act (Act of 2016 (2016)., 2016), which forbid the use of automated bots that evade security mechanism such as CAPTCHA on ticket seller web applications. The United Kingdom has passed the similar legislation to target the bad bots known as Digital Economy Act 2017 (A better deal for consume, 2017). The Digital Economy Act protects the Internet users in a numerous ways in digital society. It includes the illegal booking of tickets using automated Bots by making it a criminal offence.

In general, for the analysis under criminal law for illegal bots, the Convention on Cybercrime (CoC) is considered as a first step. It is a worldwide convention under the direction of the Europe Council and all the countries of European Union and the United States are party to the CoC (Picotti and Salvadori, 2008). For the analysis of bot crime, the Article 2 of the convention is very important. The Article 2 of the convention deals with three specific issues related to bot: (1) what to access to? (2) breach in security mechanism and (3) when does a bot enter without permission?

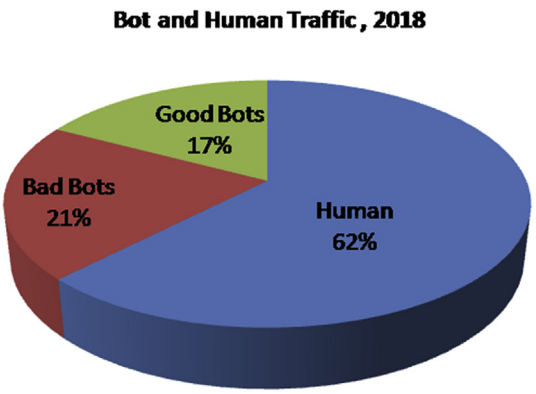

Fig. 1. Web traffic by type (2019

Although web bots offer huge benefits to Information Technology, for example search engines can never be achieved without the development of automated programs. Despite the benefits of automated programs or bots, it also presents a challenge. How do you prove that the activity happened on websites actually done by human or by automated programs? If the activity was done by an automated program, whether the intention was good or bad?

Unexpectedly, this challenge does not come into view in the courts as an argument and as a result, it is not being topic of research by forensic examiners as part of their web forensic investigation process.

With the increasing number of web applications and the development of bots, this is just a matter of time that a web application forensic analyst is going to face that defense in the court.

Given the lack of study on this subject, it implies that small numbers of web application forensic analysts are considering this during their investigation.

Hence, in this paper, we propose a web forensic framework that guides web forensic analysts in their investigation to answer the following question: “Is the website accessed by bot or by humans? If the website was accessed by bot whether the intention was malicious or benevolent?”

The proposed web forensic framework is composed of four phases: Timing Pattern Analysis, Movement Pattern Analysis, Pressure Pattern Analysis and Error Pattern Analysis. To the best of our knowledge this is the first study on web forensic investigation of bot crimes which includes the forensic framework and its real time implementation. The proposed framework is independent of any web technology such as vASP.NET, PHP and JSP.u

The rest of this paper is organized as follows: Section Related work discusses the related work of web forensics research. We introduce proposed approach in Section Web Forensic Framework for Bot Crime Investigation. We evaluate the proposed framework in section applying the proposed framework to real web application. We also present forensic data analysis in subsequent sections. Finally, we conclude the paper in the last section.

2. Background and related work

Web application forensics is a branch of digital forensic which deals with searching and collecting the evidence material found on web servers. The main objective of web application forensics is to identify and trace back cyber attacks on web applications to its originator (Babiker et al., 2018). Other than web application forensics, digital forensics includes fields such as Network Forensics, Cloud Computing Forensics, and Web Services Forensics. Generally, the web application consists of multiple components. These components include web server, database server, firewall, and browser. Web application forensics requires a complete understanding of the web application working process and their components. To identify and trace back a web security attack, we depend on the analysis of the different server logs, database logs, and browser logs (Kyaw et al., 2015).

Numerous digital forensics frameworks have been proposed by a number of authors in academia. A significant contribution in digital forensic investigation has come from the work of Mark M.

Pollitt (Pollitt Mark, 1995). He proposed four distinct phase framework including acquisition, identification, evaluation and evidence. G Palmer proposed forensic framework based on six phases. These six phases are identification, preservation, collection, examination, analysis, presentation, and decision (Palmer, 2001). These are the general forensic investigation frameworks using conventional media and file systems. On the other hand, we need a precise forensic investigation framework with a new concept of flexible data model.

Given the massive amount of web server logs and database logs that need to be analyzed during a web application forensics, numerous tools and techniques have been proposed. These techniques can be categorized into three parts:

- Web server forensics
- Database forensics
- Web browser forensics

Web server forensic is the process of examining web access logs, detection of any modification in web access logs and the recovery of those modifications. Jianhui proposed a standard method for web server forensic based on log content semantic checking (Jianhui, 2008). According to his method, the web forensic framework records the access to the web server logs and merges it with the timestamp and other clues in web server logs. As a result complete information of server access log is formed and this information is represented by XML. Decision tree is applied to analyze the, intrusion behavior and evidence can be extracted. Particularly when an attacker tried to delete his trace, the forensic framework can detect it efficiently.

Kumar et al. (2011) proposed method for detecting any tampering or modifications in access logs. In the first step, they transform a web access log file into image log file with the use of the bit encoding algorithm. They were able to achieve tamper and modification detection capability through self embedding watermark scheme. If any tempering attempt is made on image log file then due to characteristic of fragile watermark, the location of the tampered region can easily be detected.

The work of Jiqiang et al. (Jiqiang et al., 2002) is focused on security of server access logs. They introduced a new system that applies the encryption algorithm to server access logs known as Secure Audit Logs Server (SALS). This method provides a new angle to prevent any tempering attempt to the log files in SALS and identify the modification of logs. This method increases the security of web server, firewall and Intrusion Detection System.

The second important component of web application is the database server. There are a small number of studies about the database forensic frameworks. The early work in database forensics was done by Fowler (2007) who introduced the database forensic analysis of SQL Server 2005. He carried out forensic analysis of the SQL Server 2005 in seven phases: verifying the system, describing the system, collecting the evidence, creating the timeline, analyzing the media, recovering the data, and searching the string. Khanuja et al. (Khanuja and Adane, 2012) introduced six steps framework for database forensic. They include the following steps identification, collection, analysis, validation, interpretation and generation of forensic reports.

Yoon et al. (2016) worked on forensic analysis of NoSQL databases and proposed five phase general forensic framework for NoSQL database. They include the following phases in framework preparation, logical evidence acquisition and preservation, distributed evidence identification, acquisition and preservation, examination and analysis, reporting and presentation. They evaluated their proposed framework by applying it to document-oriented MongoDB database, which is considered as the widely used NoSQL database. For the purpose of evaluation, they build an experimental environment with crime investigation scenario.

The third component of web application is the web browser which works on the client side. At the client side, forensic investigation is generally done to determine if an application user is involved in crime or a victim of the cyber crime.

Most of the work on web browser forensics has been aimed to particular web browsers or to the examination of specific log files. Jones (2003) has given the details of the index.dat file in Internet Explorer. He demonstrated the extraction of deleted activity records from the index.dat file. He simulated an actual crime with the help of Pasco tool. Pereira (2009) demonstrated in detail the modification in the file that contains the browsing history that occurred when Firefox version 2 was updated to Firefox version 3. He proposed a novel approach for searching the deleted browsing history using unallocated fields.

Junghoon et al. (Oh et al., 2011) proposed a new approach for evidence collection and analysis. In this paper, a new tool known as Web Browser Forensic Analyzer is introduced. The developed tool is able to analyze six different types of analysis: Integrated analysis, Timeline analysis, Analysis of search history, Analysis on URL encoding, Analysis of user activity, Recovery of deleted information.

Patil and Meshram (Meshram and Patil, 2018) suggested a methodology to perform the combined forensic investigation of different web browsers of Linux operating system. They included the analysis of URL history, Cache files, and log files for different web browsers of Linux operating system to identify malicious activity.

The work of Nelson et al. (2020) is to discover the digital artifacts from different browsers and their web browsing modes. They were able to recover effectively numerous artifacts including history of the web browser, cookies, and form auto fill data in browsing sessions.

However, there are problems associated with the traditional web forensic frameworks. First, these researches are targeted to a specific Web server such as IIS or a specific web access log.

Second, existing research mainly dependent on parsing the web server logs. Generally, the server log contains the basic information such as IP address, byte sent, error code etc. It does not provide any information on how the website is accessed in terms of timing patterns and movement patterns. In web application forensic, it is essential to extract more important patterns related to web forensics, such as timing patterns, movement patterns and pressure patterns.

Therefore, existing research is not enough capable to use web forensics for Bot crime investigation. In this case, an advanced approach to defeat the limitation of existing research is required. Particularly, the authors believe the following requirements as important:

Timing Pattern Analysis

The access timing pattern of bots is quite different from humans. For example Humans usually wait for an entire web page rendering to find the hyperlinks before they do any further web request. The bots are fast as compared to human users in finding hyperlinks in the returned web page.

Movement pattern analysis

The movement on web pages such as scrolling with the mouse and keyboard gives us the number of clues in web investigation. Generally humans do not scroll the web pages with constant speed. A web page of useful information will slow down the scrolling speed. In contrast, bots usually keep a constant scrolling speed when they scroll a web page.

Pressure pattern analysis

The web access log such as server log and other logs do not give any information of pressure pattern. In this phase left click pressure and right click pressure of mouse are analyzed.

Error pattern analysis

This analysis is used to determine whether humans are making mistakes when typing the characters in web forms. Practically, it can be analyzed by keeping the log of delete and backspace key usage.

3. Web forensic framework for bot crime investigation

In order to answer the following questions:

- Whether the specific web application is requested by Human or by Bot
- If the web application is requested by bot, whether the purpose was malicious or benevolent

We propose a web forensic framework consisting of four phases: 1) Timing Pattern Analysis Phase 2) Movement Pattern Analysis Phase 3) Pressure Pattern Analysis Phase 4) Error Pattern Analysis Phase as shown in Fig. 2.

Phase 1- Timing pattern analysis

The first step of the web forensic framework is to examine the timeline for any clue that will assist the forensic examiner in making an opinion. For example, bots are able to request website at regular period whereas the pattern of human users is not periodic. Human users usually wait for hyperlinks and other webpage contents to appear on browser whereas bots are much faster in finding the contents of the webpage. Since bots are automated programs, so they can hit keys on keyboard at much faster speed than human users. Similarly, when they use mouse their patterns are far different from humans.

We analyze the timing patterns in three different steps Keystroke Timing Patterns, Mouse Timing Patterns and Web Access Timing Patterns. Based on these analyses, it is possible to differentiate between human user and Bot. By keystroke timing patterns we mean characteristic related to the keys that a user hits such as key flight time and key dwell time. By mouse timing patterns we mean characteristic related to the mouse that a user clicks such as left click time and right click time. Web access timing patterns include inter request delay and its derivatives such as standard deviation of inter request delay and entropy of inter request delay. The taxonomy of timing features is shown in Fig. 3 and the description of each timing feature used in forensic framework is as follows.

Dwell Time of Key: It is the time period for which particular key is held down. In other words, it is the time period between the key press and key release for a particular key and this is also known as the hold time (Bours, 2012). For analyzing the dwell time of bot and human we have taken the average dwell time. For example in particular web session if there are 'n' number of key hits then it is the average of dwell time of each key.

Flight Time of Key: It is the time period between releasing a specific key and pressing the subsequent key. It is also known as Up down time. Similar to average dwell time we have taken an average flight time.

Left Click Time of Mouse: It is the time period between the mouse down and the mouse up of left click. Similar to flight time and dwell time; left click time of bot and human is also the average left click time. For instance in particular web session if there are 'n' number of left mouse clicks, then it is the sum of all left mouse click time divided by number of left clicks.

Right Click Time of Mouse: It is the time period between the mouse down and the mouse up of right click.

Inter Request Delay: It is time delay between two consecutive web page requests. Human users typically wait for webpage contents such links and buttons to appear on the browser. On the other hand, bots can click buttons and hyperlinks on a web page without complete rendering of a web page. Hence, the inter request delay of humans are much higher than Bots.

Entropy of Inter Request Delay (IRD): Entropy signifies the uncertainty in the system and reveals that request to a web application is random or directed. A directed and fixed request to the web application will have high entropy value. The entropy of inter request delay is computed using the equation given below (Tan, 2018).

$$\text{Entropy(IRD)} = -\sum_{i=1}^{n} (\text{IRD})_i \log(\text{IRD})_i$$

Standard Deviation of Inter Request Delay: It is the Inter request delay variance of all web requests in a session. For instance, if there are ten web requests in a session, then it is the mean of the squared differences from the average value. Many periodic Bots are capable of making requests at constant period. So obviously their standard deviation will be much lower as compared to humans. Standard Deviation of inter request delay is calculated using the equation given below (Falk, 2018).

Fig. 2. Web forensic framework for bot crime investigation.

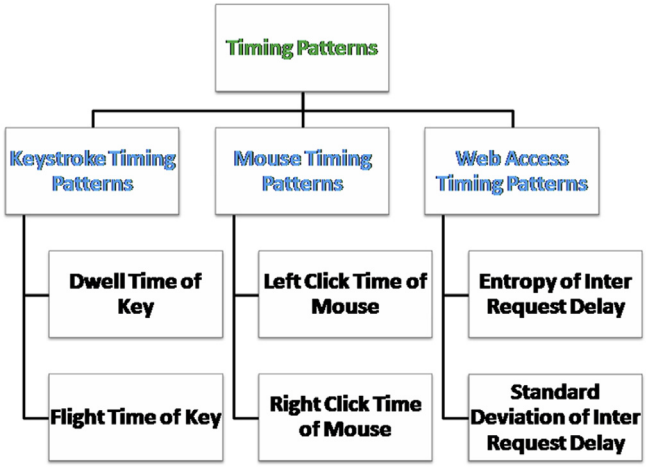

Fig. 3. Taxonomy of timing features.

$$\sigma_{IRD} = \sqrt{\frac{\sum_{i=1}^{n}(IRD - IRD_{mean})}{n-1}}$$

Phase 2- Movement pattern analysis

In a web forensic investigation, it is very important to detect the movement of user on web applications. The movement patterns give the path of motion from one area to another in a web page. Practically, it can be analyzed by determining scrolling and the mouse movement in web pages. We analyze the movement patterns using the following features.

Scrolling Amount in X-direction: It is the distance covered by user or bot when scrolling the web page in X-direction. The unit of this feature is pixels.

Scrolling Amount in Y-direction: It is the distance covered by user or bot when scrolling the web page in Y-direction.

Scrolling Speed in X-direction: It is the speed of user or bot when scrolling the web page in X-direction. It can be obtained by dividing the scrolling amount in X-direction by time taken. The unit of this feature is pixels/millisecond.

Scrolling Speed in Y-direction: It is the speed of user or bot when scrolling the web page in Y-direction. It can be obtained by dividing the scrolling amount in Y-direction by time taken.

Phase 3- Pressure pattern analysis

Practically, we can analyze pressure patterns in web pages using mouse click pressure and key hit pressure. Mouse click pressure is the amount of pressure applied when a user clicks the mouse. The value of pressure is recorded in a number between zero and one. Zero represents no pressure while one represents maximum pressure. To calculate values of Click Pressure, we have exploited Pressure.js (Pressure.js library (2019, 2019) library. Pressure.js is a JavaScript library for calculating the touch pressure, and click pressure on web applications. We analyze the pressure patterns using the following features.

Left Click Pressure: It is the amount of pressure applied when a user clicked the left mouse button.

Right Click Pressure: It is the amount of pressure applied when a user clicked the right mouse button.

Phase 4- Error pattern analysis

When humans interact with web applications and they typewrite in web pages, naturally the make typing errors.

On the contrary bots are automated scripts which actually generate the keystrokes for filling the data in web pages. Practically, we can analyze the error patterns by the usage of delete and backspace key hits.

4. Applying proposed framework to web applications

In order to evaluate the proposed web forensic framework we have applied it to the real web application and build the comprehensive web forensic investigation system. To validate the proposed web forensic framework, we developed a realistic environment and an experimental scenario. Using these cases, we give a detailed description of the web forensic investigation procedures for the real institutional web application.

The experimental case scenario is a web forensic case of an illegal use of web applications using automated bots. The web forensic investigator will be able to examine following information:

- Whether the web application is illegally scraped by automated bots (Price scraping and Email harvesting).
- Whether the comments are posted by bots on a web application (Form spamming and Comment posting).
- Whether the data is submitted by bots to web server (Automated online ticket booking and automatic account creation).

For the purpose of gathering the web forensic data, we exploited two different forensic data sources. The first forensic data source is real institutional web application log and the second is an experimental case scenario of different bot attacks.

In the first phase, we took about thirty day's web access log of institutional website. To extract the web access patterns such timing, movement, pressure and error patterns, we developed two unique access loggers. The first logger is developed in JavaScript for extracting the pressure, error and movement patterns. The second logger is developed in PHP for extracting the patterns such as inter request delay, Entropy of inter request delay and Standard Deviation of inter request delay.

An important issue that should be encountered during the collection of web forensic data is the privacy and confidentiality of user data. As our forensic framework exploits two access loggers, we need to discuss privacy and confidentiality issues of using the access logger to collect web forensic data. First of all, no human user data such as user name or any other data is collected by access loggers. As the access loggers are developed in JavaScript and PHP, it is obviously restricted by the same origin policy imposed by the web browser, and as a result it is not able to obtain the content of other websites. Therefore, the access logger for collecting the web forensic data does not cause any danger to user privacy.

Secondly, we focus on the confidentiality of web forensic data transmitted over the network. When human users fill the form on the website, we do not record and collect the key values instead we calculate the timing, movement, pressure and error patterns using JavaScript. In our experimental case scenario, the communication between client and server is not encrypted.

So, to avert man-in-the-middle attack from capturing forensic data in plain text and retrieving the human access patterns, the access logger obfuscates all the features using simple JavaScript. This JavaScript obfuscation imposes the confidentiality on web forensic data i.e., human access patterns. On the other hand, if the communication between the client and the server is encrypted with Secure Socket Layer (SSL), then this step is not required.

In the second phase, the experimental case scenario is created for investigating about the bots illegal activities. In the experimental case scenario, virtual machines are configured. Consequently, in installed virtual machine bots and automated scripts are executed. These bots include XRumer, Comment Anywhere, Visual Web Ripper, Integromat, and AutoIt. A brief Description of these Bots is as follows:

XRumer (Xrumer, 2012) is automated software made for spamming online forums and comment sections of web applications. It was created by Botmaster Labs and released in the market as a program for SEO (Search Engine Optimization). This tool is able to register and write posts in the forums which is referred as forum spamming.

Comment Anywhere is basically an automated program which searches for the relevant pages that is similar to the search made by the user as per the user's product description. Its main purpose is to save time and money which the advertisers spend in order to get the response and revenue by advertising the content and getting the positive user response.

Visual web ripper (Haddaway, 2015) is a strong visual tool that is used for web harvesting, web scraping and this is also used for extraction of content from the web. This tool scans all the information that is present in a website, the content structure of the

website as well as the search results. This tool targets the data extracts it and then send the content in structured data in XML files, databases or CSV files.

Integromat (Hughes et al., 2017) is very strong tool that is capable to connect web applications, transfer and transform data. It works automatically and there is no user intervention needed to make this tool work. It gives direct support to the application and services with the help of HTTP/SOAP and JSON modules the user can easily connect to any of the web applications.

AutoIt (Brand and Balvanz, 2005) is a Windows based scripting language that imitates user actions, including mouse movement, mouse clicks and keystroke. It is quite different from the other scripting languages. It can surf the websites by precisely using the mouse and keyboard.

Apart from these standard Bots, a Bot attack scenario is created and executed in the experimental case scenario. The attack scenario is form spamming playback Bot which is a malicious program implemented for automatic account registration. When human users interact with webpage using mouse and keyboard, playback bot records their actions. Subsequently, it masquerades as the human user by playing back the recorded traces on webpage. A playback Bot is generally supported by an attacker who locates a web form and identifies the name of the form fields such as textbox. This human trace is temporarily stored together with the URL of the webpage in an attacker database. The playback bot could then send requests to the webpage without visiting the actual webpage, which causes a playback process to simply submit data on web form. The block diagram of the form spamming attack scenario is shown in Fig. 4.

5. Analysis and visualization of forensic data

Based on the collected forensic data from experiment case scenario of bot attacks, we have analyzed the different patterns of bot and human.

5.1. Analysis and visualization timing patterns

In Fig. 5, key dwell time histogram of bot and human is shown. In a key dwell time histogram, bucket size is 50 ms. The average dwell time of all the requests come from humans is 159 ms and the average dwell time of all the requests come from bots is 63.65 ms. Approximately seventy five percent of the dwell time distribution of humans is less than 200 ms. Alternatively, more than ninety percent of the dwell time distribution of bots is less than 200 ms. Evidently, it proves that human users hit keys at quite slower speed than automated bots.

Fig. 6 shows the distribution of key flight time of bot and human respectively. The average flight time of all the requests come from humans is 516.82 ms and the average flight time of all the requests come from bots is 264.55 ms. From this observance, the average key flight time of human is almost double than the bots. This is because, human users look for key on keyboards before they type and flight time of human users also differs from person to person.

Fig. 7 demonstrates the variation in Entropy of Inter Request Delay of bot and human respectively. The two figures demonstrate the Normalized Entropy values of human are below 0.5 in most of the cases. This proves that the human generally wait for an entire web page rendering before they proceed for any other HTTP request. Therefore, their inter request delay varies depending on the web page. In contrast, the Normalized Entropy values of bots are above 0.4 in most of the cases. This proves that the bots are quick as compared to human in searching the hyperlinks. Additionally, bots are capable to click links without waiting for an entire web page rendering. As a result, their inter request delay does not differ as compared to human. We have also investigated that various bots have the normalized entropy more than 0.9. Such observation is absent in humans, because it is nearly impossible.

The distributions of standard deviation values of inter request delay of bot and human users are shown as box plots in Fig. 8. The given box plots visualize the first, second and third quartiles and median of standard deviation values of inter request delay in milliseconds. There are small numbers of outliers residing below the first quartile in human data distribution. On the whole, a significant distribution among the bots and humans is visible, which is not unexpected for two reasons. First, human behaves quite differently when surfing the web pages as compared to bots. Second, many bots which use periodic timer functions for visiting the web pages so their standard deviations of inter request delay are very low.

In Fig. 9 left click time and right click time histogram of bot and human respectively is shown. In given histogram, bucket size is

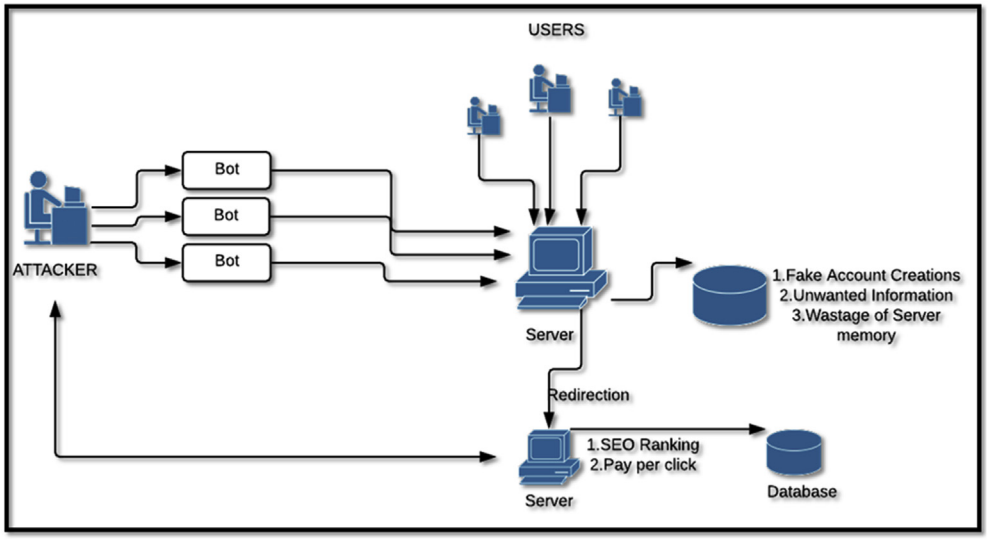

Fig. 4. Block diagram of bot attack scenario.

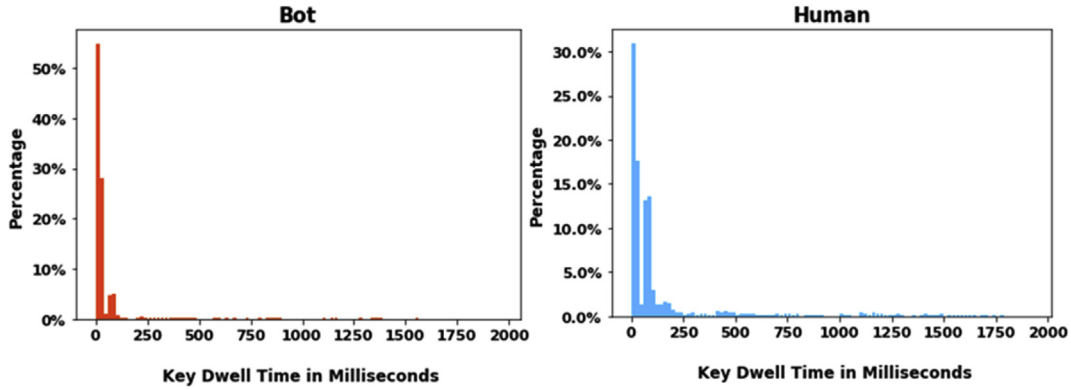

Fig. 5. Key dwell time histogram of bot and human.

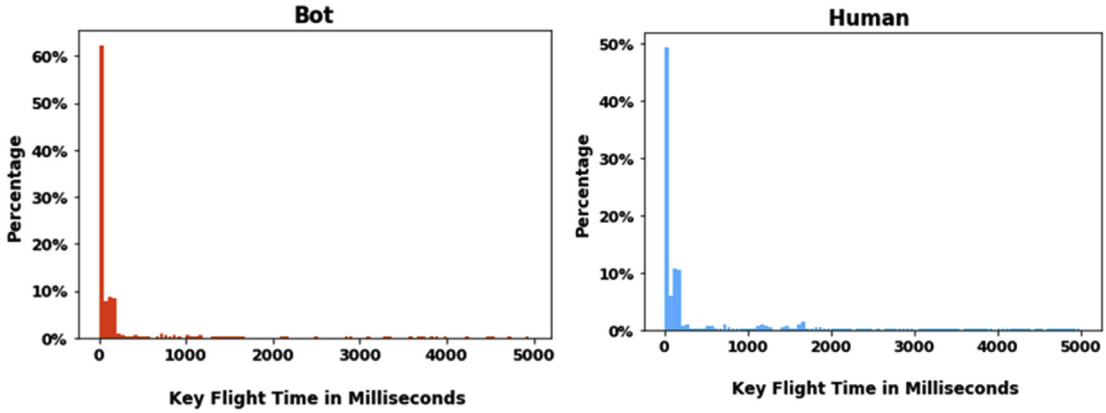

Fig. 6. Key flight time histogram of bot and human.

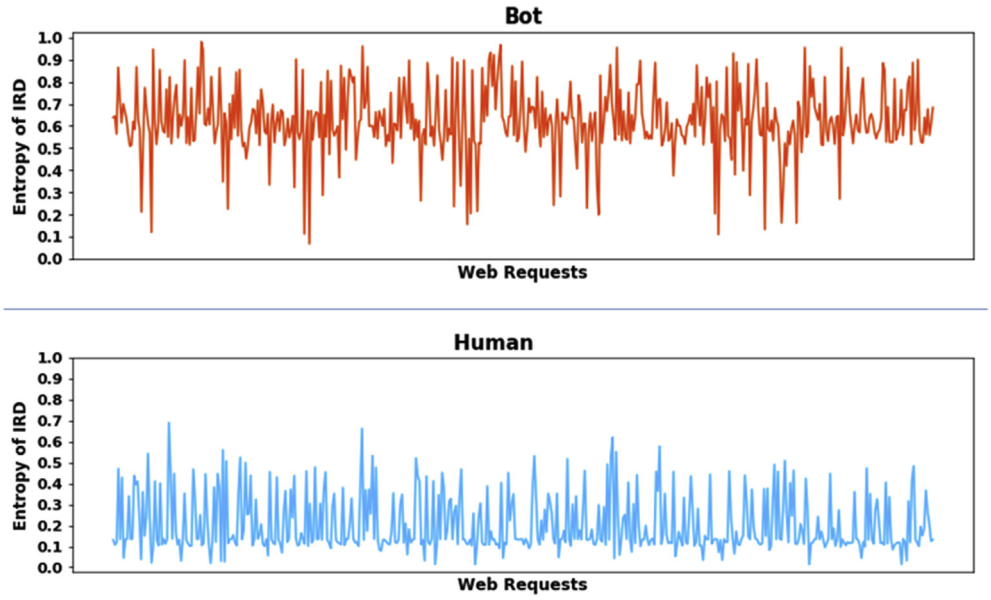

Fig. 7. Entropy of inter request delay of bot and human.

50 ms. The average left click of human users is 187.47 ms and the average left click time of bots is 82.63 ms. On the other hand the average right click of human and bot are higher than the left click time. The obtained average values of human and bot are respectively 216.01 and 119.173 ms. It gives the evidence that bot clicks the mouse much faster than the humans.

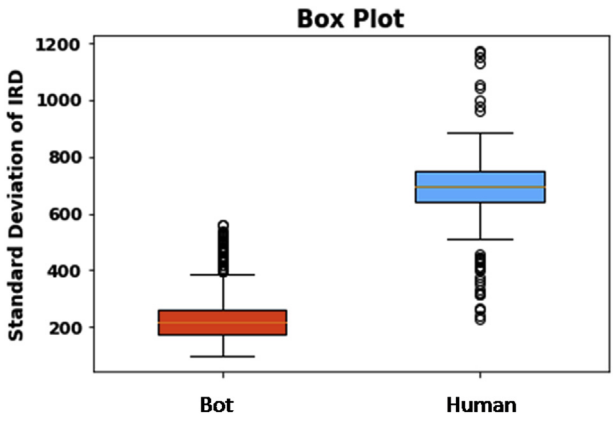

Fig. 8. Box Plot of Standard Deviation values of Inter Request Delay.

5.2. Analysis and visualization of movement patterns

In Fig. 10, Y-direction scrolling speed histogram is depicted with bin size 0.1 pixels per milliseconds. In our case scenario, human scrolled the web pages with lesser speed in comparison to bots. More than seventy percent of the Y-directional scrolling speed is less than 0.78 pixels per millisecond. In contrast bots scrolled the web pages in Y-direction with 0.29 pixels per millisecond to 1.9 pixels per millisecond. Lastly, we have investigated that various bots scrolled the web pages in Y-direction at constant speed.

In the same way X-direction scrolling speed histograms in Fig. 11 is shown. A similar investigation has been achieved with small variations. In case of humans, more than eighty one percent of the scrolling speed in X-direction is less than 0.51 pixels per millisecond. On the contrary bots scrolled the web pages in X-direction with 0.15 pixels per millisecond to 1.75.0 pixels per milliseconds. Some of the bots did not scroll the web pages in X-direction and numerous bots scrolled the web pages at constant speed.

We have investigated the correlation between Y-scrolling amount and time taken to scroll. We have applied linear regression model with these two features.

Fig. 12 shows the correlation between Y-scrolling amount and the time duration of bot and human respectively. From the given scatter plot two key investigations are made. First, human scrolled the web pages in Y-direction haphazardly with random time duration in web pages. In contrast, bots scrolled the web pages in a well-ordered way with a fixed time period. Secondly, Y-scrolling amount of bots varies linearly with time duration. We have applied linear regression model and computed the correlation coefficient between Y-scrolling amount and the time duration. The obtained value of the correlation coefficient of bots is 0.715. The reason for such an excellent value of correlation coefficient is that many periodic bots use constant timers for scrolling the web pages. On the contrary, this observation is absent in human users. In this case, the correlation coefficient between X-scrolling amount and the time duration is 0.254.

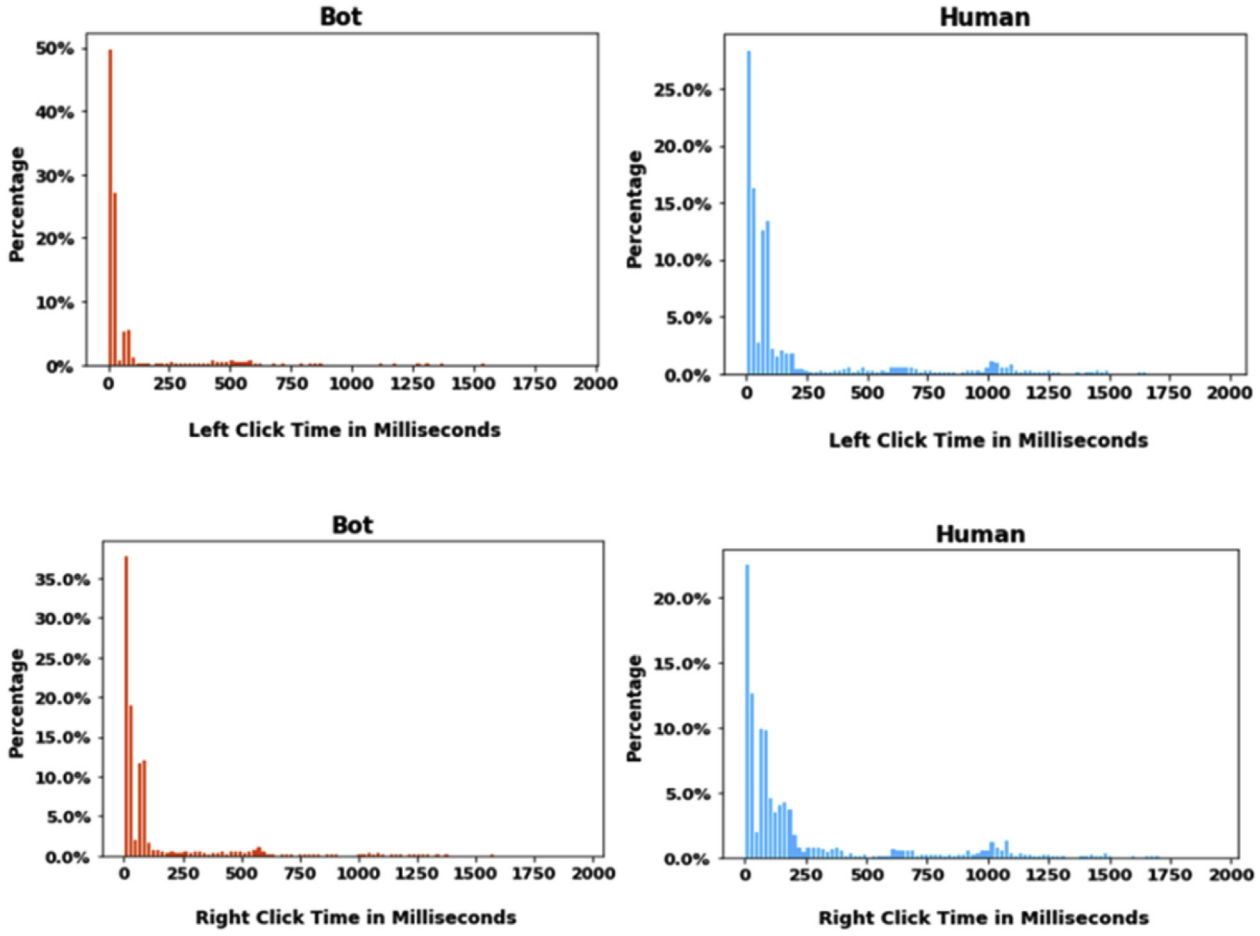

Fig. 9. Left click time and right click time histogram of bot and human.

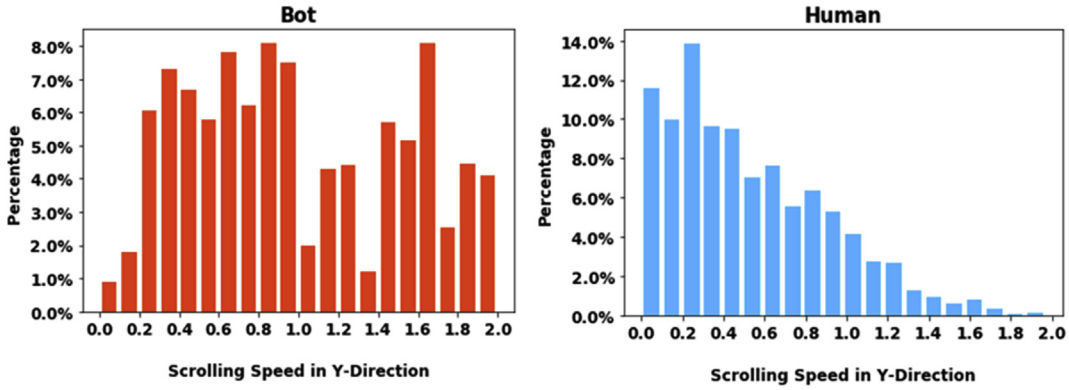

Fig. 10. Y-direction scrolling speed histogram.

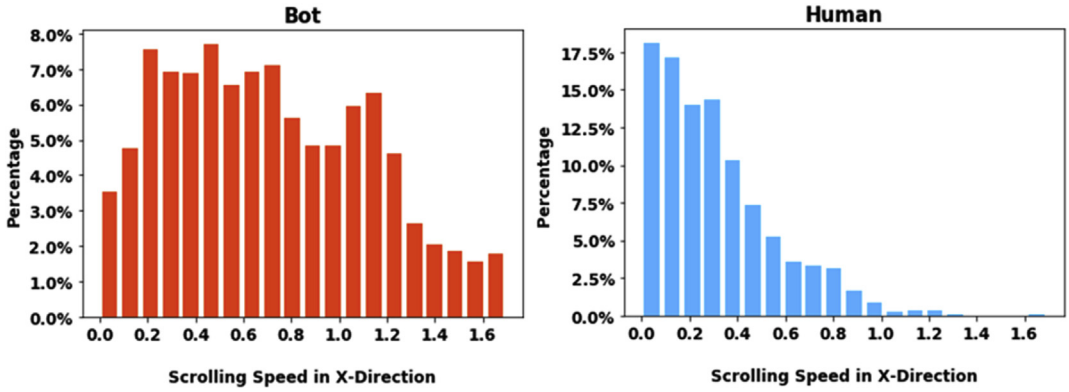

Fig. 11. X-direction scrolling speed histogram.

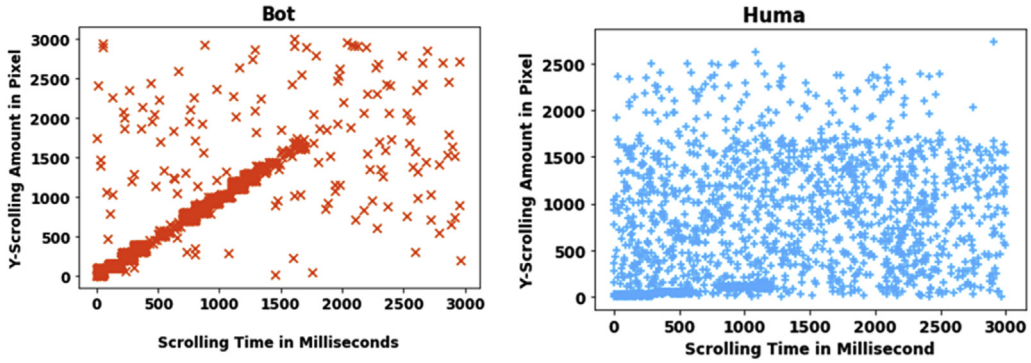

Fig. 12. The orrelation between Y-Directional Scrolling Amount and Scrolling Time.

5.3. Analysis and visualization pressure patterns

In Fig. 13 Left click pressure histogram shows the distribution of bot and human user respectively. In this figure the histogram bucket size 0.01. A significant investigation can be done from the generated figure. Almost all the human users clicked the left mouse button with pressure less than 0.3 units. Nearly 85% human users clicked the left mouse button with pressure less than 0.1 units and the rest of the 15% users between 0.1 and 0.3 values. On the contrary, bots actually generated left click pressure from scripts without using the genuine mouse device. Bots distribution of left click pressure is much different from humans; in general they produced the value of left click left pressure from 0.1 to 0.8.

Similar observations are also made from the right click pressure histogram of bot and human user respectively shown in Fig. 14. The observed difference is all the human users clicked the right button with less than 0.2 values and clicked the left button with less than 0.3 values. More than 99% human users clicked the right button with less than 0.2 values.

5.4. Analysis and visualization error patterns

In this phase, we have analyzed the error patterns in terms of backspace and delete keystrokes. Fig. 15 shows the correlation between the total number of keystrokes and delete keystrokes of bot and human respectively. From the given figure an important investigation is made that in case of human the relation between the total numbers of keystrokes and delete keystrokes is more

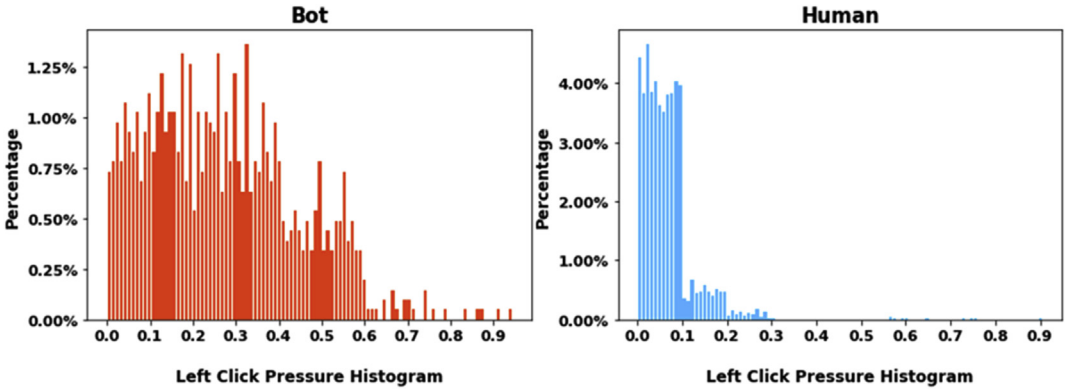

Fig. 13. Left click pressure histogram.

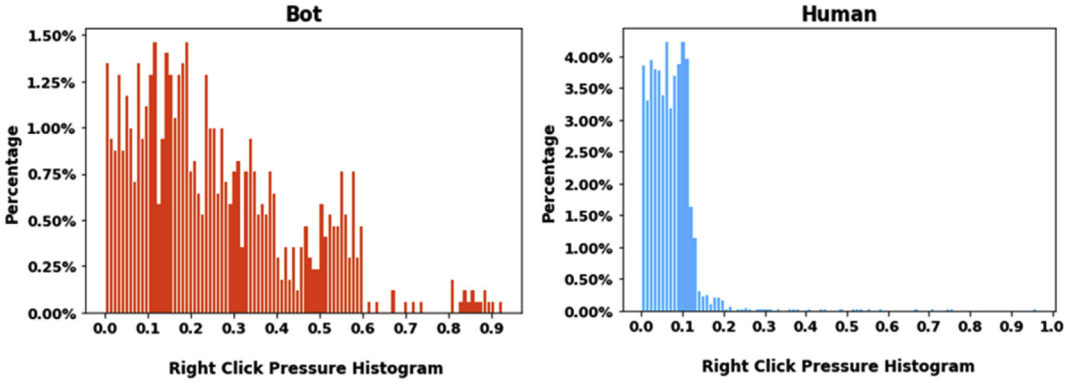

Fig. 14. Right click pressure histogram.

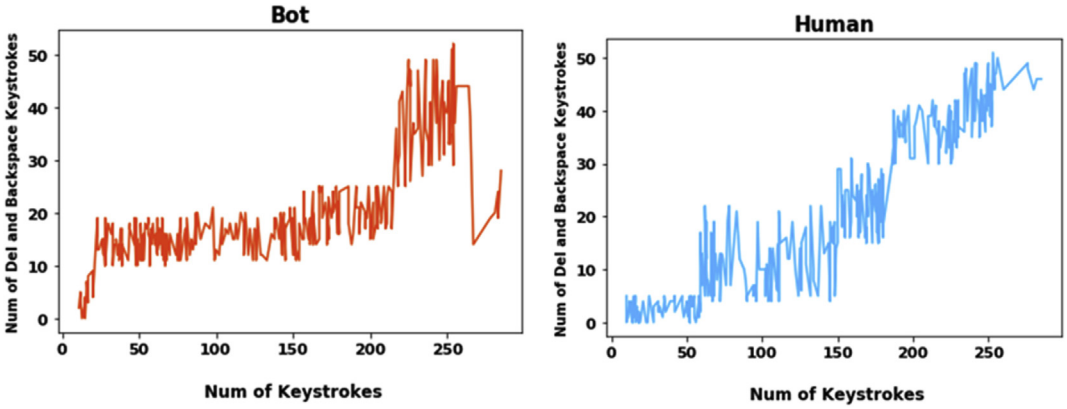

Fig. 15. Correlation between Total number of Keystrokes and Delete Keystrokes.

linear than the bots. We have also applied linear regression model and computed the correlation coefficient between total number of keystrokes and delete keystrokes. The obtained value of the correlation coefficient of human is 0.902. On the other hand the obtained value of the correlation coefficient of bot is 0.683 which is much lower than human value.

We have also analyzed the typing error rate of bot and human. The typing error is defined as the total number of delete and backspace keystrokes divided by the total number of keystrokes. The distribution of the typing error rate of bot and human users is shown as box plots in Fig. 16. The given box plots visualize the first, second and third quartiles and median of a typing error rate. The minimum value of the typing error rate is zero in both the cases. There are small numbers of outliers residing above the third quartile in bot data distribution. On the other hand, there are many outliers residing above the third quartile in human data distribution.

6. Discussion and limitations

Since our prototype of web forensic framework is implemented using JavaScript access logger, it could not generate the web forensic data, if the JavaScript is turned off at client side. However, according to numerous web traffic reports (How many people are missi, 2013; What percentage of browse, 2016), the percentage of Internet users

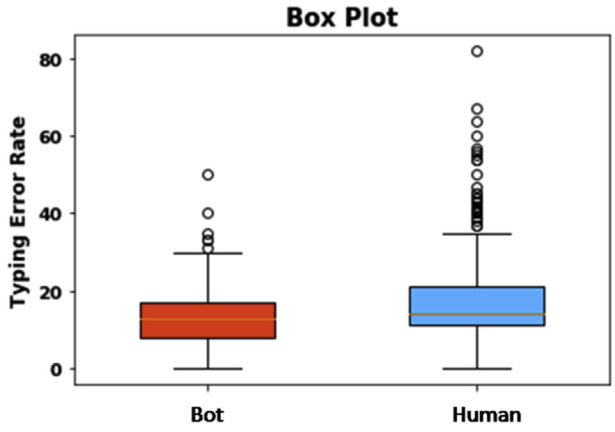

Fig. 16. Typing error rate box plot.

who turned off the JavaScript is less than two percent. Yahoo (How many people are missi, 2013) investigated the web traffic and reported that 0.25%–2% of the Internet users have turned off the JavaScript. Blockmetry (What percentage of browse, 2016) analyzed the web traffic from all devices in the year 2016 and reported that 0.2% of page views from across the globe had JavaScript turned off. Additionally, web traffic from a few countries such as Taiwan, China, and South Korea had higher rate of disabled JavaScript. Put the matter another way, the majority of the Internet users will produce web forensic data from access loggers and turned off JavaScript will give the evidence that bad bot is behind the web access.

The other limitation of our prototype is it relies on a third party library for computing the pressure patterns. If at any point, we are supposed to change the libraries, the access logger might have to undergo modification in order to adjust it to the new library.

If the attacker recognizes the working of proposed forensic framework, so we can expect that the attacker will find a way to conceal their presence and attempt to deceive the system. Here, we discuss possible way of forging the evidence at the client-side. The bot attacker can copy the patterns while a human is interacting with websites and play it back. The playback bots are not able to produce human patterns with high accuracy and therefore they can give us measurable evidence. For instance, the movement pattern such as horizontal and vertical scrolling of playback bots are more regular as compared to the human users. The other patterns which spotted the noticeable difference between playback bots and human users are standard deviation of inter request delay and entropy of inter request delay.

Conclusion

As the development of automated programs is increasing, so does the chances of its forensic analysis are producing key evidence in web forensic investigations. In view of the fact that we cannot completely stop automated programs since many Bots such as web crawler and chat Bot offer number of facilities to Information Technology. On the other hand automated programs can be exploited for variety of cyber crimes as presented in this paper. Conventional web forensic investigations are mainly based on server access logs which stores very basic information. Therefore, existing tools and research are not enough capable for advance Bot crime investigations. After systematically reviewing the existing research and tools, this paper proposes a new web forensic framework for Bot crime investigation to aid this process.

The proposed framework is the initial step in raising the awareness of cyber crimes using automated programs within the digital forensic community and giving a possible solution. Our web forensic framework is based on four different types of patterns which provide us the evidence of bad Bot activity on web applications. To evaluate the proposed framework, we applied it to the real web application environment and in experimental case scenario of Bot crimes. We presented the comprehensive web forensic investigation procedure for Bot crimes based on our framework and given the detailed technical methods used in each phase. Notably, we have visualized Bot forensic data in each pattern analysis phase and extracted the evidence of Bot activities on web applications.

References

A better deal for consumers in the digital age. https://www.gov.uk/government/news/a-better-deal-for-consumers-in-the-digital-age, 2017.
BOTS Act of 2016. https://www.congress.gov/bill/114th-congress/senate-bill/3183, 2016.
Babiker, M., Karaarslan, E., Hoscan, Y., 2018. Web application attack detection and forensics: a survey. In: 2018 6th International Symposium on Digital Forensic and Security (ISDFS). IEEE, pp. 1–6. March.
Bad bot report. https://resources.distilnetworks.com/white-paper-reports/bad-bot-report-2019, 2019.
Bours, P., 2012. Continuous keystroke dynamics: a different perspective towards biometric evaluation. Inf. Secur. Tech. Rep. 17 (1–2), 36–43.
Brand, J., Balvanz, J., 2005. Automation is a breeze with autoit. In: Proceedings of the 33rd Annual ACM SIGUCCS Conference on User Services. ACM, pp. 12–15. November.
Convicted AT&T hacker, 2012. https://www.forbes.com/sites/andygreenberg/2012/11/21/security-researchers-cry-foul-over-conviction-of-att-ipad-hacker/#370f896f6853.
eBay Inc and Bidder's edge Inc, 2000. http://www.tomwbell.com/NetLaw/Ch06/eBay.html.
Falk, R., 2018. Understanding Probability and Statistics: a Book of Problems. AK Peters/CRC Press.
Fowler, K., 2007. SQL Server database forensics. In: Black Hat USA Conference.
Franklin, S., Graesser, A., 1996. Is it an agent, or just a program?: a taxonomy for autonomous agents. In: International Workshop on Agent Theories, Architectures, and Languages. Springer, Berlin, Heidelberg, pp. 21–35. August.
Gilani, Z., Wang, L., Crowcroft, J., Almeida, M., Farahbakhsh, R., 2016. Stweeler: a framework for twitter bot analysis. In: Proceedings of the 25th International Conference Companion on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 37–38. April.
Haddaway, N.R., 2015. The use of web-scraping software in searching for grey literature. Grey J. 11 (3), 186–190.
How many people are missing out on JavaScript enhancement?. https://gds.blog.gov.uk/2013/10/21/how-many-people-are-missing-out-on-javascript-enhancement/, 2013.
Hughes, K., Lecky-Thompson, J., Ammon, M., Murphy, H., 2017. Track the Impact of Your Publications.
Jianhui, L.I.N., 2008. A web forensic system based on semantic checking. In: 2008 International Symposium on Computational Intelligence and Design, 1. IEEE, pp. 99–102. October.
Jin, J., Offutt, J., Zheng, N., Mao, F., Koehl, A., Wang, H., 2013. Evasive bots masquerading as human beings on the web. In: Dependable Systems and Networks (DSN), 2013 43rd Annual IEEE/IFIP International Conference on. IEEE, pp. 1–12. June.
Jiqiang, L., Zhen, H., Zengwei, L., 2002. Secure audit logs server to support computer forensics in criminal investigations. In: 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering. TENCOM'02. Proceedings, 1. IEEE, pp. 180–183. October.
Jones, K.J., 2003. Forensic Analysis of Internet Explorer Activity Files. Forensic Analysis of Microsoft Windows Recycle Bin Records.
Khanuja, H.K., Adane, D.S., 2012. A framework for database forensic analysis. Comput. Sci. Eng.: Int. J. 2 (3), 27–41.
Kumar, V., Singh, A.P., Rai, A.K., Wairiya, M., 2011. Self alteration detectable image log file for web forensics. Int. J. Comput. Appl. 975, 8887.
Kyaw, A.K., Sioquim, F., Joseph, J., 2015. Dictionary attack on Wordpress: security and forensic analysis. In: 2015 Second International Conference on Information Security and Cyber Forensics (InfoSec). IEEE, pp. 158–164. November.
Meshram, B.B., Patil, D.N., 2018. Digital forensic analysis of hard disk for evidence collection. Int. J. Cyber-Secur. Digital Forensics 7 (2), 100–110.
Nelson, R., Shukla, A., Smith, C., 2020. Web browser forensics in google chrome, mozilla Firefox, and the tor browser bundle. In: Digital Forensic Education. Springer, Cham, pp. 219–241.
Non-infringing content become copyright infringement. https://www.techdirt.com/articles/20090605/2228205147.shtml, 2009.
Now you can have more chances to get tatkal seats, thanks to Railway's this action. https://economictimes.indiatimes.com/industry/transportation/railways/indian-railways-roots-out-illegal-softwares-more-tatkal-tickets-for-passengers-now/articleshow/74193975.cms, 2020, February.
Oh, J., Lee, S., Lee, S., 2011. Advanced evidence collection and analysis of web

browser activity. Digit. Invest. 8.
Palmer, Gary, 2001, August. L. A Road Map for Digital Forensic Research. DFRWS. DTRT0010e01.
Pereira, M.T., 2009. Forensic analysis of the Firefox 3 Internet history and recovery of deleted SQLite records. Digit. Invest. 5 (3–4), 93–103.
Picotti, L., Salvadori, I., 2008. National Legislation Implementing the Convention on Cybercrime-Comparative Analysis and Good Practices. Directorate General of Human Rights and Legal Affairs Council of Europe.
Pollitt Mark M. Computer forensics (1995). an approach to evidence in cyberspace. In: Proceeding of the National Information Systems Security Conference, pp (487-491).
Pressure.js library. Retrieved from. https://pressurejs.com/.
Rahman, R.U., Tomar, D.S., 2018. Botnet threats to E-commerce web applications and their detection. In: Improving E-Commerce Web Applications through Business Intelligence Techniques. IGI Global, pp. 48–81.
Railways to fine Rs. 2, 00,000 for online ticket fraud. https://www.business-standard.com/article/indian-railways/beware-scamsters-railways-to-fine-rs-2-00-000-for-online-ticket-fraud-soon-118091600159_1.html, 2018.
Tan, P.N., 2018. Introduction to Data Mining. Pearson Education India.
Thelwall, M., 2001. A web crawler design for data mining. J. Inf. Sci. 27 (5), 319–325.
United States court of appeals. http://www.internetlibrary.com/pdf/efculturaltravel-zefer-1-cir.pdf, 2001.
Wang, X., Kohno, T., Blakley, B., 2014. Polymorphism as a defense for automated attack of websites. In: International Conference on Applied Cryptography and Network Security. Springer, Cham, pp. 513–530. June.
What percentage of browsers with javascript disabled?. https://blockmetry.com/blog/javascript-disabled, 2016.
Wilbur, K.C., Zhu, Y., 2009. Click fraud. Market. Sci. 28 (2), 293–308.
Xrumer, 2012. http://www.botmasterlabs.net/.
Yoon, J., Jeong, D., Kang, C.H., Lee, S., 2016. Forensic investigation framework for the document store NoSQL DBMS: MongoDB as a case study. Digit. Invest. 17, 53–65.