Towards Algorithmic Identification of Online Scams

Emad Mohammad Hussein Badawi

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the
Doctorate in Philosophy degree in Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

In "web-based scams", scam websites provide fraudulent business or fake services to steal money and sensitive information from unsuspecting victims. Despite the researchers' efforts to develop anti-scam detection techniques, the scams continue to evolve and cause online threats. State-of-the-art anti-scam research still faces several challenges, such as automatically acquiring a labeled scam dataset and providing early detection and prevention mechanisms to attacks that use cryptocurrency as a payment medium.

In this thesis, we implement a data-driven model to detect and track web-based scams with a web presence. Given a few scam samples, our model formulates scam-related search queries and uses them on multiple search engines to collect data about the websites to which victims are directed when they search online for sites that may be related to the scam. After collecting a sufficient corpus of web pages, our model semi-automatically clusters the search results and creates a labeled training dataset with minimal human interaction.

Our model proactively looks for scam pages and monitors their evolution over time rather than waiting for the scam to be reported. Whenever a new scam instance is detected, the model sends it automatically to the eCrime eXchange data warehouse in real-time. We have used the model to investigate and gain knowledge on two scams; the "Game Hack" Scam (GHS) and the "Bitcoin Generator Scam" (BGS). To the best of our knowledge, GHS and BGS have not been well studied so far, and this is the first systematic study of both scams.

GHS targets game players, in which the attackers attempt to convince victims that they will be provided with free in-game advantages for their favorite game. Before claiming these advantages, the victims are supposed to complete one or more tasks, such as filling out "market research" forms and installing suspicious executable files on their machines. Over a year of crawling, we uncovered more than 5,900 unique domains. We estimate that these domains have been accessed at least 150 million times from 2014 until 2019.

BGS is a simple system in which the scammers promise to "generate" new bitcoins using the ones sent to them. BGS is not a very sophisticated attack; the modus operandi is to put up some web page that contains the address to send the money and wait for the payback. Over 21 months of crawling, we found more than 3,000 addresses directly associated with the scam, hosted on over 1,200 domains. Overall, these addresses have received (at least) over 9.6 million USD. Our analysis showed that a small group of scammers controls the majority of the received funds. The top two groups have received around 6 million USD, which is more than half of the total funds received by the scam addresses.

Acknowledgements

All praises and glory be to **Allah** who helped me to achieve this work. **Prophet Mohammad** (Peace be upon him) said "**He who does not thank the people is not thankful to Allah**".

First of all, I would like to thank my supervisor, Professor Guy-Vincent Jourdan, for his patient guidance and helpful suggestions during my study. My deepest gratitude also goes out to Professor Gregor V. Bochmann and Dr. Viorel Iosif Onut for their advice and feedback through my research.

Many thanks to all the people of my defense committee: Professor Kenneth B. Kent, Professor Anil Somayaji, Professor Paula Branco, and Professor David Knox for giving valuable feedback on this thesis.

I would like to thank my colleges of Software Security Research Group (SSRG), Qian Cui and Sophie Le Page. Thanks for your help during my study and the happy time we spent together.

Thanks to the financial support from the IBM Center for Advanced Studies (CAS) and the Natural Sciences and Engineering Research Council of Canada (NSERC). Without these financial support, I couldn't start my study and life here.

Last but not least, I want to express my sincere gratitude to my parents and my wife for providing me with endless support and continuous encouragement.

Dedication

This is dedicated to my late father, who was not able to see me graduate.

Table of Contents

List of Tables **x**

List of Figures **xii**

List of Abbreviations **xv**

1 Introduction **1**

1.1 The Game Hack Scam ("GHS") 2

1.2 The Bitcoin Generator Scam "BGS" 6

1.3 Motivation and Challenges 12

1.4 Contribution 14

1.4.1 Publications 15

1.5 Organization 16

2 Literature Review **17**

2.1 Introduction 17

2.2 Cryptocurrencies Emerging Threats and Defensive Mechanisms: Systematic Review 20

2.2.1 With the introduction of cryptocurrencies, what are the types and scales of cybercriminal activities reported by researchers? [Rq.1] . . 25

2.2.2 For cryptocurrency cybercrimes detection and prevention, what are the public datasets provided in the literature, and how have these datasets been collected? [Rq.3] 29

2.2.3 What are the proposed defensive mechanisms available to detect cybercriminal activities, and what is the reported effectiveness of these mechanisms? [Rq.2] . . . 31
2.2.4 SLR Conclusions . . . 36
2.3 Survey Scam . . . 36
2.4 Technical Support Scam . . . 37
2.5 Other Scams . . . 39
2.6 Malware Detection in Android Mobile Applications . . . 41
2.7 Discussion . . . 41
2.7.1 Communication Medium Used to Reach the Victims . . . 41
2.7.2 Techniques Used to Approach the Victims . . . 43
2.7.3 Effects on the victims . . . 44
2.7.4 Mitigation Methods . . . 45
2.8 Conclusion . . . 46

3 The "Game Hack" Scam 49

3.1 Introduction . . . 49
3.2 Methodology . . . 50
3.2.1 Training Dataset Creation . . . 50
3.2.2 Search Query Generator . . . 51
3.2.3 Web Crawler . . . 52
3.2.4 Classifier . . . 53
3.2.5 Offers Crawler . . . 56
3.2.6 Clustering and Analysis . . . 57
3.3 Scam Collection and Measurement . . . 57
3.3.1 Classification Result . . . 57
3.4 Analysis . . . 59
3.4.1 Page Contents . . . 59
3.4.2 GHS Analysis . . . 59

3.4.3 Offers . . . 62
3.4.4 Domains Analysis . . . 64
3.4.5 Executable Files and Modified APKs . . . 66
3.5 Bitly Links Analysis . . . 67
3.5.1 Click Through Analysis . . . 67
3.5.2 Monthly URL Clicks and Creation Analysis . . . 68
3.5.3 Country and HTTP Referrer Clicks . . . 68
3.6 Study of two Templates Providers . . . 69
3.6.1 *CPABUILD.COM* . . . 70
3.6.2 *OGADS.COM* . . . 72
3.7 Limitations . . . 72
3.8 Conclusions . . . 73

4 The Bitcoin Generator Scam **75**

4.1 Introduction . . . 75
4.2 Applying the Methodology . . . 76
4.2.1 Training Dataset Creation . . . 76
4.2.2 Search Query Generator . . . 78
4.2.3 Web Crawler . . . 79
4.2.4 Classification Module . . . 79
4.2.5 Cryptocurrencies Addresses Crawler . . . 81
4.2.6 Clustering and Analysis . . . 82
4.3 Scam Collection and Measurement . . . 83
4.4 Analysis . . . 84
4.4.1 Page Contents . . . 86
4.4.2 Crawler Effectiveness . . . 86
4.4.3 Bitcoin Addresses Payment Analysis . . . 87
4.4.4 Payback Analysis . . . 88
4.4.5 Scam Addresses Delivery Techniques . . . 91

4.4.6 BGS Addresses Statistics . . . 93
4.4.7 Addresses Reuse . . . 95
4.5 Other BGS Cases . . . 96
4.5.1 Malicious Executables . . . 96
4.5.2 Click per Action (CPA) Scam . . . 96
4.6 Scam Clustering . . . 98
4.7 Investigating BGS in Other Languages . . . 101
4.8 Limitations . . . 103
4.9 Conclusion . . . 105

5 Generalizing our Model and Increasing Automation 107

5.1 Introduction . . . 107
5.2 Query Generator . . . 108
5.3 Web Crawler . . . 109
5.4 Clustering and Dataset Creation . . . 110
5.5 Classification Model . . . 111
5.6 Creating Training Datasets Automatically for BGS and GHS . . . 112
5.6.1 Collecting the Corpus of Web Pages . . . 113
5.6.2 Clustering and Dataset Creation . . . 113
5.6.3 Validating the Training Datasets . . . 117
5.7 Discussion . . . 120
5.7.1 Model Persistence . . . 120
5.7.2 User Education . . . 121
5.8 Limitation and Future Work . . . 122
5.9 Conclusion . . . 123

6 Conclusion and Future Work 124

6.1 Conclusion . . . 124
6.2 Future Work . . . 125

6.2.1 Study of the GHS Templates Providers 125
6.2.2 Validate our Approach Using other Types of Web-based Scams . . . 126
6.3 Improving the Automation Aspect of our System 126

A Cryptocurrencies Emerging Threats and Defensive Mechanisms: A Systematic Literature Review (Research Method) 127

A.1 Research Questions 127
A.2 Search Strategy 128
A.2.1 Source Databases 129
A.2.2 Abstract Search Query 129
A.3 Inclusion Criteria 130
A.4 Exclusion Criteria 130
A.5 Quality Assessment Criteria 131
A.6 Study Selection & Data Collection 131

B Cryptocurrencies Emerging Threats and Defensive Mechanisms: A Systematic Literature Review (Breakdown per Article) 133

C doublebitcoin.win Addresses Graph Representation 144

References 146

List of Tables

2.1 Summary of the papers included in the SLR. 22

2.2 Publicly available data provided in the literature. 32

2.4 Useful resources provided in the literature. 33

2.5 Utilized sources for extracting detection features. 34

2.6 Fraud activities categorization 42

3.1 Summary and Examples of Generated n-grams Related to GS. 52

3.2 Results of a 10-Fold cross-validation on the five classifiers. 55

3.3 The effect of applying the filters on the training dataset. 56

3.4 Most common top-level domains (TLDs) for the final URLs of GHS instances. 65

3.5 Referrers and countries with the highest number of clicks (countries analysis is normalized using the clicks-population ratio) 70

4.1 Results of a 10-Fold cross-validation with five classifiers. 80

4.2 Classifier accuracy on pages that have not been observed in the training phase. 81

4.3 Detailed analysis for the scam addresses payback (Transac refers to transactions). 92

4.4 General statistics 94

4.5 Crosschecking the BGS dataset with other public datasets 95

5.1 Summary of the Generated Queries for BGS and GHS 112

5.2 Summary of the datasets obtained through the crawler 113

5.3 Clustering results 116

5.4 Results of 10-Fold cross-validation of the five classifiers on BGS dataset . . 118

5.5 Results of 10-Fold cross-validation of the five classifiers on GHS dataset . . 118
5.6 Classifier results on pages that have not been used in the training phase . . 119

A.1 Search query related terms 129

B.1 The reported detection results in the literature 133
B.2 Reported cybercrimes scale estimation in the literature. 136
B.3 Sources used in the literature for dataset collection. 139

List of Figures

1.1 An example of resources used to buy in-game outfits (PUBG). 3
1.2 An example of resources used to buy in-game helper tools (Clash of Clans). 3
1.3 In-game resources for the game Toon Blast. 3
1.4 In-game resources for the game Candy Crush. 3
1.5 An example of GHS attack 4
1.6 Examples of the GHS offers 5
1.7 2021 cryptocurrency anti-money laundering report (reproduced from CipherTrace report[11]). 7
1.8 An example of BGS attack. 11
1.9 Scam statistics (reproduced from scamwatch [17]) 13

2.1 Number of cryptocurrency attack papers published per year. 22
2.2 Number of published articles per cyberattack type. 26
2.3 Number of published papers per currency. 27
2.4 Categorization based on the communication medium used to reach the victim. 43
2.5 Categorization based on the attacker approaching technique. 44
2.6 Categorization based on the effects on the victim. 45
2.7 Categorization based on the detection/prevention methods. 46

3.1 Games scam detection and analysis model 50
3.2 Number of GHS instances found by search engines per month 58
3.3 Percentage of GHS instances found in the second page over all scam pages 58
3.4 Word cloud based on the text contents of the gathered technical GHS pages 60

3.5 Examples of GHS templates 61
3.6 Number of games each scammer spans 63
3.7 Breakdown of the number of GHS instances per ID. 63
3.8 An example of the scam content locker. 64
3.9 Percentage of Offers Reached per Number of CLs 64
3.10 GHS click through analysis 68
3.11 GHS click duration analysis 68
3.12 GHS clicks count per month 69
3.13 GHS clicks per country 69
3.14 *Cpabuild.com* Template Creation Process 71
3.15 *CPAbuild.com* Locker Types 72
3.16 *OGADS.com* Locker Types 73

4.1 BGS detection and analysis model. 76
4.2 Number of BGS URLs detected per week. 84
4.3 Number of Bitcoin addresses detected per week. 85
4.4 Word cloud based on the text contents of the gathered technical GHS pages 86
4.5 Daily incoming transactions to BGS addresses. 88
4.6 Daily deposited money to BGS addresses. 89
4.7 A comparison between the incoming and outgoing deposits in BGS addresses. 90
4.8 A real world example of a BGS instance in which the payment address is selected randomly from a list. 93
4.9 An example of virus total scan results. 97
4.10 An example of the scam content locker. 98
4.11 An example of the scam offers. 98
4.12 BGS addresses clustering analysis. 100
4.13 Cluster #511 BGS domains and addresses connection graphs (generated by Gephi using layout Fruchterman Reingold). 102
4.14 Features classifiers area under the curve (AUC). 104

5.1 A Generic Model for Scam Detection and Analysis. 108
5.2 Tag vectors 114
5.3 Optimal threshold of BGS clusters 115
5.4 Optimal threshold of GHS clusters 115
5.5 Number of clusters vs number of scam words in the web pages. 116

A.1 Review methodology. 128

C.1 A graph representation of the doublebitcoin.win BGS domain addresses. . . 145

List of Abbreviations

(in alphabetical order)

API	Application Program Interface
APK	Android Application Package
BGS	Bitcoin Generator Scam
CL	Content Locker
DDoS	Distributed Denial-of-Service
DOM	Document Object Model
GHS	Game Hack Scam
HYIP	High Yield Investment Programs
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
KNN	K-nearest Neighbors
ML	Money Laundering
MLP	Multi-layer Perceptron
NB	Naive Bayes
RF	Random Forest
SVC	Support Vector Classifier
TLD	Top Level Domain
TNR	True Negative Rate
TPR	True Positive Rate
URL	Uniform Resource Locator

Chapter 1

Introduction

The Internet has caused a shift in people's lifestyles by providing companies and people with a new way of connecting, which created a more connected world. For example, online games have begun to match or even replace traditional games. This has also caused companies to focus on creating new Internet services and products to dominate the market. However, this shift enables criminals to launch new types of crime using computers and networks, known as cybercrime. These cybercrime attacks include, but are not limited to, game-related scams, cryptocurrencies scams, survey scams, technical support scams, and phishing.

Scam attacks are a type of cybercrime in which attackers are getting unsuspecting victims to willingly reveal sensitive information, hand over cash, or even do harmful actions for the scammer's benefit. The attackers use social engineering to misrepresent themselves by impersonating familiar contacts of the victims or someone with authority or skill, such as a company representative, police officer, lawyer, and Internal Revenue officer. Despite industry and academia's efforts in recent years, anti-scam research still faces several challenges, especially when it comes to acquiring a labeled scam dataset and creating a 0-day detection model to the attacks that use cryptocurrencies as a payment medium.

In this thesis, we propose a data-driven system to detect, track, and analyze web-based scam attacks. Furthermore, we look at the problem of cryptocurrency web-based scams from new perspectives. Instead of studying obsolete previously reported scam campaigns, we proactively look for scam pages and monitor their evolution over time. We thus detect these instances before any victims fell for the scam. Furthermore, we have adapted our methodology to create a training dataset for the scam algorithmically. After collecting a sufficient corpus of web pages, our model semi-automatically clusters the search results and creates a labeled training dataset with minimal human interaction.

In this chapter, we first discuss two types of scam that we used to validate our model; the game hack scam in Sections 1.1, and the bitcoin generator scam in 1.2). In section 1.3 we discuss the motivation and challenges of this research. We finally summarize our contributions and the organization of the thesis in Section 1.4 and Section 1.5 respectively.

1.1 The Game Hack Scam ("GHS")

The gaming industry is one of the most profitable industries in the world. Its total market value is 159.3 billion worldwide in 2020[1]. This value is expected to increase and reach 200 billion in 2023[1], and the number of game players is expected to increase from 2.6 billion in 2020 to be 2.725 billion by 2021[2].

Game developers depend mostly on the purchase of in-game resources as well as in-game advertisements to make a profit[3][4], Figures 1.3 and 1.4 are examples of in-game resources. These resources can be used in different ways, such as buying in-game outfits (Figure 1.1), in-game extra lives, and in-game helper tools (Figure 1.2).

With the modern connected era, many people have become accustomed to nearly instant gratification in many things they do, fast turnaround times, and quick wins[5][6]. As a result, users do not hesitate to pay money[7] or use cheats to win. They are willing to bypass the regular route and use "cracks", game-modifying software, or any other means of hacking to obtain these resources.

In this thesis, we give insight into an understudied social engineering attack targeting everyday web users, especially games players. We call this attack the Game-Hack Scam (GHS). In a nutshell, in GHS, the attackers claim that they can hack a specific game and provide the victim with free, unlimited resources or other advantages for their favorite game. To obtain these claimed advantages, the victims are asked to complete one or more tasks, called "offers". These so-called offers include, but are not limited to, subscriptions to questionable services and installation of executable files on the victim's device.

Figure 1.5 illustrates GHS and how a user is exposed to malicious advertisements or malware. Usually, the scam starts when a victim searches for cheats and hacks for their

[1] https://www.wepc.com/news/video-game-statistics/

[2] https://financesonline.com/number-of-gamers-worldwide/

[3] https://electronics.howstuffworks.com/free-to-play-games-make-money.htm

[4] https://financialpost.com/technology/download-code-how-exactly-do-companies-make-money-off-free-to-play-games

[5] https://studybreaks.com/culture/instant-gratification/

[6] https://theamericanscholar.org/instant-gratification/

[7] https://www.cnbc.com/2015/08/03/the-shocking-truth-about-mobile-gaming.html

Figure 1.1: An example of resources used to buy in-game outfits (PUBG).

Figure 1.2: An example of resources used to buy in-game helper tools (Clash of Clans).

Figure 1.3: In-game resources for the game Toon Blast.

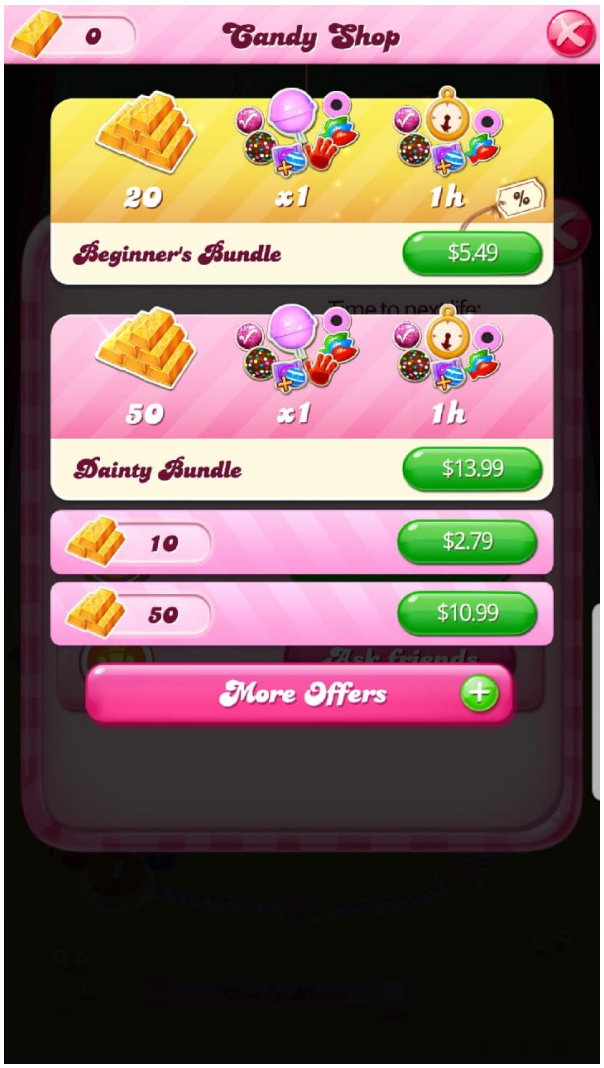

Figure 1.4: In-game resources for the game Candy Crush.

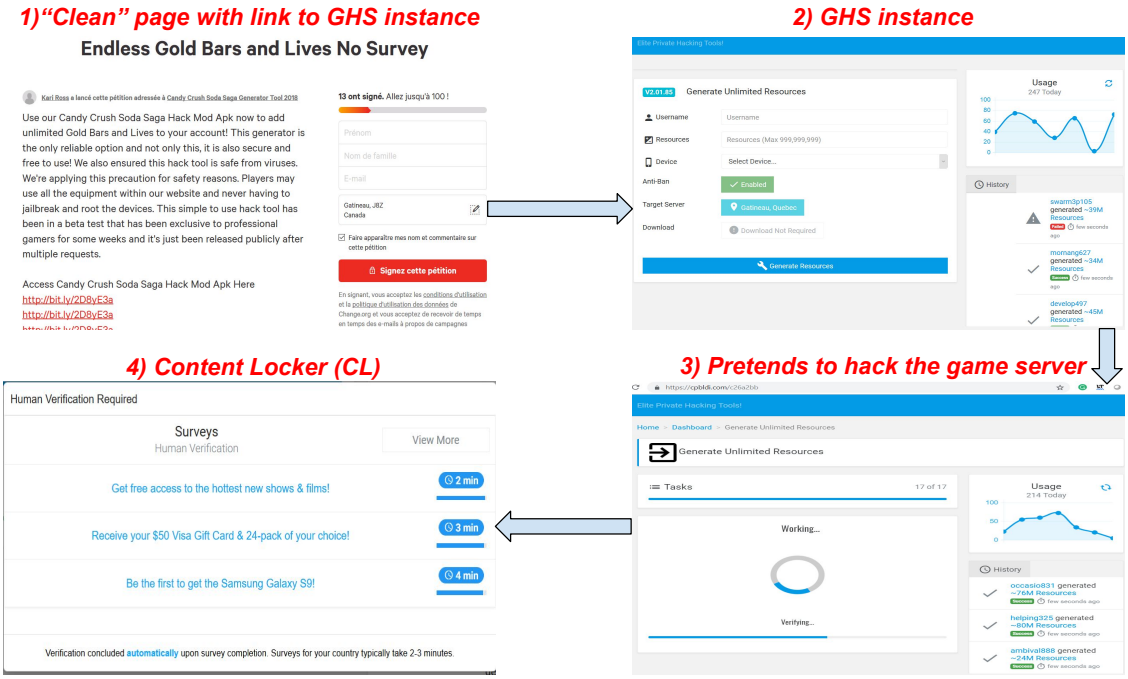

Figure 1.5: An example of GHS attack

game using search engines, social media, streaming sites, blogs, or any other site. The returned search results may directly contain GHS instances (GHSi) such as `https://cpbldi.com/c26a2bb` in Figure 1.5.

In other cases, the search results are benign pages that the attackers use to advertise for the scam. For example, the article published in *change.org*[8] shown in Figure 1.5 was written by a scammer to advertise a scam instance, and it contains a link leading to the actual scam page[9].

We call the scam websites "generators". These generators are carefully designed web pages that attempt to convey to the victim the advanced technical abilities of the scammer and a large, satisfied user base for the GHS instance. GHS instances tend to use a variety of similar templates. Many of these templates ask for the victim's identifier on the game and the resources that the victim wants. Other templates attempt to be more convincing by asking for additional information such as the game platform, the region they live in, and the ability to use a proxy. Also, these advanced templates could display a fake chat box and a pop-up showing claimed current users and the number of resources they supposedly gained.

Once the information is provided, the generator page pretends to perform some hacking

[8]Accessible at `https://bit.ly/2F4IE2I` at the time of writing.
[9]`https://flipmix.win/r/c26a2bb3`

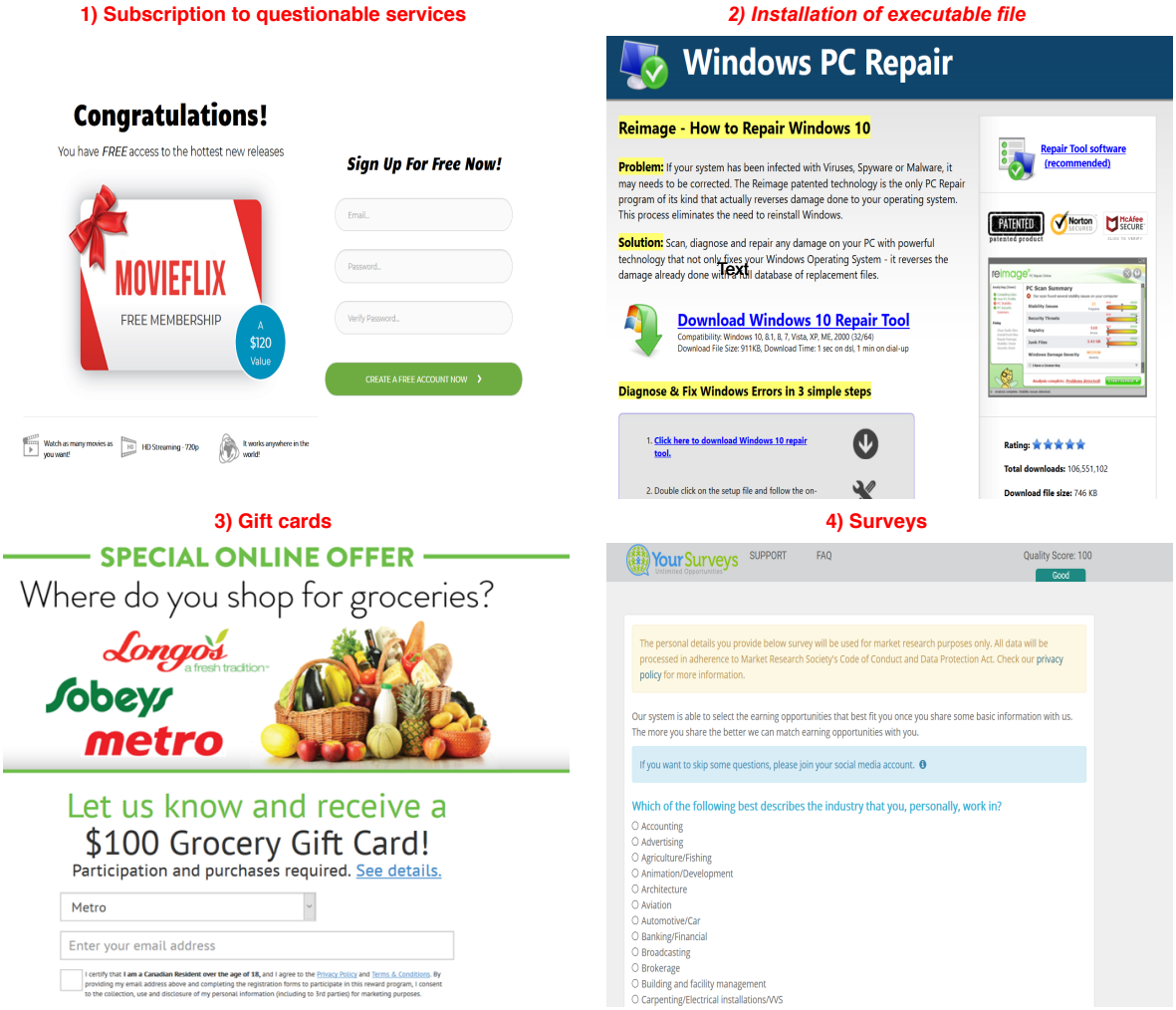

Figure 1.6: Examples of the GHS offers

process, as seen in Figure 1.5 image 3. After that, a pop-up appears claiming that the hack was successful, and the victim is then invited to a "verification" step. During this verification process, some screen is shown to the user, asking to complete one or more tasks, called "offers". This type of screens is called a "content-locker" (CL) by the creator of these scams. The "CL" with its set of offers is what the scammer ultimately wants the victim to see, as they lead to the payload. An example is shown Figure 1.5 image 4. These offers include, but are not limited to, subscriptions to questionable services (such as image 1) in Figure 1.6 and installation of executable files on the victim's device (such as image 2) in Figure 1.6. In some cases, the generator is bypassed, and the victim is directly presented with the CL or the payload.

In other cases, the scammers carry out their attacks without using the generators. Instead, the victims are asked to install new software, claimed to be either a modified

version of the original game or some sort of game modifying software ("cheat-engines"). We have found more than 300 modified android games that specifically target android phones. We also found more than 50 games modifying software, which are flagged as harmful files by some of the anti-viruses used by virus total online scan, while other anti-viruses did not catch them.

1.2 The Bitcoin Generator Scam "BGS"

In recent years, there has been a rise in the use of cryptocurrencies as an investment platform [82]. As of November 6^{th}, 2021, there are 13,677 different cryptocurrencies, with a capitalization market of approximately $2,719 billion USD[10]. The most popular cryptocurrencies are Bitcoin and Ethereum, which have a capitalization market of approximately $1,156 billion USD and $532 billion USD, respectively.

Bitcoin is a decentralized cryptocurrency that has become popular in the last ten years. It is a peer-to-peer electronic currency that can be sent from one user to another without the involvement of a trusted authority such as an administrator or a central bank [109, 147, 154]. It first appeared in a white paper by "Satoshi Nakamoto" [109]. The actual identity of Nakamoto is still unclear. Unlike traditional currencies, bitcoin has two key features: Transparency and Pseudo-anonymity [109,147,150]. It is transparent because the transactions are publicly announced in a decentralized ledger called a blockchain. The Pseudo-anonymity comes from the fact that the users use pseudonyms (addresses). These pseudonyms are not related to individuals; they are computed from the user's public key [147]. Moreover, bitcoin addresses can be generated at will [147]. As a result, users can create a unique address for each transaction. This increases privacy by creating an additional layer to keep the addresses from being linked to a specific owner [109].

Cybercriminals have leveraged bitcoin pseudo-anonymity in their attacks. According to CipherTrace spring-2020 report[11] , the value of thefts, hacks, and scams has more than doubled in 2019 when compared to 2018 and was more than 230 times the value of 2017; in 2019 only, more than $4.52 billion USD was stolen away from cryptocurrency exchanges and users. However, although 2020 had the second-highest value in crypto-crime ever, the crypto-crime value had a significant decreased of 57% decreasing from $4.5 billion USD in 2019 down to $2.1 billion USD in 2020.

[10] `https://coinmarketcap.com/`

[11] https://ciphertrace.com/2020-year-end-cryptocurrency-crime-and-anti-money-laundering-report/, https://ciphertrace.com/cryptocurrency-crime-and-anti-money-laundering-report-may-2021/

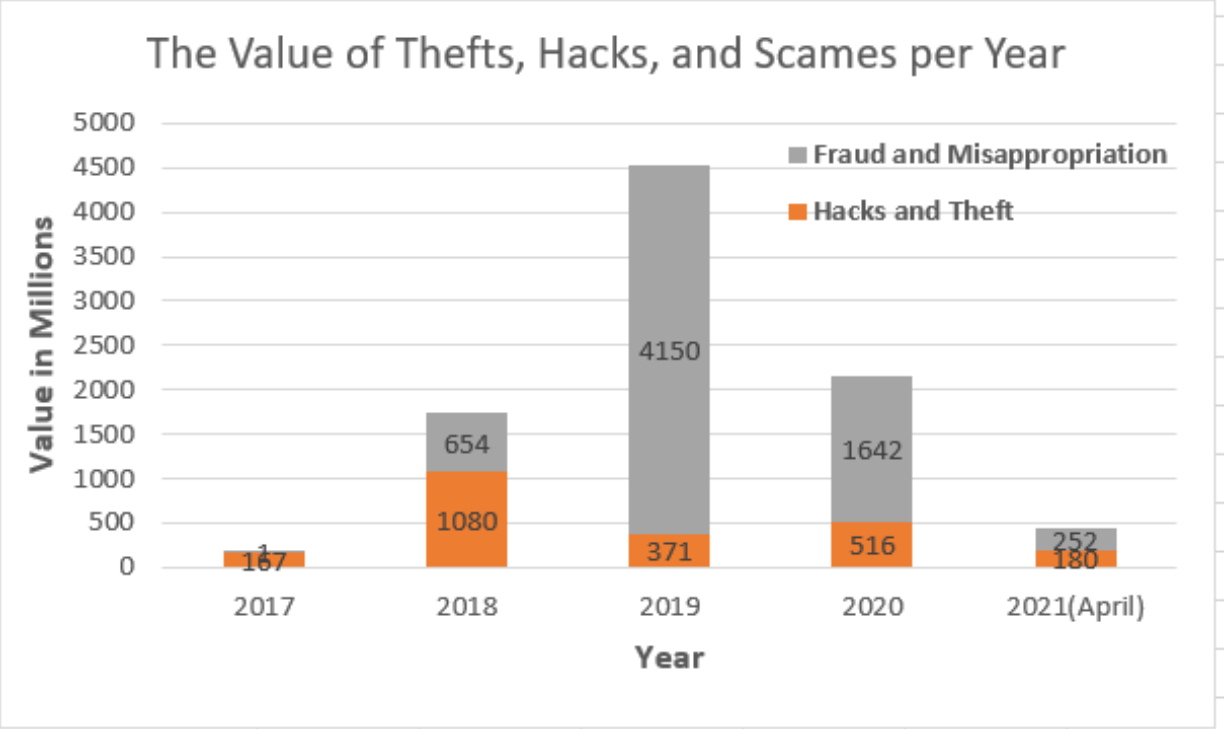

Figure 1.7: 2021 cryptocurrency anti-money laundering report (reproduced from CipherTrace report[11]).

As shown in Figure 1.7, in 2020, losses from cryptocurrencies exchanges, hacks, and thefts reached $516 million USD, while the majority of the losses ($1.642 billion USD) was due to fraud and misappropriation of funds. For example, *KuCoin*, the Singapore-headquartered digital asset exchange was one of the targets of the hacks and theft attacks in 2020; on September 26, the exchange announced an unauthorized transfer of Bitcoin and Ethereum tokens to an unknown wallet, affecting around $150 million USD in users fund. On the other hand, the "WoToken" Ponzi scheme defrauded investors of over $1 billion USD with its "exist" scam[11].

Lendf.me, a decentralized lending protocol operated by Chinese DeFi upstart dForce was one of the targets of the hacks and theft attacks in 2020; on April 19, $25 million USD worth of cryptocurrency were stolen *Lendf.me*. On the other hand, the "EOS Ecosystem" wallet defrauded investors of $52 million USD in a Ponzi scheme by enticing investors with promises of favorable returns. DeFi-related hacks and fraud grew quarter over quarter. In just the first 4 months of 2021 the value of DeFi-related hacks and fraud has already surpassed the 2020's all-time high[11].

Cybercriminal attacks using Cryptocurrencies take many forms. "High yield investment programs" (HYIP) is one of the popular examples of the scams that cybercriminals carry

out [21,147,153,154]. HYIP is a scam in which investors are promised a high interest rate, e.g., more than 1-2% per day [147]. Perhaps the most famous HYIP scammer was Charles Ponzi, who claimed in the early 1920s to run an arbitrage; the investors were promised a 50% profit within 45 days, or 100% profit within 90 days. Because of Charles Ponzi, HYIP is sometimes called a *Ponzi scheme* [147].

Money laundering (ML) [29,105], ransomware [26,98,132], and pump and dump (P&D) [39, 82,156,167] are other popular examples. ML describes the process of disguising the sources of illegal profits generated by criminal activity. It aims to hide the link between original criminal activities and the corresponding funds by passing the money through a complex sequence of commercial transactions or banking transfers [118].

Ransomware is a denial-of-access attack in which a malicious piece of software locks and encrypts a victim's device data until a sum of money is paid [26]. Cryptocurrencies, usually Bitcoin, are often used for these payments. Recently, Riviera Beach officials voted to pay 65 bitcoins, worth $600,000 USD at the time, to a cybercriminal who seized and shut down the city's computer systems. The resulting outage forced the local fire and police departments to write down hundreds of 911 calls on paper[12].

P&D scheme is a type of fraud in which the fraudster aims to make a profit from stock trading by artificially manipulating stock prices. In P&D, the attackers purchase stocks at a low price (pump) then spread misleading recommendations and positive statements to convince other investors to buy that stock, which increases its price. The attacker then sells (dump) their stock at a mark-up, causing a decrease in the stock price and inflicting losses to other investors [39,82,156]. P&D is an old fraudulent activity that started in the 1700s in London's South Sea Company. Aiming for an easy profit by selling cheap stocks at high prices, a stock owner started making positive statements about the company and its profit. This fraudulent activity becomes to be known as "the South Sea Bubble", and became an early example of a P&D scheme [82].

Another way to attack cryptocurrencies is to use a distributed denial-of-service (DDoS) attack [3,19,57,155]. DDoS are cyber-attacks that render a website or a service inoperable by overwhelming it with a flood of traffic. Although blockchains distributed ledgers are robust against DDoS attacks, it is still possible to attack mining nodes that use an outdated protocol [165], or to attack cryptocurrencies mining pools or exchanges [19, 78, 166, 172]. Although these attacks are not meant to directly steal currencies or affect the network's performance, they are affecting the value of the currency and ultimately lead to the currency's depreciation and benefit the attacker [3, 19, 57, 155].

[12] `https://cbs12.com/news/local/riviera-beach-commissioners-vote-to-pay-ransom-to-hacker-who-shut-down-city-computers`

A completely different attack based on cryptocurrencies is what is called "Cryptojacking" [175]. It leverages the ability of web browsers to execute code. The code in question is meant to "mine" cryptocurrencies. For example, the now-defunct website coinhive.com distributed browser-based cryptomining code to mine bits of the Monero cryptocurrency. The original idea was that it was a way for a user to compensate a website provider by lending some CPU cycles of their browser when accessing the site. This was seen as an alternative to advertisement to monetize 'free access" resources. In-browser cryptomining can also be used for rate limitation as a replacement for CAPTCHAs [87, 107, 126]. However, this can be abused in the so-called cryptojacking attack, when this is done without the consent of the user or the site owner, or when the code is tampered with, e.g., to modify the payment address [87, 107, 126, 175]. Cryptojacking attacks are easy to deploy, difficult to detect, and can be found on any Internet-connected device with a CPU, such as mobiles, PCs, and IoT devices [175].

The current state of the art for bitcoin scam detection usually relies on a classification model to detect scam addresses based on transactions history [21, 147, 150, 154]. These addresses are either collected manually, e.g., by searching on bitcoin discussion forums such as bitcointalk.org [21], or they come from semi-automated web crawls of the same forums, followed by manual addresses collection [147, 150, 154]. Once a set of addresses used in the scam has been collected, the transaction history of these addresses is used to train a classification model [21, 147]. The classifier is trained on features such as the frequency of transactions, the ratio of received/sent transactions to all transactions, the address lifetime, or the "payback" ratio, which is the ratio of addresses that appears in the input and output sides of address transactions.

However, the increasing number of transactions recorded on the blockchain[13] makes it difficult to extract meaningful patterns that can be used in fraud detection [21]. Additionally, based on transaction history, these methods are by nature only able to detect a scam address after the fact, once some victims have been defrauded.

In this thesis, we look at a scam that has emerged with the rise of cryptocurrencies. We call this attack the "Bitcoin Generator Scam" (BGS). In BGS, the attackers claim that they will provide free bitcoins in return for a small mining fee, using dubious claims such as their ability to "hack the blockchain ledger". BGS attacks start with an online website targeting their victims. We call these websites "generators". These generators are carefully designed web pages that attempt to convey to the victim the advanced technical abilities of the scammer and a large, satisfied user base for the BGS instance. Some BGS instances

[13]Over 6840 million transactions at the time of writing:`https://www.blockchain.com/charts/n-transactions-total`.

display a fake chat box and a pop-up showing claimed current users and the number of mined bitcoins they supposedly gained.

BGS attacks can be directly advertised, e.g., on social media. Still, victims can also be actively seeking easy profit by looking online for "Bitcoin hack services" using search engines, social media, streaming sites, blogs, etc. (Figure 1.8, image 1). The search results may link directly to BGS instances or benign pages that the attackers have used for advertising the scam instances, in which the attacker describes the scam and provides a link to access the scam page.

Once a BGS instance like the one shown in Figure 1.8 image 2 is accessed, the victim is asked to provide the number of coins they want to mine and the bitcoin address in which the mined coins will be deposited. Once the victim provides the information, the BGS pretends to perform some "hacking" (Figure 1.8, image 3). Finally, some success message is displayed, and the victim is asked to pay a mining fee to collect the funds (Figure 1.8 image 4). In many cases, the fees are a fixed number of satoshis. In other cases, the attacker promises that the victim will receive multiple of the amount they pay.

In other variations of the attack, rather than asking for a mining fee, the scammers ask the victims to either complete one or more tasks or download and install a mining executable file to complete the mining process. In the former case, after the success message is displayed, the victim is invited to a "verification" step. During this verification process, some screen is shown to the user, asking to complete one or more "offers". These so-called offers include, but are not limited to, filing out "market research" forms, clicking through endless "surveys", getting the victims to subscribe to questionable services, collecting personal information, installing suspicious executable files on their machines, etc.. In the latter case, the mining executable files were reported as harmful by virus total[14].

Some authors (e.g. [20]) characterize Ponzi schemes by their pyramidal structure and the payout to existing investors using funds from new investors. By this definition, BGS is not a Ponzi scheme since most BGS instances do not require investors to enroll new investors, and as discussed in Section 4.4.4 we usually do not find any evidence of payout at all. However, some other authors characterize Ponzi schemes by their extremely high rates of return [104,153], and BGS certainly fall under that category, with advertised return rate in the range of 100% in 24 hours.

[14] `https://www.virustotal.com/gui/`

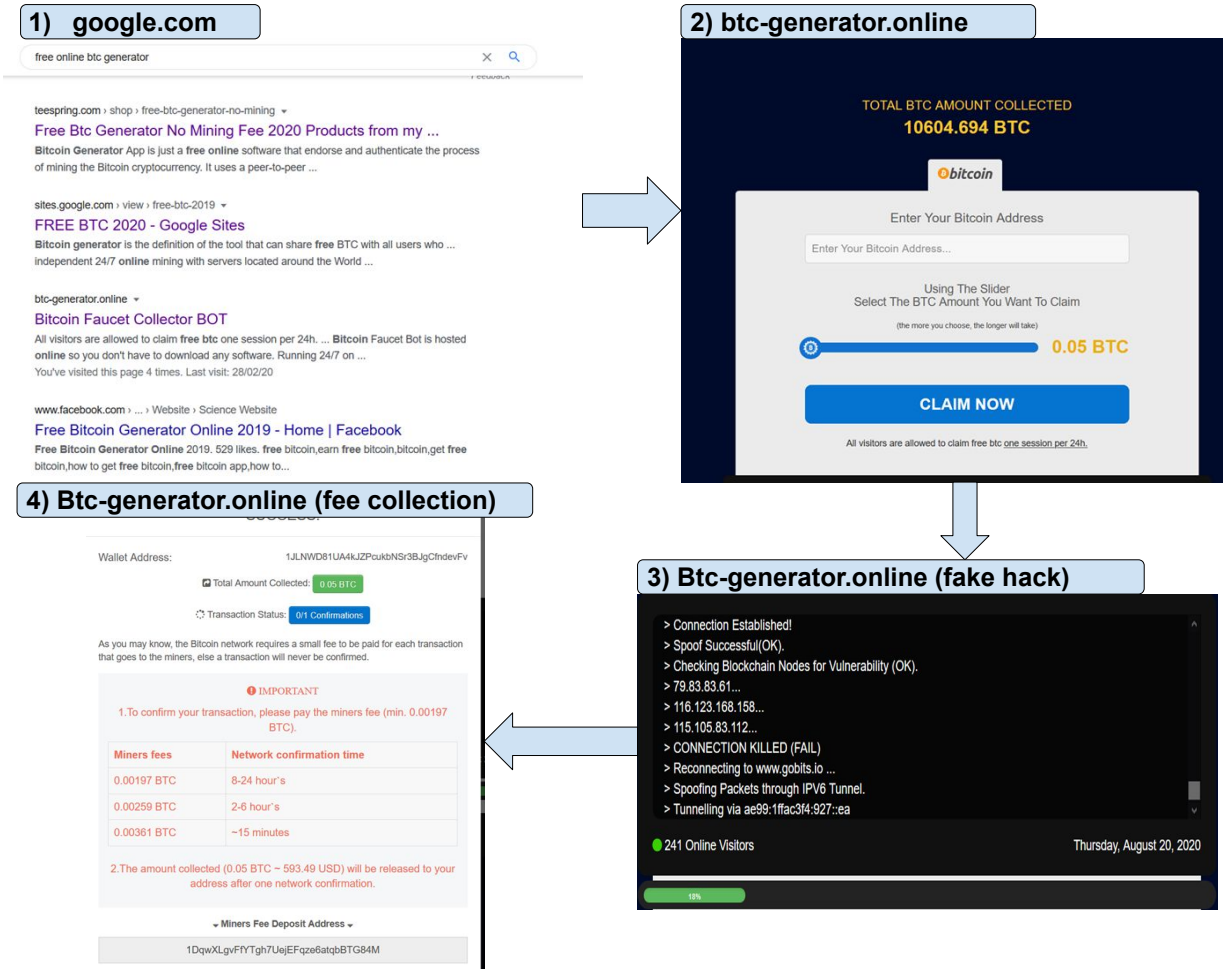

Figure 1.8: An example of BGS attack.

1.3 Motivation and Challenges

Both academia and industry have put effort into the research of combating scams and inventing various anti-scam solutions. For example, Google Safe Browsings[15] is used by browsers such as Google Chrome, Safari, and Firefox to provide their users with a builtin service to prevent scam attacks. Microsoft Edge and Internet Explorer browsers use a similar built-in protection tool called SmartScreen[16].

However, the growth of scam attacks has never stopped or even slowed down. According to Scam Watch[17], the value of stolen money and the total number of the scam cases are increasing over the years. As shown in Figure 1.9, the loss value in 2019,2020, and until mid 2021 are $142,698,377 USD, $175,694,583 USD, and $192,681,546 USD, respectively. Although the report covers the first eight months of 2021, the amount of stolen money is more than the total loss of the entire 2020 year. These numbers are even worse in the case of cybercrimes that targets cryptocurrencies. According to CipherTrace, the value of cryptocurrencies thefts, hacks, and scams has more than doubled in 2019 when compared to 2018 and was more than 230 times the value of 2017; in 2019 only, more than $4.52 billion USD was stolen away from cryptocurrency exchanges and users. However, although 2020 had the second-highest value in crypto-crime ever, the crypto-crime value had a significant drop down of 57% compared to 2019, going down from $4.5 billion USD to $2.1 billion USD[11]. In this thesis, our work was motivated by the following problems.

- **Analyzing GHS attack.** None of the current studies have analyzed the GHS attack. The only related studies are the ones studying the survey scam [43,88]. Many of the GHS final payload sites promise free vouchers, gift cards, and free products in exchange for completing surveys, and these websites are part of the survey scam.

  In this thesis, we report the first systematic study of GHS. We use our model to collect thousands of GHS web pages, interact with them, and collect the pages to which the victims will be directed after falling for the scam. We then analyze the collected data to give insight into this attack. Our results show that the attackers routinely target a vast array of games. Our results also show that the existing public blacklists are ineffective against this scam, and the scam instances exist for a long time.

- **Analyzing BGS attack.** Similar to the GHS attack, none of the current studies have investigated the BGS attack. The only related studies are the ones studying

[15]https://safebrowsing.google.com/

[16]https://support.microsoft.com/en-ca/help/17443/microsoft-edge-smartscreen-faq

[17]https://www.scamwatch.gov.au/scam-statistics

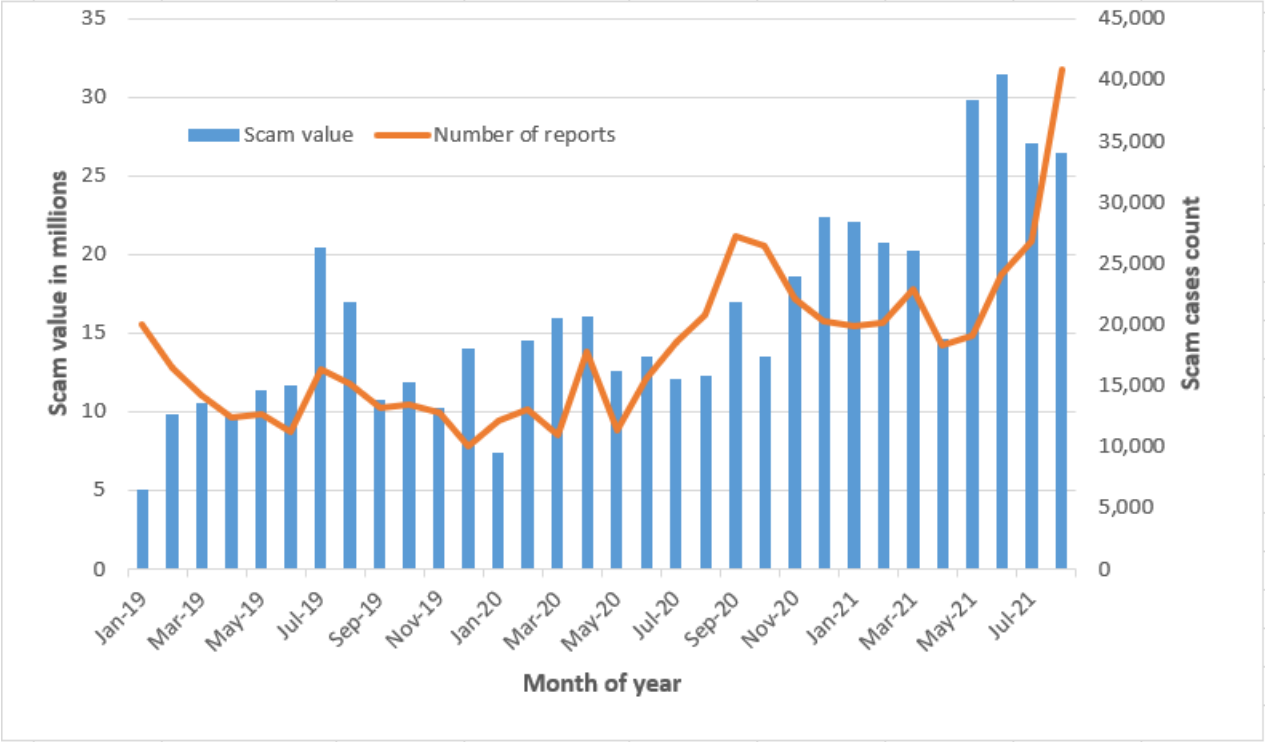

Figure 1.9: Scam statistics (reproduced from scamwatch [17])

the high yield investment program (HYIP) schemes. Many of the BGS instances promise a high return rate, which is one of the definitions used to characterize HYIP schemes [104, 153]. However, our analysis showed that the BGS instances do not pay back the victims, which is against the soul of HYIP attacks.

In this thesis, we report the first systematic study of BGS. We use our model to collect hundreds of BGS instances, interact with them, and collect thousands of cryptocurrencies addresses associated with the scam. We then use the collected data to give insight into this attack. Our results show that a small group of scammers controls the majority of the received funds. Our results also show that the scammers use different variations of the attack, such as asking the victims for mining fees, complete tasks for the scammer, and installing an executable file on their devices.

- **Few usable scam datasets.**. Although academia has proposed various anti-scam studies, only a few have published their datasets used in the experiments. Even for those published datasets, only scam URLs are provided, and these URLs are usually no longer available at the time of publication. This makes it difficult for subsequent studies to compare their methods with previous work. As a result, most anti-scam studies report perfect results under their dataset, making it hard to assess these methods' effectiveness.

In this thesis, we compile and publish large GHS and BGS datasets, including more than 65,000 GHS URLs and more than 1,200 BGS URLs, that are collected using our live crawler and from public sources, such as the Internet Archive[18]. In addition to the URL of scam attacks, we also provide the Document Object Model (DOM) of scam pages, allowing others to compare their methods to ours. Moreover, we publish more than 9,000 bitcoin addresses used in the BGS attacks, which is one of the biggest bitcoin scam databases published by academic research, if not the biggest one. Finally, we provide more than 180 scam addresses that belong to other cryptocurrencies, such as Ethereum, Litecoin, and Monero.

1.4 Contribution

In this thesis, our goal is to create an automatic data-driven system that can track web-based scam attacks, analyze their internal connections, and then provide an effective anti-scam solution. We have made the following contributions:

- **Cryptocurrencies emerging threats and defensive mechanisms: a systematic literature review.** In Chapter 2, we explore and aggregate the state of the art threats that have emerged with cryptocurrencies and the defensive mechanisms that have been proposed. We also discuss the threat types, scales, and how efficient the defensive mechanisms are in providing early detection and prevention. We also list out the resources used to collect datasets and identify the publicly available ones.

- **An automatic detection system and analysis of the Game Hack Scam.** Although recent researches have provided important insights into different types of scam, to our knowledge, GHS is not fully studied yet.

  In this thesis, we present and discuss a data-driven model to detect and analyze the game hack scam. Through our analysis, we have found that the scam pages are powered by templates providers and has been accessed by millions while not being adequately addressed and reported.

- **New research direction to detect Bitcoin web-based attacks.** State of the art in academic work on bitcoin scam detection is usually based on some manual collection of addresses involved in the scam. The starting point could be a manual search on a forum in which the attack is being discussed, e.g., bitcointalk.org [21], or it could be by a semi-automated crawl of that same forum, followed by manual

[18] https://web.archive.org/

addresses collection [147, 150, 154]. Furthermore, some researchers use "multiplier" techniques such as the multi-input heuristic clustering algorithm [127] to collect the bulk of addresses controlled by the same scammers [21]. Once scam addresses have been collected, their transaction histories are used to extract distinguishing features and tell benign addresses apart from scam addresses [21, 147, 149, 150, 154]. These features are then used to train a classifier [21, 147].

In this thesis, we do not base our analysis on previously reported campaigns only. Instead, we search for new, previously unreported instances. What is more, at this stage, we do not use existing transactions in the detection phase, which allows us to find addresses that do not have any payment yet. Following our approach, we have detected more than 70% of the current active scam addresses before receiving any transaction, which is impossible using traditional detection methods.

- **A data-driven model for web-based scam detection.** Although recent research has provided important insights into different scams, the main focus was on scam understanding, analysis, and detection. Researchers followed different ways to collect datasets, used to investigate scams or train classifiers to collect and identify additional scam instances. These studies expect previous knowledge and understanding of the scam, and they require extensive human intervention while creating and validating the dataset.

  In this thesis, we propose a data-driven model to detect and track web-based scams. Given a small number of scam samples, we infer search queries and use them on search engines and customized historical search engines to collect potential scam pages. We then cluster these pages to create a labeled training dataset with minimal human interaction.

1.4.1 Publications

We have published four papers out of this research:

- [18] Emad Badawi, Guy-Vincent Jourdan, Gregor Bochmann, Iosif-Viorel Onut, and Jason Flood. *The "game hack" scam.* In ICWE 2019. Springer LNCS 11496, pages 280–295, 2019.

- [17] Emad Badawi, Guy-Vincent Jourdan, Gregor Bochmann, and Iosif-Viorel Onut. *Automatic Detection and Analysis of the "Game Hack" Scam.* Journal of Web Engineering, 18(8), 2020.

- [16] Emad Badawi, Guy-Vincent Jourdan, Gregor Bochmann, and Iosif-Viorel Onut. *An automatic detection and analysis of the bitcoin generator scam*. In 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pages 407–416, Los Alamitos, CA, USA, sep 2020. IEEE Computer Society.
- [15] Emad Badawi and Guy-Vincent Jourdan. Cryptocurrencies Emerging Threats and Defensive Mechanisms: A Systematic Literature Review. IEEE Access, 8, 2020.

We have also submitted an additional journal paper and conference paper.

1.5 Organization

The rest of this thesis is structured as follows:

- In Chapter 2, we present an overview of several scam types focusing on web-based attacks. We also present the cybercrimes that target cryptocurrencies as a payment medium in the form of a systematic literature review.
- In Chapter 3, we use a data-driven model to investigate the GHS.
- In Chapter 4, we reuse our model and adapt it to give insight into the BGS.
- In Chapter 5, we generalize our model and leverage the expertise and insights gained from studying the BGS and GHS to increase automation and drastically reduce the manual efforts required to setup such a study.
- In Chapter 6, we conclude our work and present additional future work.

Chapter 2

Literature Review

2.1 Introduction

Fraud is the process of deceiving an individual or group for certain gains. Fraud has a broad range of wrongdoing and is not limited to financial fraud; for example, a miscreant can take advantage of marriage to earn citizenship in a desirable country. A type of fraud that generally involves money or business transaction is called a scam. In this section, we will briefly discuss some parts of the fraud history and its evolution over time with focusing on scam[1].

Fraud activities can be traced to ancient Egypt. Before 526 B.C., although Egypt was cashless, the rulers of Egypt still collected taxes in different ways, such as forced labor (called corvee) and traded goods. At that time, tax collectors manipulated the measures to weigh the grain to be taxed, which fooled the householders into overpaying their taxes. The tax collectors then skim the extra tax to their benefit [124].

Around 300 B.C. Hegestratos, a Greek merchant, committed insurance fraud when he purchased an insurance policy known as bottomry for his ship and the goods carried on it. Bottomry is a type of insurance where the lender receives their money back with interest in case the ship and cargo arrive at the destination port safely. On the other hand, the merchant would not be entitled to pay the loan back if the ship sank. At that time, Hegestratos was caught trying to sink his empty ship to keep the insurance loan and sell the goods for profit [143].

By early AD, money was used in many parts of the world, and fraud did not stop and was an issue in many aspects of life. At that time, Verres, the governor of Cilicia,

[1]In this thesis, we will use the words fraud and scam interchangeably through our analysis.

had committed fraud collection of tax, bribery, and theft until Cicero's Verrine Orations attested against him [22].

Not long after that, in 193 AD, the Praetorian Guards committed one of the biggest scams of ancient times when they sold the rights to the Roman throne with an enormous bid of 250 pieces of gold per soldier to Didius Julianus who used it to claim the emperor position. However, the throne was stolen after killing the Pertinax emperor, which caused the transaction to be invalid, and thus, Julianus couldn't claim the throne and get recognized as an emperor [11].

In 1699, the 20-year-old young artist Michelangelo started his career by forging an ancient sculpture and selling it to a cardinal Riario in the Catholic church [10].

In the late 17^{th} century, the Master of the Royal Mint, Sir Isaac Newton, investigated and fought against forgers, especially William Challoner. The investigation results have shown that Challoner had created wealth through theft, counterfeiting, and fraud [79]. At the end of his career, Sir Isaac was a victim of a market manipulation fraud when he invested in the South Sea Bubble of 1720. South Sea Company was formed in 1711 by John Blanc to solve Great Britain's massive debt. John convinced Great Britain debt owners to accept South Sea company shares instead of money by promising huge profits due to the monopoly the company had over trading in Central and South America and the South Seas. Many people purchased the company shares, and the share value skyrocketed from just 100 pounds to 1,050 pounds in less than a year. However, the company was selling dreams and did not end up doing much trading. Furthermore, John resorted to schemes that allow people to buy shares by only paying 20% upfront. Eventually, John ran out of tricks, and the bubble burst, leaving the first case of market manipulation in history behind [114, 119].

In the same period, Gregor Mcgregor, a Scottish General in the army, advertised the existence of a fictional island called "Poyais" in the Central American territory and attempted to draw settlers and investors to migrate to it. McGregor claimed to be the island crowned prince and started selling its lands, properties, and even a fabricated currency. However, when migrants traveled to that island, they found only an untouched jungle [67].

In the early 20^{th} century, the Ponzi scheme and advanced fee scams came to light. Ponzi scheme was originated by Charles Ponzi, who claimed to run an arbitrage where the investors were promised a 50% profit within 45 days, or 100% profit within 90 days [147]. The advanced-fee scam was started by P. Crentsil, who signed himself "P. Crentsil, Professor of Wonders" in 1920. Crentsil wrote and sent several letters to different people offering magical services in exchange for a fee. He was then caught and charged by the police [52]. Another type of scam that was popular at the same time is the real estate scam where

George C. Parker sold the Brooklyn Bridge to tourists up to twice a week in the 30 years leading up to 1928[2].

The audacity of con artists has risen over time. At the age of 15, Frank Abagnale Jr. had committed many types of fraud, including but are not limited to, forging and duplicating cheques, using false identities to open bank accounts, and posing as an airline pilot to get free flights. Furthermore, Abagnale disguised himself as security officers and convinced employees of car rental companies and airlines to hand over their cash deposits rather than putting them in the dropbox that he labeled as"Out of Service" [2, 131].

As technology advanced and played a part in banking and finance, fraudsters changed their attacks vectors and targeted a new audience. An early scam that used technology was the premium-rate telephone lines in the 1980s. Scammers tried to trick people into calling expensive premium-rate numbers or participating in games. Other victims started receiving phone calls from scammers indicating that they were a part of a lottery prize draw and won a big prize. However, the victim should pay the processing fees and requisite taxes before claiming the prize. Motivated by the considerable gain, many victims fell for the scam and transferred fees [110].

In the late 1990s, identity theft became widely used by fraudsters. The US public law identifies the act of identity theft fraudsters as "anyone who knowingly transfers or uses, without lawful authority, any name or number that may be used, alone or in conjunction with any other information, to identify a specific individual with the intent to commit, or to aid or abet, any unlawful activity that constitutes a violation of Federal law, or that constitutes a felony under any applicable State or local law" [111].

Around the same time, the Nigerian scam attacks became a known problem. In these scams, the victims are approached by an African prince, chief, or a wealthy businessman who asks for help in retrieving large sums of money which won't be accessible without their help. The scammer usually promises a share of this money that usually amounts to between 20%-30% of millions of USD [66]. This scam has started by postal mail, moved to fax, and ended up using email. This scam has many variations, including fake lottery and advance fees. The scammers gain profit by tricking the victim into paying a fee under the promise of a larger payoff in the future [74].

Another fraud that appeared in the 1990s was the plastic card fraud (credit/debit cards). Card fraud has many variations such as refund fraud which involves returning products ineligible for a refund to a warehouse or a merchant in exchange for money or other goods. The returned products may have been discarded damaged goods, or they may

[2] `https://www.newspapers.com/image/59901394/?terms=George%2BParker`

have been acquired illegally. For example, the fraudster may use a fake receipt to return a product picked up from a store shelf or return a stolen product to get cash [36].

In 2004, a data breach burst when an Engineer working at an online service provider company called AOL stole and sold information for 92 million accounts to spammers. This information was used in identity theft to take over existing accounts, open new accounts, and impersonate people for financial transactions[3]. In 2006, another major data breach incident happened to AOL when they publicly released search data of more than 65 thousand subscribers [25].

In the last decade, cryptocurrencies have become a rich environment for cybercrimes. Cybercriminals have leveraged bitcoin pseudo-anonymity to carry out a wide range of attacks vectors such as HYIP [21, 147, 153, 154], and Ransomware [26, 98, 132]. According to CipherTrace latest report report[4], the value of thefts, hacks, and scams has had a high increase in the recent years.

Fraudsters can abuse anything that can be used to create profit and wealth. This includes, but not limited to, Food fraud [86], healthcare and pandemic-related frauds [27,63, 108], citizenship and immigration scam [45], Internal Revenue Service (IRS) [24, 106, 151], telephone scam [151], cybercriminal activities that used the cryptocurrencies as a medium, web-based scams that provide fake and scam services through websites, etc....

In this chapter, we discuss the literature related to various web-based cybercriminal activities in detail. There is a significant body of academic work focusing on the detection and analysis of web-based attacks. These studies have provided valuable insights into different scams; to our knowledge, GHS and BGS are not thoroughly studied yet. The closest and most relevant studies to ours are studies about the so-called "Technical Support Scam" (TSS), online survey scams, and high-yield investment programs (HYIP) schemes. To give substance to this proposal, we discuss different types of web-based attacks and attacks targeting cryptocurrencies as a payment medium.

2.2 Cryptocurrencies Emerging Threats and Defensive Mechanisms: Systematic Review

In recent years, researchers have been actively working on analyzing the cyberattacks that utilized the cryptocurrencies as a payment medium. In this section, we present the lit-

[3] `https://www.nytimes.com/2004/06/23/technology/aol-engineer-sold-92-million-names-to-spammer-us-says.html`

[4] https://ciphertrace.com/cryptocurrency-crime-and-anti-money-laundering-report-august-2021/

erature related to these attacks in the form of a systematic literature review. We aim to complete the literature part of our thesis and provide researchers with an easily interpreted and comprehensive literature listing, which is the first step to develop more powerful defensive mechanisms against these attacks.

To this end, we present a summary of cybercriminal activities related to cryptocurrencies, and the scale of these crimes, as reported in the literature. We then analyze the detection methodologies proposed, the classifiers used in the process, and how effective these methodologies are. Furthermore, we summarize the sources that can be utilized to collect datasets for cryptocurrency research purposes. Finally, we list the datasets that have been publicly disclosed, as well as listing some useful tools and resources used to collect and analyze this type of data. For this purpose, we worked on answering the following research questions:

RQ1: With the introduction of cryptocurrencies, what are the types and scales of cybercriminal activities reported by researchers?

RQ2: What are the proposed defensive mechanisms available to detect cybercriminal activities, and what is the reported effectiveness of these mechanisms?

RQ3: For cryptocurrency cybercrimes detection and prevention, what are the public datasets provided in the literature, and how have these datasets been collected?

Our analysis shows that in the last four years only, 68 papers studying these attacks were published. Moreover, many of these publications proposed defensive mechanisms. Overall, our analysis includes 94 papers. Figure 2.1 shows the number of articles published per year on the subject of attacks that use cryptocurrency as a payment medium. Our review contains only 10 papers published in 2021. However, we believe that the number of papers published in 2021 will significantly increase before the end of the year. Based on our results when updating our review over time, many papers become reachable on the search databases in the last few months of the year. For example, around 50% of the papers published in 2020 and included in this review were found when we updated our records to include the last 3 months of the year 2020.

A summary of the papers is shown in Table 2.1. The table contains basic information about the papers we used in our analysis. In particular, we present the publication year, the publication location (Conference or Journal), the cybercrime type in the discussion, and the targeted cryptocurrency.

We present the review protocol of the systematic literature review in Appendix A.

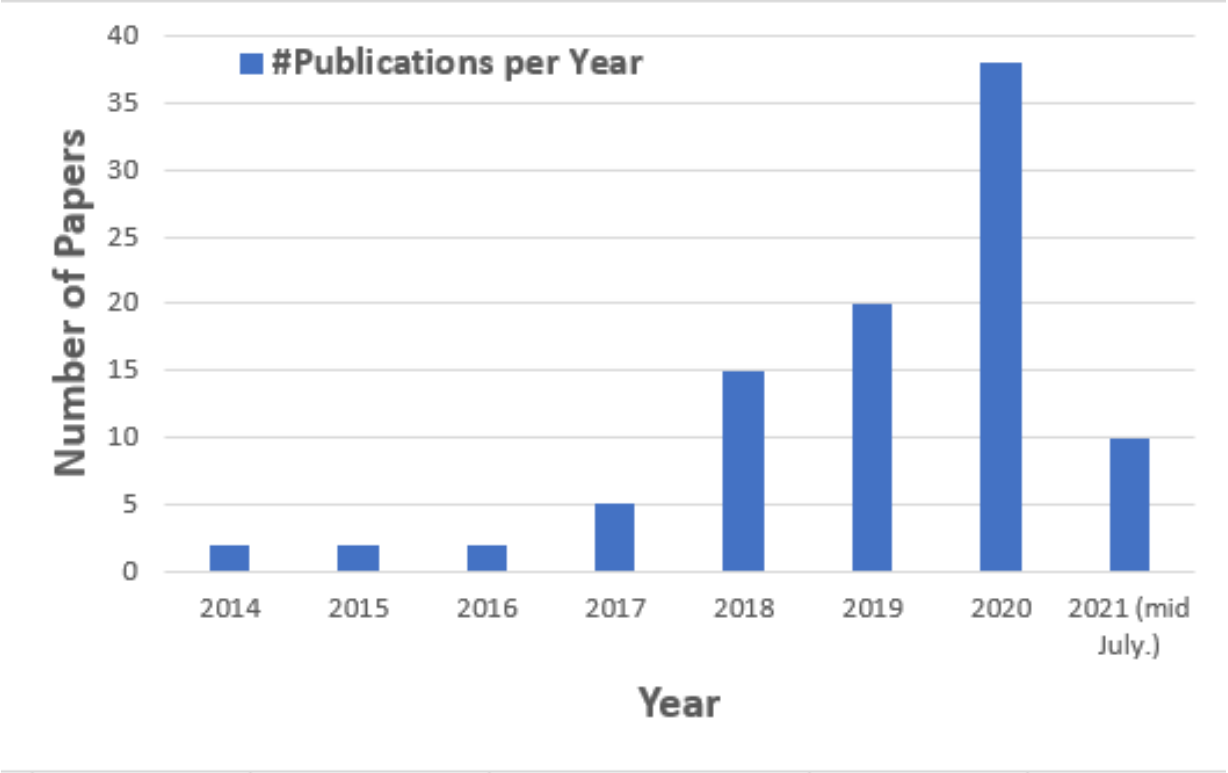

Figure 2.1: Number of cryptocurrency attack papers published per year.

Table 2.1: Summary of the papers included in the SLR.

	Reference	Published Year	Journal/ Conference	Crime Type (D1)	Currency (D3)
1	[30]	2014	Journal	ML	Bitcoin
2	[155]	2014	Journal	DDoS	Bitcoin
3	[29]	2015	Conference	ML	Bitcoin
4	[153]	2015	Conference	Services detection	Bitcoin
5	[98]	2016	Conference	Ransom	Bitcoin
6	[128]	2016	Journal	ML	Bitcoin
7	[168]	2017	Conference	Services detection	Bitcoin
8	[149]	2017	Conference	HYIP	Bitcoin
9	[138]	2017	Conference	Mining/Jacking	Multiple
10	[92]	2017	Journal	Ransom	Bitcoin
11	[57]	2017	Journal	DDoS	Bitcoin
12	[150]	2018	Conference	Services detection	Bitcoin

(Continued on next page)

	Reference	Published Year	Journal/ Conference	Crime Type (D1)	Currency (D3)
13	[70]	2018	Conference	Ransom	Bitcoin
14	[100]	2018	Conference	ML	Bitcoin
15	[21]	2018	Conference	HYIP	Bitcoin
16	[130]	2018	Conference	ML	Bitcoin
17	[68]	2018	Conference	Phishing	Bitcoin
18	[26]	2018	Conference	Ransom	Bitcoin
19	[82]	2018	Journal	P&D	Multiple
20	[69]	2018	Conference	Mining/Jacking	Monero
21	[40]	2018	Conference	HYIP	Ethereum
22	[33]	2018	Journal	ML	Bitcoin
23	[44]	2018	Journal	Ransom	Bitcoin
24	[90]	2018	Conference	Mining/Jacking	Monero
25	[154]	2018	Conference	HYIP	Bitcoin
26	[148]	2018	Conference	HYIP	Bitcoin
27	[101]	2019	Conference	ML	Bitcoin
28	[80]	2019	Conference	HYIP	Ethereum
29	[41]	2019	Journal	HYIP	Ethereum
30	[147]	2019	Journal	HYIP	Bitcoin
31	[126]	2019	Conference	Mining/Jacking	Multiple
32	[58]	2019	Journal	Mining/Jacking	Multiple
33	[146]	2019	Conference	Honeypot	Ethereum
34	[117]	2019	Conference	General	Ethereum
35	[135]	2019	Conference	Mining/Jacking	Multiple
36	[175]	2019	Journal	Mining/Jacking	Monero
37	[39]	2019	Conference	P&D	Bitcoin
38	[87]	2019	Conference	Mining/Jacking	Monero, JSECoin
39	[107]	2019	Conference	Mining/Jacking	Monero
40	[7]	2019	Journal	ML	Multiple
41	[113]	2019	Conference	Mining/Jacking	Monero
42	[156]	2019	Conference	P&D	Multiple
43	[167]	2019	Conference	P&D	Multiple
44	[174]	2019	Conference	Mining/Jacking	Multiple

(Continued on next page)

	Reference	Published Year	Journal/ Conference	Crime Type (D1)	Currency (D3)
45	[3]	2019	Conference	DDoS	Bitcoin
46	[19]	2019	Conference	DDoS	Bitcoin
47	[20]	2020	Journal	HYIP	Ethereum
48	[62]	2020	Journal	ML	Multiple
49	[56]	2020	Journal	General	Ethereum
50	[6]	2020	Conference	ML	Bitcoin
51	[5]	2020	Conference	ML	Bitcoin
52	[71]	2020	Conference	General	EOS
53	[48]	2020	Journal	ML/ransom	Bitcoin
54	[23]	2020	Conference	Mining/Jacking	Multiple
55	[139]	2020	Conference	Mining/Jacking	Multiple
56	[51]	2020	Journal	Ransom	Ethereum
57	[145]	2020	Conference	honeypot	Ethereum
58	[84]	2020	Conference	Ransom	Ethereum
59	[46]	2020	Conference	ML	Bitcoin
60	[122]	2020	Conference	General	Bitcoin, Ethereum
61	[61]	2020	Journal	Ransom	Bitcoin
62	[158]	2020	Journal	Phishing	Multiple
63	[142]	2020	Conference	ML	Multiple
64	[123]	2020	Conference	ML	Multiple
65	[93]	2020	Journal	General	ETH
66	[141]	2020	Journal	ML	Bitcoin
67	[4]	2020	Conference	Ransom	Bitcoin
68	[83]	2020	Conference	Ransom	Bitcoin
69	[49]	2020	Journal	Cryptojacking	General
70	[50]	2020	Conference	Cryptojacking	General
71	[169]	2020	Conference	Cryptojacking	General
72	[60]	2020	Conference	Cryptojacking	General
73	[59]	2020	Conference	Cryptojacking	General
74	[94]	2020	Conference	P&D	Bitcoin
75	[170]	2020	Conference	Phishing	Ethereum
76	[171]	2020	Conference	Phishing	Ethereum

(Continued on next page)

	Reference	Published Year	Journal/ Conference	Crime Type (D1)	Currency (D3)
77	[38]	2020	Journal	Phishing	Ethereum
78	[37]	2020	Journal	Phishing	Ethereum
79	[28]	2020	Conference	HYIP	Bitcoin
80	[121]	2020	Conference	HYIP	Ethereum
81	[54]	2020	Conference	HYIP	Ethereum
82	[99]	2020	Conference	HYIP	Ethereum
83	[55]	2020	Conference	HYIP	Ethereum
84	[140]	2020	Conference	Cryptojacking	General
85	[152]	2021	Conference	Ransom	Bitcoin
86	[97]	2021	Journal	Ransom	Bitcoin
87	[85]	2021	Journal	Cryptojacking	General
88	[35]	2021	Journal	Cryptojacking	Bitcoin, Monero, and Bytecoin
89	[64]	2021	Journal	P&D	General
90	[112]	2021	Journal	P&D	General
91	[160]	2021	Conference	Phishing	Ethereum
92	[53]	2021	Journal	HYIP	Ethereum
93	[72]	2021	Conference	HYIP	Ethereum
94	[42]	2021	Journal	HYIP	Ethereum

2.2.1 With the introduction of cryptocurrencies, what are the types and scales of cybercriminal activities reported by researchers? [Rq.1]

Cyberattacks related to cryptocurrencies

Several attacks that use cryptocurrencies as a payment medium, such as "high yield investment programs" (HYIP), ransomware, and money laundering (ML), have been studied in the literature. Figure 2.2 shows the breakdown of these attacks and the number of articles that cover each attack. Some papers cover several attacks, so the sum does not add up to the number of papers in our study.

As shown in Figure 2.3, the majority of the attacks studied in the literature target Bitcoin, Ethereum, and Monero. As of November 6^{th}, 2021, Bitcoin and Ethereum have the

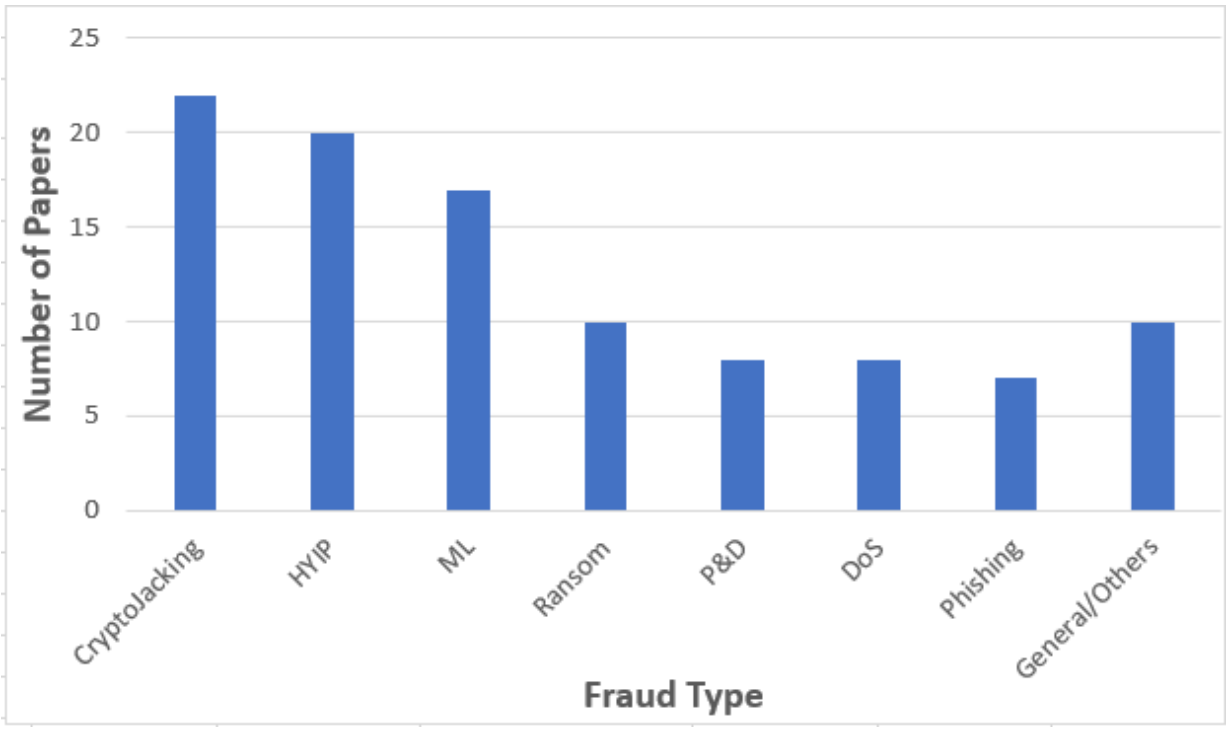

Figure 2.2: Number of published articles per cyberattack type.

highest capitalization market of approximately $1,156 and $532 billion USD, respectively[5]. The market capitalization of Monero is currently much smaller, just above a billion USD. It is, however, widely used in so-called "Cryptojacking attacks" because Monero is specifically designed to not give advantage to application-specific integrated circuit mining. Therefore, any computing device has a fair chance at establishing proof-of-work, and thus hijacking average computers for mining Monero can be profitable. In contrast, the same attack on, e.g. Bitcoin, has little chance of generating any revenues at all. As a result, cryptojacking can be found on any Internet-connected device with a CPU, such as mobiles, PCs, and the Internet of Things [87, 175].

The scale of the cyberattacks

According to Kshetri and Voas [92], the denial of services and productivity losses due to ransom attacks are in billions of USD. Furthermore, by applying their classification model on features extracted from the transactions of 100K unclassified Bitcoin addresses, Yin and Vatrapu [168] estimate that 10.95% to 29.81% of the Bitcoin addresses are involved in cybercrime activities. These addresses are involved in transactions classified into five different cybercrimes: mixing, ransomware, scam, stolen-bitcoins, and tor-market.

[5] `https://coinmarketcap.com/`

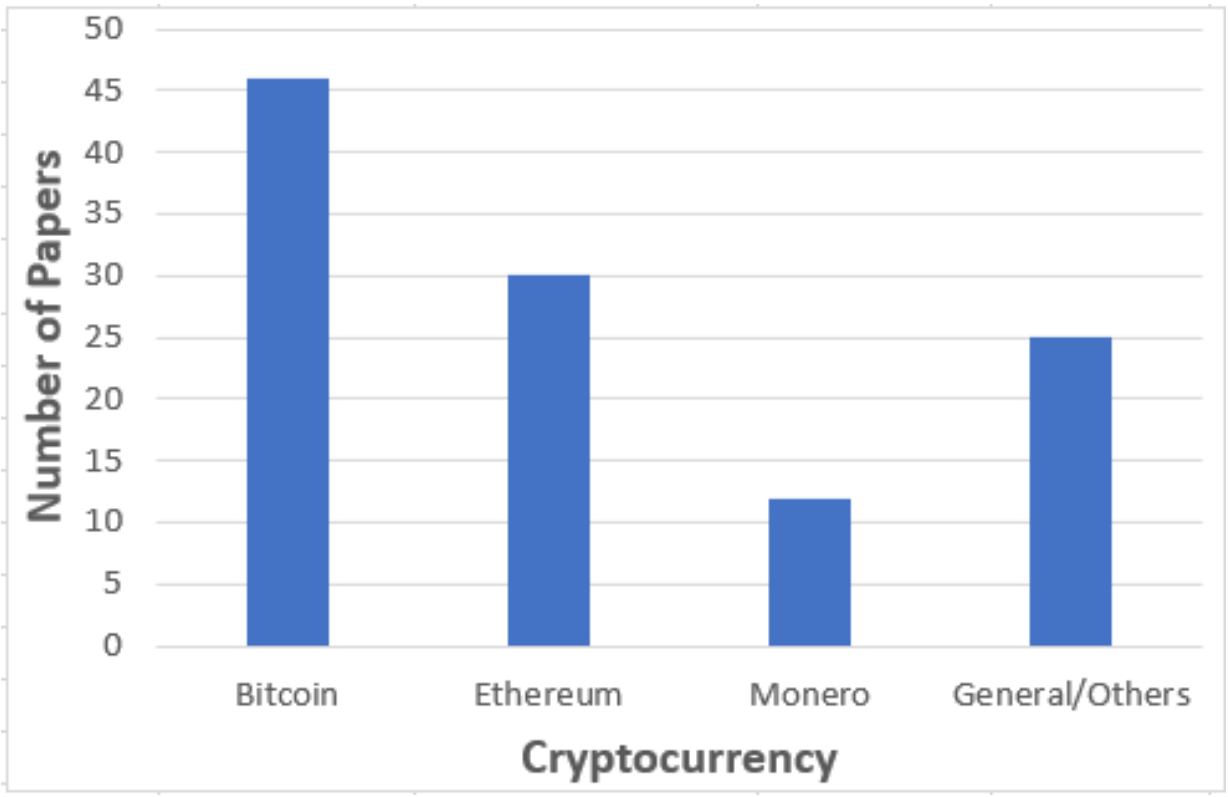

Figure 2.3: Number of published papers per currency.

Several datasets and scale measurement techniques were utilized to analyze the fraud activities scale in the literature, including:

- One of the most common scale measurement techniques is estimating the value of stolen money by analyzing the blockchain transaction history of the collected cybercrime addresses; such as in the case of the crimes targeting Bitcoin and Ethereum currencies [20, 21, 26, 44, 68, 70, 98, 146, 153].
- In the case of P&D schemes, the authors inferred an estimation of the theoretical maximum possible profit based on the average P&D events per day and the currency price variation during the P&D event [39, 82, 156, 167].
- With the high privacy provided by Monero, and with no public available transaction history, the researchers inferred an estimation of cryptojacking attack scale by applying mathematical analysis on information extracted from the cryptojacking campaigns such as the number of visits, the visit duration, the hardware resources usage, CPU utilization, and the number of sites in each scam campaign [69, 87, 90, 107, 135]. For example, Hong *et al.* [69] used the following formula to measure the profit of each cryptojacking campaign.

$$\sum \frac{\#Visitors \times Duration \times HashSpeed}{Difficulty} \times Reward$$

Where $\#Visitors$ is the number of visitors (in millions per month), $Duration$ is the average length of time (in second) a user stays on the site, $HashSpeed$ is "the average hashing speed of users' processors" [69], $Difficulty$ is the current hardness of the proof of work, and $Reward$ is the block reward at the time of analysis.

- Other researchers provided an estimation based on extrapolating the results of their classification model or by applying the classifier on an unknown dataset [40, 41, 168]. For example, Yin and Vatrapu [168] reported the results of applying their classification model on 100k unclassified addresses while Chen *et al.* [40, 41] used the classification model precision and recall values to estimate the scale of smart Ponzi schemes on Ethereum.

In this section, we report the attacks with the highest number of victims and the ones with the highest profit for each type of crime[6]. We provide the full scale as reported in the literature in Table B.2 in Appendix B.

As researchers conducted different studies and analyses in the literature, the scales of the cybercrime activities were reported in many ways, even for the same cybercrime and the same cryptocurrency. A breakdown of the scale of these activities addressed in the literature is:

1. **HYIP (Bitcoin):** The Pirate@40's HYIP scheme had raised 700,000 Bitcoin from the investors before they were charged by the Security and Exchange Commission (SEC) in 2013 [148]. Bartoletti *et al.* reported an estimate of $10 million USD in [21], and Vasek and Moore [154] reported that 11,990 users have responded to 1,780 different scams on the bitcointalk forum.

2. **HYIP (Ethereum):** In the litterature, we find that 0.03% [41] to 0.15% [40] of the smart contracts are HYIP. In [20], the authors estimated the value of HYIP with Ethereum is approximately half a million USD.

3. **Phishing:** Holub and O'Connor reported that $50 million USD were stolen by the attackers in 3 years [68].

4. **Ransom:** The scale of the ransomware was reported as the payment values received by the attackers. Conti *et al.* [92], reported a ransom payment of 7,059.9 Bitcoin (∼

[6]The full raw data is available on our public repository **http://ssrg.site.uottawa.ca/slr/**

$2.8 million USD), Liao *et al.* [98] reported payments of 1,128.40 Bitcoin ($\sim$ $310,000 USD) over a 5 months period, and Huang *et al.* [70] reported the highest ransom value, $16 million USD paid by 19,750 victims. However, the main monetary loss due to ransom is the denial of services and productivity losses, which are estimated in billions of USD from about 300,000 infected computers in 150 countries [92].

5. **P&D**: In the literature, it is estimated that on average, 1.6 [82] to 2 [156] P&D events are organized per day. Xu and Livshits [167] estimated that P&D events generate an aggregate, artificial trading volume of $6 million USD a month. In [39], Chen *et al.* analyzed a leaked transaction history of the Mt. Gox Bitcoin Exchange from April 2011 to November 2013, and reported that the transactions with an abnormal price involved 13.09% of the users in the dataset.

6. **Mining/Cryptojacking:** As transactions history can not be accessed with monero, the scale of cryptojacking has been estimated using the CPU usage consumed by the mining scripts and the campaigns size. Zimba *et al.* [175] estimated that 32% of the users in the US are exposed to browser-based crypto mining. Additionally, Hong *et al.* [69] estimated that 10 million web users are affected by cryptojacking monthly, at a daily cost of $59,000 USD due to 278K kWh of extra power consumption. Furthermore, in [90] the profit of each cryptojacking campaign is estimated at $14.36 USD to $31,060.80 USD per month on average, while in [107] it was estimated at $340 USD per campaign per day (about $10,200 USD per month).

2.2.2 For cryptocurrency cybercrimes detection and prevention, what are the public datasets provided in the literature, and how have these datasets been collected? [Rq.3]

In this section, we present the resources used in the literature to collect datasets on which to train, detect and analyze the attacks discussed in each papers. Some researchers collected the training data manually, e.g. by searching online fora such as `bitcointalk.org` [21]. Other researchers used a semi-automated crawling process followed by manual data collection [147, 150, 154]. Furthermore, some datasets were collected by extracting the system resource usage data of the devices under attack [58, 69, 113]. Our analysis shows that four different resources were used to prepare the training dataset:

1. Collecting data from online fora and blogs, such as bitcointalk.org and Reddit [21, 28, 44, 46, 57, 70, 98, 147–150, 153–155, 167]. The researchers relied on crawling these fora as they are used by scammers to advertise for their schemes. For example,

Vasek and Moore [154] crawled the entire history (from June 2011 to November 2016) of the bitcointalk.org subforums that scammers use to advertise Ponzi schemes. Their crawling returned 11,424 threads, which they further refined to 2,617 threads by removing threads discussing online card games and only including threads that contain URLs or bitcoin address for the scam. In [44], the authors collected the scam addresses manually, by searching online ransomware knowledge base (such as Kaspersky Lab, ESET, Symantec, and Malwarebytes), ransomware removal guides (such as MalwareTips.com, BleepingComputer.com, and 2-spyware.com), online fora where researchers and victims publish their data (such as Reddit), and available ransomware screenshots in different search engines image databases (such as Yahoo and Google).

2. Using dataset provided by third parties including previous studies, `chainalysis.com`, and public blacklists [5, 6, 19, 26, 37–40, 42, 53, 54, 56, 61, 68, 71, 80, 93, 97, 99, 107, 112, 117, 121, 126, 139, 140, 145, 146, 160, 168, 170, 171]. For example, Chen *et al.* [39] used a leaked transaction history of Mt. Gox Bitcoin Exchange from April 2011 to November 2013 in their analysis. Chen *et al.* [40] and Jung *et al.* [80] used the dataset from Bartoletti *et al.* [20] in their study. Ostapowicz and Zbikowski [117] used the wallets reported in *Etherscan.io* as being used in fraud activities .

3. Collecting data from free online sources, online exchanges, Telegram groups, and smart contracts with public source code. These sources require manual analysis to distinguish between scam and benign data [3, 4, 20, 23, 41, 64, 69, 82, 87, 94, 122, 135, 152, 156]. For example, Kamps and Kleinberg [82] used the CCXT python library to collect cryptocurrency market data, from April 2018 to May 2018, from a variety of cryptocurrency exchanges, including Binance, Bittrex, Kraken, Kucoin, and Lbank. Victor and Hagemann [156] collected the price and volume of cryptocurrencies from Binance exchange, the chat histories from Telegram P&D groups (fraud ads), and general data about the currency capitalization from *coinmarketcap.com*.

4. Collecting system resources, such as system runtime parameters [35, 49, 58, 59, 69, 85, 113, 169]. For example, Ning *et al.* [113] collected 12 system runtime parameters (such as interrupts per second, page reads/write/fault per second, and packets received/sent per second) from 13 different devices while running 5 different applications on each device.

Further breakdown is provided in Table B.3 in Appendix B.

In some activities such as HYIP, authors had difficulties collecting a large number of addresses. In most cases, they manually visited online fora to collect scam addresses ad-

vertised by the scammers. However, in many instances the addresses were not included in the posts. In such cases, the authors visited the HYIP website and manually extracted the deposit address. When the websites were no longer online, the researchers tried to recover old snapshots through the Internet Archive [21]. To increase the number of collected addresses, some authors are using "multiplier" techniques. We have found two such techniques used in the literature:

- **Multi-input heuristic:** in this heuristic, the assumption is that the same person owns all the addresses on the input side of any transactions [21, 44, 70, 98, 147–150].
- **Shadow/change address algorithm:** in this heuristic, the assumption is that if there are only two addresses in the output side of any transactions, and one address has appeared before in the blockchain while the other address has not been used before, then it can be safely assumed that the new address is a shadow/change address generated to accept the change from the transaction back to the sender, and thus is owned by the sender [44, 98, 147].

Some authors have disclosed their datasets, which, in turn, provides an opportunity for other researchers to use them. In fact, the dataset prepared by Bartoletti *et al.* [20] was later used in [37, 40, 42, 53–55, 80] to implement defensive mechanisms against the cybercriminal activities that utilize cryptocurrencies as a payment medium. The full list of disclosed datasets in the literature is presented in Table 2.2. If a dataset is used in more than one research paper, we only show the most recent publication year in the table. Furthermore, Table 2.4 contains other supplementary websites and tools that can be used in the data collection and analysis process.

2.2.3 What are the proposed defensive mechanisms available to detect cybercriminal activities, and what is the reported effectiveness of these mechanisms? [Rq.2]

State of the art defensive mechanisms reported in the literature are usually based on extracting distinguishing features from the training dataset and using these features to train a classifier such as random forest (RF) [4,21,35,41,59,80,94,117,138,147,149,150,167], XGBoost [39, 40, 55, 117, 168] or support vector machine (SVM) [59, 72, 87, 117, 170] to tell benign data apart from cybercrime data. The features are based on the type of cybercrime being discussed as well as on the type of the available dataset. For example, to detect HYIP schemes in Bitcoin and Ethereum, publicly readable blockchain transaction records

Dataset (D5)	Description	Reference	Published Year
https://bitbucket.org/mhuzai/mineguard/src/master/	The data and code used in the paper.	[138]	2017
https://goo.gl/sQJKdx	List of Bitcoin addresses categorized per the service they are used in.	[150]	2018
https://goo.gl/ToCho7	List of scam Bitcoin addresses.	[21]	2018
https://osf.io/827wd/	The data and code used in the paper.	[82]	2018
https://github.com/deluser8/cmtracker	The data and code used in the paper.	[69]	2018
https://www.walletexplorer.com/wallet/Btcst.com-pirateat40/addresses	Btcst.com-pirateat40 Bitcoin addresses.	[148]	2018
https://github.com/teamnsrg/outguard	The data and code used in the paper.	[87]	2019
https://github.com/hoshsadiq/adblock-nocoin-list	NoCoin adblock list. Block "browser-based crypto mining".	[126]	2019
https://goo.gl/k5PCOZ	List of scam domains and Bitcoin addresses.	[147, 149]	2019
https://github.com/pan-unit42/iocs/blob/master/6908_of_8712_coin_mining_urls_in_pandb.txt	List of URLs contains scam coin mining code.	[135]	2019
https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/25541	142 distinct DDoS attack reports on 40 Bitcoin services	[19, 155]	2019
https://github.com/pan-unit42/iocs/blob/master/4457_of_4633_scam_js_urls_in_pandb.txt	List of URLs contains scam JS code.	[135]	2019
https://github.com/blockchain-unica/ethereum-ponzi	The data and code used in the paper.	[20]	2020
https://goo.gl/CvdxBp	List of scam Bitcoin addresses.	[20, 40, 80]	2020
Hardcoded in the paper	List of Bitcoin addresses, mining script URLs, online mixers, etc...	[26, 44, 46, 107, 146]	2020
https://bit.ly/32pmC2A	Dataset and code used in the paper	[56]	2020

Table 2.2: Publicly available data provided in the literature.

URL (D5)	Description	Reference
https://github.com/bitcoinponzi	A public tool provided by the author for features extractions from the transactions history	[21]
https://github.com/ccxt/ccxt	CCXT python library to collect data from variety of cryptocurrency exchanges.	[82]
https://bitcointalk.org/index.php?topic=75883.0	List of clean gampling domains.	[154]
https://go.aws/2R1Jktx	Alexa top 1M domains list.	[107]
https://coinmarketcap.com/	Market Capitalization related data.	[107]
https://chromedevtools.github.io/devtools-protocol/	Allows for tools to instrument, inspect, debug and profile Chromium, Chrome and other Blink-based browsers.	[107]
https://bit.ly/2TpkEMh	Binance API.	[156]
https://github.com/LonamiWebs/Telethon	Telegram's API.	[156]
https://etherscamdb.info/scams	Scam DB.	[56, 170]
https://bit.ly/3ouSZrw	Bot index (list of bots in ESO).	[71]
https://bit.ly/38PButR	Scam domains/addresses dataste provided by CryptoScamDB.org.	[122]
https://urlscan.io/	Online service that scans and analyze websites.	[122]
https://www.malware-traffic-analysis.net	Malware traffic analysis.	[61]
https://virusshare.com/	Malware samples.	[61]
https://bit.ly/3kxpDpP	Malware repository.	[61]
https://github.com/twintproject/twint	Twitter intelligence tTool to scrape tweets.	[3]

Table 2.4: Useful resources provided in the literature.

Sources to extract features from (D2)	**Examples of extracted features (D11)**	**Reference**
The HTML code	Used global variable, WebSocket messages, and alert text.	[69, 90, 135]
The web technology and generated traffic	The use of proxies, number of packets per minute, and the IP addresses of the cryptomining domains.	[35,126,169,174, 175]
The system resources.	(Interrupts,page read-/write)/second, parallel tasks, and L1-dcache-loads.	[23,50,58–61,85, 87, 107, 113, 138–140]
The cryptocurrency addresses transaction history	The frequency of transactions, the ratio of in/out transactions, and the lifetime of the address.	[4–6, 19, 21, 37, 71, 93, 117, 140, 147,149,150,160, 168]
The cryptocurrency market movement (price, volume, etc.)	Market capitalizations, Volumes in coin, Entropy, and stability.	[39, 82, 94, 112, 156, 167]
The Ethereum account and smart contract code	Number of in/out transactions, existence of an unconditional jump, and the frequency of all the opcodes used in the smart contracts	[40,41,49,56,72, 80, 146].

Table 2.5: Utilized sources for extracting detection features.

and smart contract code are leveraged [41,80,153,154]. In Table 2.5, we present six different types of sources that are used in the literature to extract the features and examples of these features for each source type[6].

The effectiveness of the proposed detection mechanisms varies from a 0-day detection model, in which the scam is detected as soon as it is posted [80], to models that require the attack to have victims as these mechanisms depend on extracting features from the scam transactions history [147]. Several measurement metrics were used in the literature to report how successful the proposed mechanisms were. The measurements most often used are:

- **True Positive (TP)**, the number of cybercrime instances that were successfully identified as cybercrime.
- **False Positive (FP)**, the number of cybercrime instances that were wrongly iden-

tified as benign.

- **Precision**, the ratio of actual cybercrime instances to all those classified as cybercrime.
- **Recall**, the ratio of correctly classified cybercrime instances to all cybercrime instances in the training set.

In this section, we report the breakdown of the mechanisms with highest detection rate per crime type as follows[6]:

- **HYIP (Bitcoin):** a TPR of 95% and an FPR of 4.9% was reported by Toyoda *et al.* [147], and Bartoletti *et al.* [21] proposed a detection mechanism with 96.8% TPR and a recall of 96.9%. However, the proposed mechanisms for Bitcoin HYIP detection do not provide early detection and defensive models. They depend on classifying previously reported scam campaigns to extract features from the transactions history of the addresses.
- The proposed detection mechanisms fairs better with **HYIP (Ethereum) schemes** detection. 0-day detection models that can detect HYIP schemes in smart contracts at the moment of creation were proposed in [40, 41, 80]. For example, the model proposed by Jung *et al.* [80] reports a precision of 0.99 and a recall of 0.97 for full data analysis and a precision of 0.98 and recall of 0.96 for 0-day detection.
- **P&D** detection mechanisms depend on features extracted from the market movement such as market capitalization, the price, and the volume. As a result, it was possible to develop a model that predicts the likelihood of a cryptocurrency being pumped before the actual pump event [167].
- **Cryptojacking** detection methodologies achieved a high detection rate of 99.7% [138], and TPR of more than 99% [87,113,138,174]. Furthermore, the detection model proposed by Ning *et al.* [113] detects 87% of the mining scripts "instantly[7]", and detects 99% of the scripts within a window of 11 seconds.

A breakdown of the classifiers used by the papers and the results achieved is presented in Table B.1 in Appendix B

[7]The authors do not explain what "instantly" means in that context.

2.2.4 SLR Conclusions

In the systematic literature review, we identified 94 research articles discussing cybercriminal activities that used the cryptocurrencies. We analyzed the papers and offered a broad perspective on the activities type, scale, and the proposed detection mechanisms. Our analysis concludes that a significant amount of research has been carried out to detect and analyze these cyberattacks. The research articles have discussed several attacks, including high yield investment programs (HYIP), ransomware, pump and dump, money laundering, and cryptojacking. The cryptocurrencies most frequently studied in the literature are Bitcoin, Ethereum and Monero. These cyberattacks have stolen millions of USD from thousands of victims. Furthermore, millions of connected devices are abused in cryptojacking attacks. However, even greater losses are caused by ransomware denial of services and productivity losses, which are estimated in billions of USD.

We have found that in the literature, four different sources have been used to collect training datasets; some are scrapping online fora, some are using data from third parties, some are using free online sources, and finally some are using usage data of the devices under attack. Many authors have made their dataset publicly available, and we have provided a complete listing of all these datasets.

Finally, the defensive mechanisms that have been suggested in the literature relied on training classifiers such as "random forest" and "support vector machine" on distinguishing features extracted from the dataset. Our review revealed that the proposed defensive mechanisms were quite efficient with 0-day detection of HYIP in Ethereum and cryptojacking, and predicting the likelihood of a cryptocurrency being pumped before the actual pump event. However, although high accuracy late detection of HYIP in Bitcoin have been published, 0-day detection of this attack in Bitcoin is an open problem.

2.3 Survey Scam

In this section, we look into the survey scam, in which the victims are tricked into providing sensitive information and installing malware and unwanted programs. Usually, this happens while asking the victims to complete some surveys in exchange for some expected awards or offers. A variety of awards and offers are advertised, for example, free software's, gifts, as well as gift cards for different stores such as Amazon and Costco [43, 88]. Several

security companies have published reports about survey scams[8][9][10][11].

The first analysis of survey scam was presented by Clark *et al* [43], who looked into 388 Facebook spam URLs. The authors identified 283 (73% of the spam URLs) survey scam and found that they are monetized via online ad networks. They crawled these URLs and manually inspected the captured screenshots to identify to which ad networks each URL belongs. Their analysis showed that over 50% of the survey scam URLs are sponsored by four ad networks: CPAlead, Amung.us, LifeStreet Media, and ClickBanner. They concluded that intervening on these four ad networks, which are sponsoring the majority of Facebook spam seen in their study, can demonetize the spam ecosystem.

In [88], Kharraz *et al.* proposed Surveylance, an automated tool to detect survey scam websites. Using Surveylance, the authors have performed the first systematic analysis on survey scam. By deploying Surveylance for several months to crawl the web and detect online survey scams, they have discovered more than 8k survey websites that directed the victims to more than 300k online survey pages. The authors have reported that these survey websites fingerprint the victims' data to prompt customized messages and dynamically select offers based on their location. They also showed that attackers aim to deliver malware and potentially unwanted programs as well as steal sensitive information from victims, such as credit card numbers and Social Security Numbers.

Finally, Subramani *et al.* [137] presented PushAdMiner, an automated system to collect and analyze online ads delivered via web push notifications (WPNs). By deploying their system on desktop and mobile devices, PushAdMiner has crawled thousands of websites and collected more than 21k WPN messages. The authors have identified 5,143 WPN-based ads belonging to 572 different ad campaigns and found that 51% (318 (out of 572)) of the campaigns are malicious. They further reported that the existing ad blockers and URL filters are inadequate in stopping these WPNs.

2.4 Technical Support Scam

Another type of web-based attacks is the technical support scam (TSS). In TSS, scammers combine online scam and telephone fraud activities to convince their victims that their machines are infected with malware, and offer a fake technical support service. TSS awareness began to appear in 2008 as people were receiving cold calls proclaiming the

[8] `https://symc.ly/2ADviGF`

[9] `https://malwaretips.com/blogs/remove-2017-annual-visitor-survey-popups/`

[10] `https://malwaretips.com/blogs/remove-chrome-opinion-survey-popup/`

[11] `http://symc.ly/2ESLmbC`

infection of their devices and that immediate action is required. The scammer claims to be a tech support employee at one of the big companies, such as Microsoft or Dell, and offer a replacement for the victim's inadequate anti-virus. Furthermore, the scammers offer a specialist service to follow-up with the victim throughout the anti-virus installation process in exchange for a fee [65, 134].

In recent years, TSS has evolved to be a more sophisticated online scam and gets the victims to initiate the call with the scammer [134]. TSS websites are carefully designed web pages that attempt to convey the scammer trustworthiness by abusing popular software and security companies' trademarks and logos [103]. The scam starts when a victim lands on a page that resembles system error messages and claims that his/her device is infected with malware [103]. The attacker then asks the victim to call the help center through a toll-free number to get help for the infection. To make it hard for the victim to ignore these pages, the attacker uses intrusive JavaScript techniques, such as continually asking the victim to call the technical support number through displaying alert boxes [103]. In some cases, these pages are supported with scary audio messages or count-down clocks to add a sense of urgency[12]. After a victim calls the scammer, he/she is instructed to allow the technician to take full control of the device by installing remote desktop software. The scammer then tries to convince the victim his/her device is infected by showing typical system errors. Once a victim is duped, the scammer will offer a fix in exchange for a fee that costs hundreds of dollars typically [103]. To process the payment, the scammer asks for the victim's credit card number. However, in case a victim refuses to pay, the scammer may remotely set passwords to lock users out of their machines [95, 125]. In the worst-case scenario, the scammer will keep billing the victim's credit card or steal private and financial information by installing keystroke loggers or malware on his/her machine [95, 103, 125].

Miramirkhani *et al.* presented the first systematic TSS study [103]. The authors have proposed ROBOVIC "Robotic Victim", an automated tool for detecting and recording TSS instances, and used it to collect a big corpus of scams over a 250-day collection period. ROBOVIC has recorded more than 22k scam-related URLs mapped onto more than 9k unique domains. They used these instances to give insights on the prevalence of the scams, the scam ecosystem, and the evasion used by scammers. Furthermore, they reported the scammer social engineering strategies by interacting with 60 scammers. Their study estimated the call center size with 11 operators, an average call duration of 17 minutes, and a service price of $291 US dollars.

The TSS study was continued and improved by Srinivasan *et al.* [134]. The authors have

[12]https://www.ftc.gov/news-events/press-releases/2017/05/ftc-federal-state-international-partners-announce-major-crackdown

followed a data-driven approach to collect, detect, and analyze TSS. They have formulated tech support queries and used it to search daily for TSS pages using search engines for approximately eight months. Their crawling has collected over 9k TSS domains from organic search results and sponsored ads. Their analysis yielded that scammers use both aggressive and passive websites. Aggressive websites are supported with a combination of continuous pop-up messages/dialogue and audio messages describing the problem to convey a false sense of urgency. On the other hand, passive websites use official brand-based images, certifications, and simple textual content to look like genuine tech support representatives, making it more challenging for detection services.

Furthermore, a joined work between Microsoft Research and the Microsoft Digital Crimes Unit to detect TSS was conducted inn [95]. The authors have used a web scraper powered by Cloud AI services to crawl and capture TSS pages snapshots continuously. The crawler collects around 100k new images potentially having TSS related text from around 150k suspicious URLs. They then use Cloud AI services to verify the likely existence of a scam and extract phone numbers and signatures from the captured websites snapshots. Furthermore, they model the connection between the scam domains, URLs, phone numbers, and visual signatures of scam webpages in a graph structure with a friendly user interface, and expose it for forensic analysis. This, in turn, helps law enforcement partners and investigators track the evolution of scam operations.

Finally, Rauti and Leppánen [125] presented a qualitative analysis of the main characteristics of TSS. The authors have engaged with ten scammers pretending to have Windows-related issues and looking for online technical support. Their analysis showed that TSS has four phases. The first phase is attracting the victims through convincing support websites and phone calls or live chat, and try to acquire remote access to his/her device. The second phase is misusing the system tools, such as Event Viewer and netstat -ano command, and claims that the listed IP addresses are connections associated with viruses to show that something is wrong with the victim's system. In the third phase, the scammers pretend to clean up the system and optimize it through running legitimate software such as CCleaner and Malwarebytes, which is the best time to install malware and dubious software into the victim machine. Hoping that their work is convincing, the scammers ask for the payment in the last phase.

2.5 Other Scams

In this section, we are briefly discussing other types of scam, which are telephone scam and romance scam. In the telephone scam, the attacker aims to illegally acquire money

from the victims and tricking him/her into doing harmful actions for the scammer's benefit [151]. Recently, the telephone has become an attractive medium for scam operations, especially with automation, high reachability, and low economic cost [151]. According to Federal Trade Commission (FTC) reports for 2018, the telephone scam has been growing significantly, with more 69% of the unwanted call complaints received by the national Do-Not-Call Registry were conducted through phones [151]. The main threat of the telephone scam is its potential to be convincing by following social engineering techniques, such as impersonating a familiar contact ID, a company representative ID, or Internal Revenue Service (IRS) [24, 106, 151].

In [151], Tu *et al.* presented a systematic study to analyze the reasons behind the telephone scam effectiveness and how to defend against them. They have conducted ten ethical telephone phishing scams on 3k university participants without prior awareness and reported that impersonating an internal entity and spoofing Caller ID had the most significant effect on the attack success. To prevent such attacks, the authors recommend increasing the employees' awareness against telephone scams and using caller ID authentication systems to reduce the risk of caller ID impersonation.

In another study, Mubarak *et al.* [106] presented scenarios used by Malaysian scammers to carry out their scams, such as randomly calling phone numbers to try their luck in getting a victim and trying to putting the victims into a complicated situation to scare them and play with their emotions. Furthermore, the authors suggested several prevention mechanisms to protect the users from telephone scams, such as the cooperation of service providers to block the scam numbers, increase the users' awareness against telephone scams, and the use of trusted third-party applications with build-in blacklist numbers.

Another scam type is the "romance scam", which can cause considerable emotional damage in addition to financial losses. In this case, a false relationship is initiated by the scammer using chat services, social media, and dating sites. The victim is then asked to provide some financial support to the scammer. This scam and its serious emotional consequences has been studied in [31, 32, 91, 162–164].

Finally, the recent COVID-19 pandemic has created new travel bans, stay-at-home orders, and lockdowns restrictions. This, in turn, has increased the reliance on online communications and technologies. Cybercriminals leveraged the recent new measures and exploited the pandemic to facilitate a broad range of Cybercrimes such as obtaining illicit financial gains and disturbing services [63]. For example, attackers have carried ransomware attacks using a Mobile application called CovidLock. The attackers advertised that the application could monitor COVID-19 cases and provide visuals of statistics and heat maps. However, the application asks for permission to lock the user's pictures, videos, contacts,

and social media access during the installation. The attackers have threatened the victims to erase the data or publish it online if a bitcoin ransom is not paid [161].

2.6 Malware Detection in Android Mobile Applications

Finally, many researchers have studied malware detection in mobile applications. The majority of these researches focused on extracting a set of features from the android application packages (APKs) to be used in a classifier. An example of the extracted features are the application permissions, the permissions used within the app, and the API calls in the application code. Drebin [12] uses SVC classifier to distinguish between benign and malicious APKs based on a set of features extracted from AndroidManifest.xml and disassembled code. In [73] Idrees and Rajarajan proposed a detection method that utilizes classes.dex and manifest files to extract a feature set from the permissions and API calls. A similar approach was followed by Yang and Wen in [102]. Other researchers focused entirely on APK permissions or code behaviour to decide if the application is harmless or not. The reader can refer to [14, 77, 144, 173] for more information.

2.7 Discussion

In this section, we leverage our analysis and understanding of GHS, BGS, and other different fraud activities by grouping the fraud activities based on some common features. In our work, we use four different features to create the groups; a) the communication medium used to reach the victim, b) the technique used to approach the victim and the promised gain for the victim, c) the attack effects on the victims, c) and the possible mitigation to reduce the number of victims. Table 2.6 shows our analysis based on these four features.

2.7.1 Communication Medium Used to Reach the Victims

In this section, we group the fraud activities based on the communication medium the attackers can use to reach out to the victims. Overall, we have derived three groups:

- **Mobile**: Attackers use the telephone to communicate with victims. For example, the attackers can impersonate an IRS officer and claim that the victim did not pay his/her total tax amount. The attackers then ask the victim to transfer the due amount to avoid legal circumstances.

Fraud/Scam	Initiation	Effect on victim	Detection/mitigation	Communication channel
survey	Paid task Unexpected money or awards	Financial loss Sensitive information loss Malicious software	Web scrapping (distinguishing features) Increasing awareness	Web
TSS	Fake service	Financial loss Malicious software	Web scrapping (distinguishing features) Increasing awareness	Web Mobile
Telephone scams such as premium call lines	Impersonating Deceiving Threats/Extortion	Doing harmful actions Financial loss	Increasing awareness	Mobile
Romance	Deceiving Fake relation	Emotional consequences Financial loss	Web scrapping (distinguishing features) Increasing awareness	Web Mobile Physical/In-person
Health care	Fake service Impersonating Deceiving	Sensitive information loss Money loss	Web scrapping (distinguishing features) Increasing awareness	Mobile Web Physical/In-person
HYIP	Investment Fast gain	Financial loss	Web scrapping (distinguishing features)	Web Physical/In-person
P&D	Investment Fast gain	Financial loss	Market movement (price, volume, etc.)	Web Physical/In-person
DDoS	No promise to victims (unwillingly)	Service disruption Financial loss	Traffic analysis	Web
Ransom	Fake service Deceiving Threats/Extortion	Financial loss Service disruption	System resources (distinguishing features)	Web Mobile Physical/In-person
ML	Investment	Financial loss	Market movement Increasing awareness	Web Physical/In-person
Cryptojacking	No promise to victims (unwillingly)	Service disruption	Web scrapping (distinguishing features) System resources (distinguishing features)	Web
IRS	Impersonating Deceiving Threats/Extortion	Financial loss	Increasing awareness	Web Mobile
Subscription	Fake service Deceiving	Financial loss	Web scrapping (distinguishing features) Increasing awareness	Web Mobile
Nigerian scam	Paid task Unexpected awards	Financial loss	Web scrapping (distinguishing features) Increasing awareness	Web Mobile Mail
Lottery/Prize	Unexpected awards	Financial loss	Web scrapping (distinguishing features) Increasing awareness	Web Mobile Physical/In-person
Refund	No promise to victims (unwillingly	Financial loss	Increasing awareness	Web Physical/In-person
Forgery	Fake service	Financial loss	Increasing awareness	Web Physical/In-person
GHS	Fake service Unexpected awards	Financial loss Malicious software Sensitive information loss	Web scrapping (distinguishing features) Increasing awareness	Web
BGS	Investment Fast gain	Financial loss Malicious software	Web scrapping (distinguishing features) Increasing awareness	Web

Table 2.6: Fraud activities categorization

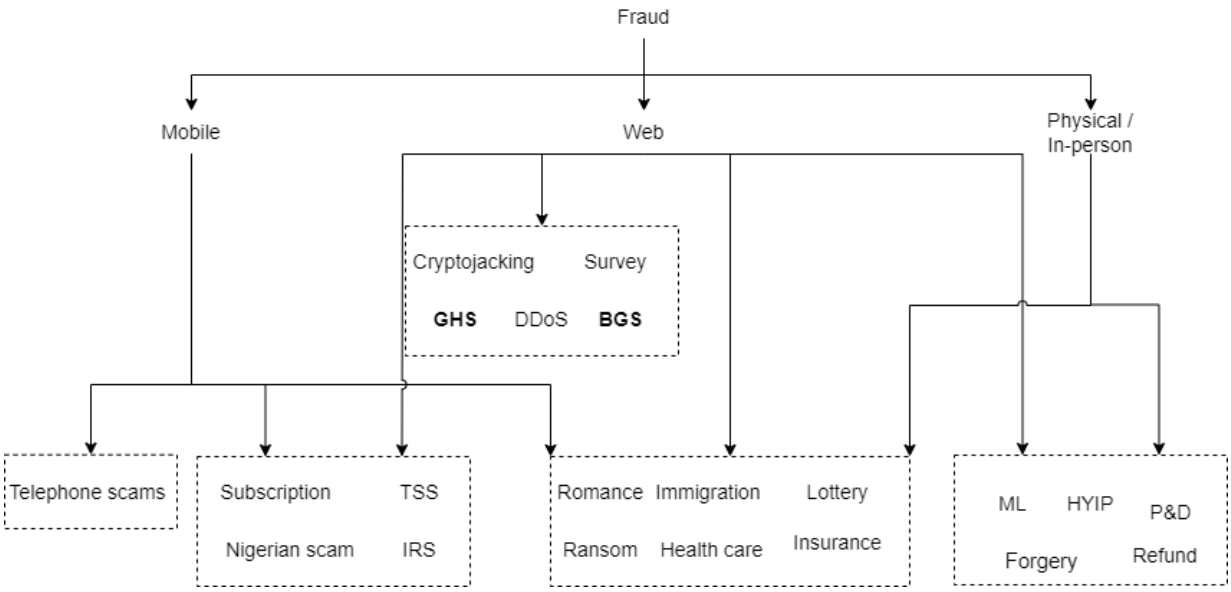

Figure 2.4: Categorization based on the communication medium used to reach the victim.

- **Web**: Attackers communicate with victims through emails or web pages. For example, in the GHS, the attackers create a website and advertise their ability to hack the victim's favorite game server and provide them with free, unlimited resources.
- **Physical/In-person**: Attackers approach victims In-person. For example, The attacker can use an old or stolen receipt to return an item that s/he picked up from the store shelf

Figure 2.4 shows a mapping between the different crimes and their related groups. As can be seen, many scam activities such as romance and lottery scams can be carried out through several different communication mediums. In the case of BGS and GHS, they are carried mainly through the web, which makes it easier to detect and mitigate them by proactively monitoring the web.

2.7.2 Techniques Used to Approach the Victims

We also group the fraud activities based on how the attackers approach the victims and the expected gain the victims will supposedly receive. Overall, we have derived four groups:

- **Impersonating, deceiving, and extortion**: In this group, attackers threaten and intimidate their victim to carry out the attack. For example, in ransomware attacks, the attackers threaten the victims to delete their data or publish it online if they do not pay the ransom.

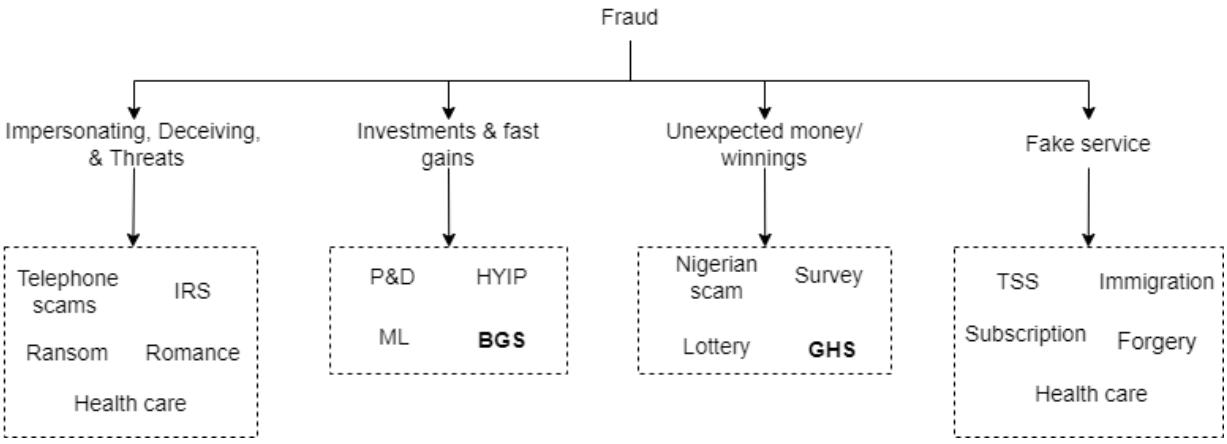

Figure 2.5: Categorization based on the attacker approaching technique.

- **Investments and fast gains**: In this group, attackers approach the victims through investment activities and the promise of fast gain. For example, in BGS, attackers promise the victims a large sum of money using a small amount they pay.
- **Unexpected money/winnings**: In this group, attackers approach the victims by promising a large sum of money or a prize in exchange for a service. The effect of the crimes in this group varies and ranges from wasting the victim's time to causing financial loss. For example, In GHS, scammers promise to provide victims with free, unlimited "resources" or other advantages for their favorite game in exchange for completing one or more tasks.
- **Fake service**: In this group, attackers convince their victims that they run a legitimate business and offer them their services. Similar to the previous group crimes, the final effect varies from wasting the victim's time to causing financial loss. For example, in TSS, the attackers convince their victims that their machines are infected with malware and offer a fake technical support service.

We present the mapping between the fraud activities and their related groups in Figure 2.5.

2.7.3 Effects on the victims

Here, we group the fraud activities based on how they affect the victims into five groups:

- **Financial loss**: In this group, victims will lose some money. For example, in BGS, victims transfer mining fees to the attackers without receiving any money back.

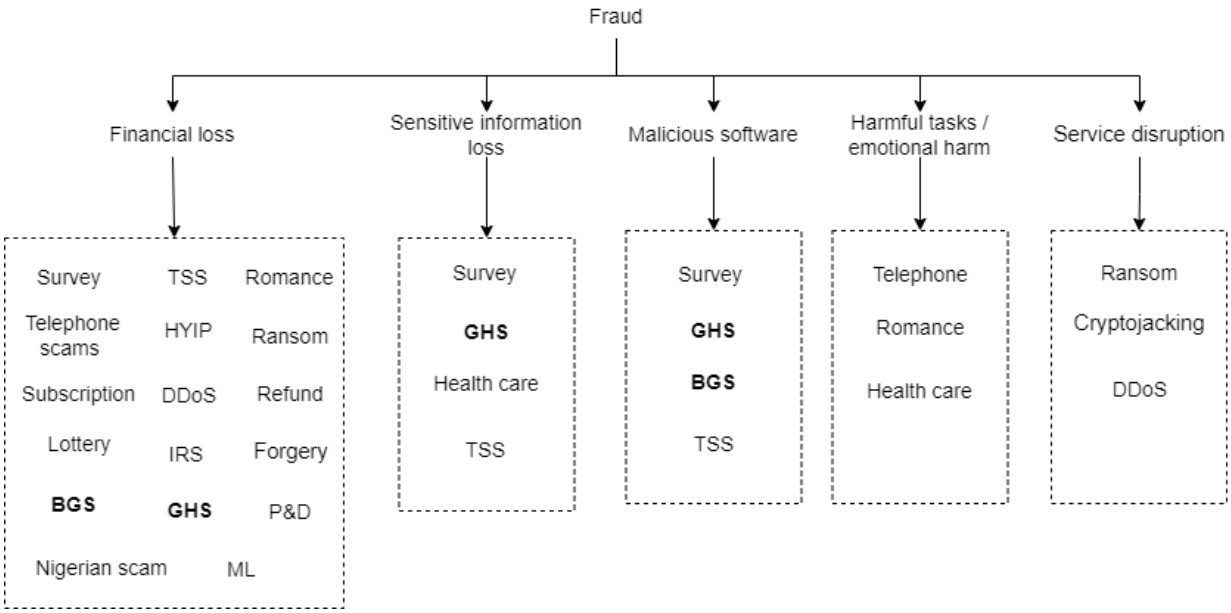

Figure 2.6: Categorization based on the effects on the victim.

- **Sensitive information loss**: In this group, attackers will steal personally identifiable information, which can be used in identity theft attacks.
- **Malicious software**: In this group, attackers ask the victims to install suspicious executable files on their machines. Attackers can then lock the victims' machines and ask for a ransom.
- **Harmful tasks or emotional harm**: In this group, attackers will lead the victims to complete harmful tasks or harm them emotionally, such as in the case of romance scams.
- Attackers can also cause a **service disruption** which may lead to financial loss or other consequences.

We map the fraud activities to their related groups in Figure 2.6. As can be seen, some fraud activities such as GHS and BGS can cause harm to the victims in different ways.

2.7.4 Mitigation Methods

Finally, we group the activities based on how we can mitigate them and reduce the total number of victims. Overall, we have derived three groups:

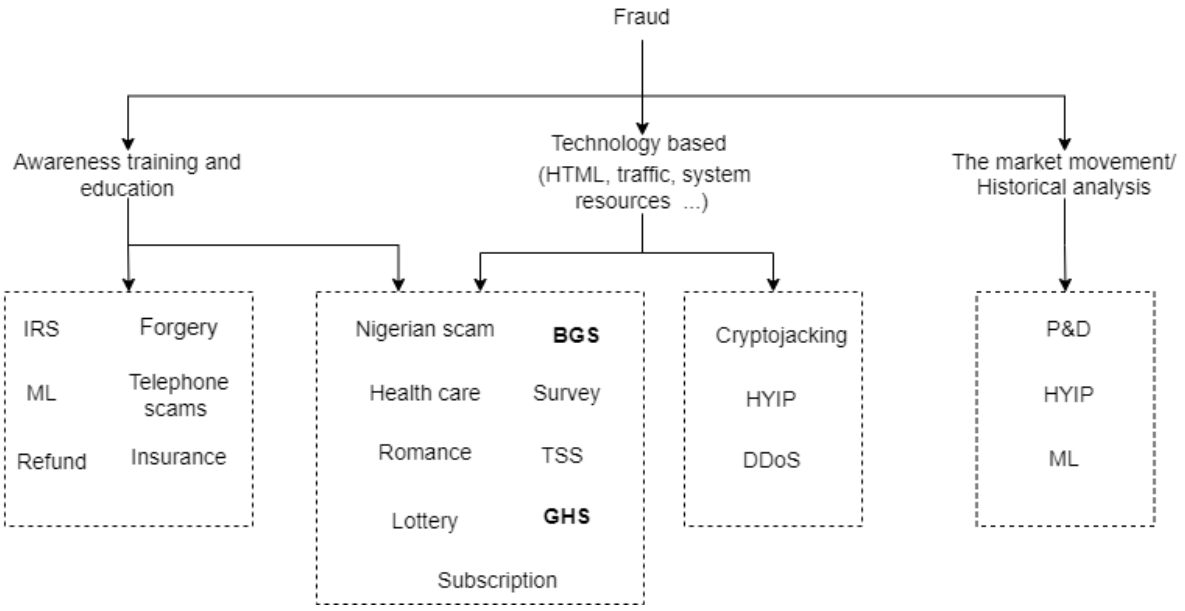

Figure 2.7: Categorization based on the detection/prevention methods.

- In the first group, we can utilize the **market movement and the historical data** to extract distinguishing features to detect the scam and provide an early warning for possible victims. For example, to detect P&D we can use features like the market capitalization, the price, and the volume to detect the attack.
- In the second group, we can utilize **technology related data** such as the HTML content, the web traffic, and system resources to detect the scam and provide an early warning for possible victims. For example, we have used the text in the web pages to extract distinguishing features to detect GHS and BGS scams.
- In the third group, we believe that **increasing the people's awareness and educating them** about these scams is the best mitigation.

We map the fraud activities to their related groups in Figure 2.7. In many cases, fraud activities can be mitigated using different methods. For example, we can reduce the number of GHS and BGS victims by implementing a model to detect the scam and increase awareness about it.

2.8 Conclusion

In this chapter, we overview different types of scam attacks, such as the TSS, survey scam, and cyberattacks related to cryptocurrencies, such as HYIP and P&D. We also discuss

what the researchers have proposed to detect and analyze these attacks. Many of the researchers studying scam detection and analysis do not disclose their datasets. This, in turn, makes it harder for others to utilize it in developing more powerful defensive mechanisms against these attacks or even to be able to reproduce their results. HYIP detection methods have the issue that they cannot adequately provide early detection and prevention. The researchers depend on extracting features from the transaction history of Bitcoin addresses found in previously reported scam campaigns. The work conducted in these studies was directed to solving a single scam (the studied scam), and none have proposed a generic system that can be used to detect other types of scams.

Our research goal is to fill these gaps. We compile and publish the first GHS and BGS scam datasets with a large GHS dataset, including around 65,000 pages mapped into more than 5,000 domains. Our BGS dataset contains more than 1,200 domains and more than 9,000 cryptocurrency addresses. In addition to scam URLs, we also provide DOMs of scam pages, allowing others to compare our methods to their approaches or use them to create more powerful detection mechanisms. In addition to our accessible dataset, we propose a generic detection and analysis system that can be applied to scam attacks with a web presence. Our system has proved its efficiency in detecting both the GHS and BGS attacks. In BGS analysis, our system was able to detect more than 70% of the live Bitcoin scam addresses before they received any funds. Finally, we propose a semi-automatic approach to prepare a training dataset for GHS and BGS using a few scam samples and minimal human interaction. We will discuss the GHS, the BGS, and the training dataset creation in more detail in the rest of this thesis.

Finally, in the light of common features shared between the different scam activities, we create a scam taxonomy and situate GHS and BGS with other types of scams. Overall, we have created four different categorizations based on the communication medium, the approaching techniques, the effect on victims, and the mitigation methods. Our analysis showed that attackers carry their attacks using the web, telephony, or in person. They approach the victims in different ways, such as deceiving the victim by impersonating another person or providing a fake service, or by incentives such as promising a good investment with a fast gain or unexpected prizes. The final effect on the victims varies and include, but are not limited to, financial loss, personally identifiable information loss, emotional harm, and service disruption. We can mitigate and reduce the effect of these attacks in two different ways; monitoring the communication channel for those attacks that are carried through the web or mobile and increasing awareness for those carried out in person. Finally, we believe that educating the public and increasing awareness about the different scam activities will reduce the overall effectiveness of the attacks.

In this thesis, our motivation was to provide insights into GHS and BGS scams that

target everyday web users. Although both scams receive high traffic and many victims fall for them, they still fly under the radar, and the current public blacklists do not report them adequately. Our analysis shows that BGS and GHS are simple scams where the scammers promise the victims different incentives, and trick them into losing their money or sensitive information. We also show that we can provide early detection of the two scams by monitoring the web.

Chapter 3

The "Game Hack" Scam

3.1 Introduction

In this chapter, we perform the first study of the game hack scam (GHS). We have developed a model for generating thousands of GHS-related search queries. We have used these queries daily on popular search engines for thirteen months. We look for GHS instances in the pages directly returned by the search engines, and we crawl the other pages down one level to find additional GHS instances. By this method, we have discovered thousands of domains associated with GHS. Moreover, we have collected 59 executable files that were reported as harmful by virus total [116], and many of them were reported by locally installed anti-virus scanners as well. Finally, we have collected more than 400 modified Android games APKs and compared them to their respective APKs from google play.

Our main contributions on studying the GHS are the following:

- We designed a search-engine-based system to study and provide insight into GHS that targets game players.
- We uncovered more than 5,900 GHS-related second-level domains, and 375 offers second-level domains.
- We show that the attackers routinely target a vast array of games.
- We show that the existing public blacklists (PBLs) are ineffective against this type of scam.
- By analyzing the GHS URLs that are shortened by Bitly, we estimate that these attacks have been clicked at least 150 million times in the period from mid-2014 to mid-2019.

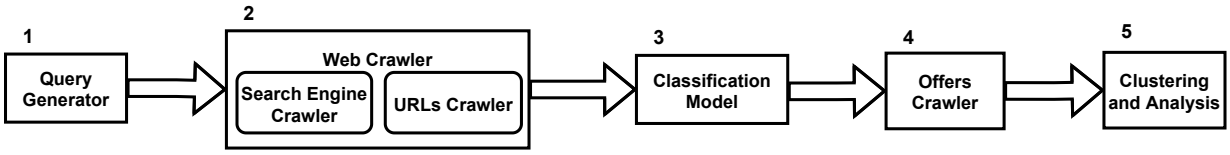

Figure 3.1: Games scam detection and analysis model

All the data used in this chapter is available at `http://bit.ly/GHSJWE`.

The remainder of this chapter is structured as follows. After this introduction, in Section 3.2, we introduce our methodology to detect the GHS. In Section 3.3, we report some basic numbers obtained during our crawling period. In Section 3.4, we carry out various analyses and discuss the results. In Section 3.5, we present a click-through analysis based on scam links shortened using the Bitly service. In Section 3.7 we discuss some of the main limitations in our model. Finally, we conclude in Section 3.8.

3.2 Methodology

We started our research by manually searching for and exploring games scam pages. This helped us get a broad understanding of the scam and provided our initial data samples, which we used to build our data-driven model to detect and track GHS. That allowed us to run automatic crawls on the web to find GHS pages which we then analyzed. After that, we employed our understanding of the scam to prepare a more extensive representative data-set to improve our model. As a result, we have created a large data set with a variety of samples, which we used to collect and identify more GHS instances. Figure 3.1 describes our complete system, which includes five modules: Search query generator, Web Crawler, Classifier Model, GHS instances triggering, and Analysis.

3.2.1 Training Dataset Creation

Preparing our GHS dataset was challenging since there was no previous large-scale analysis of the GHS or third-parties data. As a result, we manually searched for and collected GHS instances and benign pages to train an SVC classifier. We have used the classifier to filter out the URLs we have collected over the first five months of our crawling, from May to September 2018. Through that period, we have collected over 33k GHS instances that contain some noise. We then used these GHS instances as a seed to prepare a more representative GHS data set to train our final model.

Our analysis of the scam pages we identified in the first five months of crawling showed that the GHS instances are often based on similar templates that the scammers create and distribute. These templates usually have a similar DOM structure as well as similar text. The scammers only need to change the game name and the name of the in-game resources to create a new scam instance -An example of such templates is provided in Section 3.4.2-. We used this finding to filter out some of the noise in our initial data set. We have used the clustering method proposed by Cui *et al.* [47] to cluster together pages with similar DOM structures. We then manually inspected two to three pages selected randomly from each cluster to flag the clusters as true or false positive. This allowed us to identify and remove hundreds of pages wrongly classified as GHS instances. We then selected a GHS instance that we manually verified from each TP cluster. This process resulted in a dataset of 835 GHS instances, which we used as our scam training dataset[1].

We applied the same approach to a set of 8k pages classified as clean to prepare a benign training data set. We have picked 1 page from the generated clean clusters accumulating to 1,079 clean pages for the training dataset.

3.2.2 Search Query Generator

Finding good search queries that have a high likelihood of leading to scam pages is an important task. Kharraz *et al.* [88] used Google Trends service to generate such queries. Srinivasan *et al.* [134] used the context-specific corpus. In our work, we have used a combination of these two techniques. We also enhanced our queries set using the top popular games names to cover as many GHS instances as possible:

- **Utilizing the context-specific corpus:** We have used our initial corpus of pages leading to GHS instances as well as the GHS instance pages themselves to generate search queries using the Markov assumption [81] to approximate n-gram probabilities. We extracted the bag of words from our corpus. We found 1,964 words that have a frequency greater than ten. We manually selected 39 words based on their direct connection to GHS and added the stop words "without" and "no". We then generated our n-grams for $n = 3$ to 7[2]. That gave us 795 n-grams, and we manually selected 410 search queries from them. Table 3.1 shows some examples of the generated n-grams, and the full details are available at `http://bit.ly/GHSJWE`.

[1]The scam instances that belong to the same cluster hosts highly similar contents with slight modification, such as the game name and the in-game resources name.

[2]Our experiments showed that 8-grams and up did not improve our results.

n	#n-grams	Example English Phrase
3	233	generate unlimited coin
4	109	hack tool no survey
5	44	hack cheats unlimited free coin
6	18	hack cheats unlimited coins ios android
7	6	hack cheats unlimited free coins diamonds generator

Table 3.1: Summary and Examples of Generated n-grams Related to GS.

- **Popular games:** we then created a list of 966 game names by extracting Facebook, Google, and iTunes top games, and we combined each of these game names with 9 of our n-grams, thus getting 8,694 new queries, for a total of 9,104 queries.
- **Google trends service:** Google trend reflects the popularity of search queries as it is used by normal web users. The researchers can crawl Google trend API[3] to generate more queries related to the scam in the study. For example, Kharraz *et al.* [88] used it to generate a list of the 10,000 search queries related to survey scam. In our work, we have crawled Google trend API twice to generate the search queries. In the first crawling, we used the 9,104 queries generated above as search terms. After manually filtering the non-GHS related queries, we acquired 972 new GHS related queries. We used the 972 newly generated queries as a seed in the second crawling and generated 872 more queries.

Our final query list contains 10,708 search queries.

3.2.3 Web Crawler

The primary purpose of this module is to browse the web and search and identify scam instances. Here, we first use the previously identified search queries as a seed to search daily for scam pages using search engines such as *Google.com*, *Bing.com*, and *search.yahoo.com*. For each query, we only consider the first and second pages (that is, 20 search results) returned by each engine. Afterward, our crawler automatically visits the found URLs and collects their related data, including URL redirections, HTML content, a screenshot of the landing page, and resources (scripts, CSS files, etc.).

We build our crawler based on ChromeDriver[4] and Python Selenium[5]. Using Python

[3]https://trends.google.com/trends/?geo=US

[4]`http://chromedriver.chromium.org/`

[5]`https://selenium-python.readthedocs.io`

beautifulsoup[6] and the CSS selectors, the URLs resulted from the search can be extracted and crawled. For the crawling process, we propose a lightweight scripted headless browser built using python by integrating Selenium, ChromeDriver, and BeautifulSoup.

3.2.4 Classifier

In our crawling process, the majority of the URLs we collect are either benign pages having nothing to do with GHS, or benign pages with links to GHS instances. To filter out GHS instances, we have developed a two-step classification model. In the first step, we use features that we extracted from the GHS instances DOM. These features are used to filter out the crawled pages as GHS instances, benign pages, or unidentified. In the second step, we feed the unidentified pages to a text-based SVC classifier. These pages are then flagged GHS instances or benign. Our classification model achieved very high accuracy, with True Positive Rate (TPR) above 99% and False Positive Rate (FPR) lower than 0.2%.

In our analysis, True Positive (TP) refers to the number of scam pages classified as scam, True Negative (TN) refers to the number of benign pages classified as benign, False Positive (FP) refers to the number of benign pages wrongly classified as scam. Finally, False Negative (FN) refers to the number of scam instances wrongly classified as benign. From these basic measures, the F1 score is derived as follows:

$$F1 = 2 * (Precision * Recall)/(Precision + Recall) \tag{3.1}$$

Where,

$$Precision = TP/(TP + FP) \tag{3.2}$$

$$Recall = TP/(TP + FN) \tag{3.3}$$

The higher F1, the better.

Text Classifier

To evaluate our classifiers, we used 10-fold cross-validation on the labeled dataset we prepared in Section 3.2.1. We ran our experiments using five different classifiers: Linear support Vector Classifier (SVC), Naive Bayes(NB), k neighbors (KN), Random Forest (RF), and Multi-layer Perceptron (MLP) classifiers. SVC constructs a hyperplane that can divide the data into two categories. NB is a probabilistic-based classifier that applies the Bayes theorem with the "naive" assumption of conditional independence between predictors. KN classifier predicts the target based on the similarity with known targets using

[6] `https://pypi.org/project/beautifulsoup4/`

distance measurement such as the Euclidean distance. RF is an ensemble-based classifier that constructs multiple simple decision trees during the training process and uses a voting scheme between the decision trees to predicts the target. Lastly, MLP is used in supervised learning problems and uses different layers and trains on a set of input-output pairs and learns to model the correlation between the inputs and outputs. All our machine learning models are implemented by scikit-learn python library [120].

Our research aims to explore whether it is possible to identify spam from legitimate ones in an automatic way effectively. We started by using the default parameters[7] and archived good results. Of course, some tuning may be able to improve the performance further. However, Amancio *et al.* [9] have shown that in many cases, the default parameters yield good performance near-optimal performance. The model tuning is a future work we can work on.

- Linear SVC: penalty = l2, loss = squared_hinge,dual = True, tol = 0.0001, C = 1.0, multi_class = ovr, fit_intercept = True, intercept_scaling = 1, class_weight = None, verbose = 0, random_state = None, max_iter = 1000.
- NB: alpha = 1.0, fit_prior = True, class_prior = None.
- KN: n_neighbors = 5, weights = uniform, algorithm = auto, leaf_size = 30, p = 2, metric = minkowski, metric_params = None, n_jobs = None.
- RF: n_estimators = 100, *, criterion = gini, max_depth = None, min _samples_split = 2, min_samples_leaf = 1, min_weight_fraction_leaf = 0.0, max_features = auto, max_leaf_nodes = None, min_impurity_decrease = 0.0, min_impurity_split = None, bootstrap = True, oob_score = False, n_jobs = None, random_state = None, verbose = 0, warm_start = False, class_weight = None, ccp_alpha = 0.0, max_samples = None.
- MLP: hidden_layer_sizes = 100, activation = relu, solver = adam, alpha = 0.0001, batch_size = auto, learning_rate = constant, learning_rate_init = 0.001, power_t = 0.5, max_iter = 200, shuffle = True, random_state = None, tol = 0.0001, verbose = False, warm_start = False, momentum = 0.9, nesterovs_momentum = True, early_stopping = False, validation_fraction = 0.1, beta_1 = 0.9, beta_2 = 0.999, epsilon = 1e-08, n_iter_no_change = 10, max_fun = 15000

Our experiments are performed on a Windows-based system with an Intel(R) Core(TM) i7-7700HQ CPU at 2.80GHz and 16GB RAM.

[7]We have used the same classifiers and the same parameters in our analysis in Chapters 4 and 5

Classifier	Page type	Classified clean	Classified GHS	F1
SVC	clean	1,075	4	99.57
	GHS	3	832	
NB	clean	1,060	19	98.26
	GHS	10	825	
KN	clean	1,017	62	95.26
	GHS	19	816	
RF	clean	1,068	11	97.94
	GHS	23	812	
MLP	clean	1,072	7	99.21
	GHS	6	829	

Table 3.2: Results of a 10-Fold cross-validation on the five classifiers.

We used these five classifiers to classify the crawled pages based on the text as seen by the end-user. More precisely, we have used the term frequency-inverse document frequency (TF-IDF) of the words displayed to the users to extract the training features. The TF-IDF is used to scale down the impact of less informative tokens that occur very frequently in our dataset. Table 3.2 present the results obtained with the five classifiers. As we can see, the SVC text classifier achieved the highest results with 99.57 F1 score, followed by MLP with 99.21 F1 score. The other classifiers also performed fairly well with Kneighbors having the lowest F1 score value equals to 95.26. Based on these results, we used the SVC classifier throughout our experiments in this chapter. We have manually inspected the wrongly classified pages and found that most of these pages do not have enough text to extract the features from.

Filters

A further improvement on the classifier can be obtained by directly flagging the pages that are easily recognized as GSH or as clean and only using the classifier on the other pages. There are two benefits to this additional step: first, it is much faster, and second, we can reduce both FPR and FNR.

Through manual inspection of more than 100 randomly selected GHS instances from our training set, we have identified two distinguishing features:

- "Content Locker": Many of the GHS instances contain the template provider identifier or English terms related to generators. The presence of such terms can be a good

Classifier	#FP	#FP detected by filter	#FN	#FN detected by filter
SVC	4	2	3	1
NB	19	6	10	5
Kneighbors	62	9	19	8
RF	11	3	23	7
MLP	7	3	6	2

Table 3.3: The effect of applying the filters on the training dataset.

indicator of GHS instances. To employ "Content Locker" as a feature, we search for the presence of the template provider identifier or the generator terms in the targeted page text. We report the value of this feature as a boolean value where true means that "Content Locker" exists.

- "Hack button": GHS instances usually contain a button meant initiate the fake hacking process, usually alongside a text such as "generate" or "detect device". For this feature, we simply count the number of tags related to buttons. We include the tags <button>, the tag <input>when the type is "button", and any other tag with "class" or "id" related to buttons.

These features are used to classify the crawled pages as either GHS instances, benign pages, or unidentified. By setting the values of "content locker" to true and "hack button" threshold to two, we were able to filter 415 (49.7%) of the GHS instances without introducing any false positive. We did find the threshold for "hack button" by trial and error. Additionally, negating the values of "content locker" and "hack button" threshold filtered out 10% of the clean pages without introducing any false negatives. Using these filters reduces the detection execution time: the filter feature's extraction requires 67 microseconds on average, while, for example, the SVC classifier requires 25,793 microseconds on average.

Table 3.3 presents the result of applying the filters on our training dataset. As shown in the table, using the filtering step improves the performance of all the classifiers used in 3.2.4. Our filter detects many of the FN and FP pages before applying the classifier. Thus these pages will not be classified erroneously.

3.2.5 Offers Crawler

It is not always the case that the detected scam pages contain the different types of threats to which a user may get exposed. In this module, we interact with the GHS instances,

provide the necessary inputs and follow each GHS instance's instructions to reach the final stage, at which point the list of "offers" is provided. Following these offers, the victim is asked to provide personal information, subscribe to fraudulent services, or install malware. We collect for analysis the set of offers that are provided by the GHS instances we have found.

3.2.6 Clustering and Analysis

The last stage of our model is clustering and analyzing the data to provide insight into the studied scam. For this reason, we conduct four different analyses of the different pages and domains that our model identified as GHS instances. Our first analysis is done on the GHS instances themselves. We use the identifiers found in the pages to detect similarities and infer common ownership of the GHS instances. Our second analysis is done on the "offers", the final step in the scam life-cycle. We classify the different types of offers and show the convergences into a smaller set of offers. Our third analysis is done on the domain names hosting the GHS instances and the offers. We also study the effectiveness of the current PBLs against GS. Our fourth analysis targets the modified APKs and the softwares that the victims are asked to install in some scam scenarios. Finally, our analysis showed that many attackers shortened the scam URLs using the Bitly shortener service before publishing them. Bitly provides publicly available statistics for its URLs, which we used to give a unique insight into the effectiveness and the trends of the scam.

Our analyses are presented in Sections 3.4 and 3.5.

3.3 Scam Collection and Measurement

We used our university's server as well as Compute Canada dedicated servers [8] to deploy the model mentioned above to collect the possible GHS pages. The results reported in this chapter come from data collected over a year from May 2018 to May 2019. In this section, we present some basic numbers obtained directly from our crawler and classifier.

3.3.1 Classification Result

Our system identified 65,905 different GHS instances URLs, mapped onto 5,930 unique second-level domains. 3,193 (53.8%) of these domains have only 1 GHS URL. On the

[8] `https://www.computecanada.ca/research-portal/`

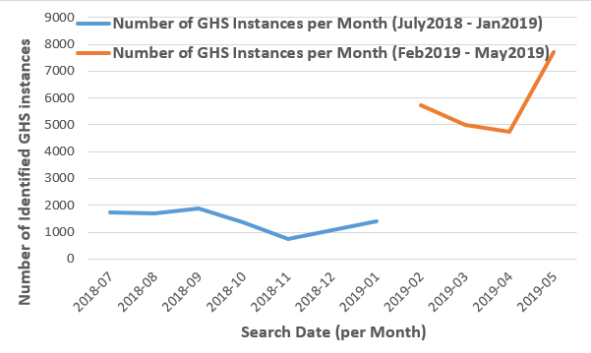

Figure 3.2: Number of GHS instances found by search engines per month

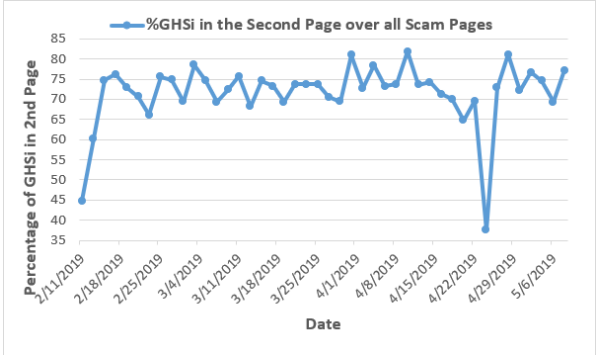

Figure 3.3: Percentage of GHS instances found in the second page over all scam pages

other hand, 739 (12.46%) of these domains host more than 9 GHS URLs. Moreover, there are many domains with a large number of GHS instances URLs. The largest three domains hosted 4,664, 2,762, and 2,439 URLs, respectively. Almost 50% of the GHS instances where identified during the filtering process. Our initial results [18] showed that none of the top 1K Alexa domains contains actual GHS instances, only links to GHS instances. Based on these findings, we only report results for the URLs hosted on domains outside Alexa top 1k.

In our analysis, we have trained our classifier on pages with English text only. Thus, we focus our research on pages with English text and ignore the other crawled pages.

Search URLs Classification

Throughout our crawling period, we have collected 775,961 different pages, 679,514 of which are in English. Our classifier identified 41,383 of these pages as GHS instances, which means our search queries yield an instance of the scam 6.09% of the time. The number of GHS instances identified per month is presented in Figure 3.2. On average, our model detected 2,009 GHS instances per month in the period from July 2018 until Jan 2019. This number has increased to 5,788 GHS instances per month in the last four months of crawling, after incorporating Google Trend services to generate new search queries and crawling the second page of each search engine. Figure 3.3 present the percentage of the GHS instances found in the second page of the search engines over the full found GHS instances. As seen in the figure, most of the GHS instances were found on the second page, this in turns explains the increase of our model performance in detecting GHS instances.

Extracted URLs Classification

We now look at the pages with links yielding to GHS instances. Overall, we have crawled 3M URLs that we extracted from the benign pages returned by the search engine. Out of these, we were able to reach and save 999,573 pages with English text. Our classifier identified 24,522 of these pages as GHS instances.

Our analysis shows that some of the domains that contain URLs yielding to GHS are blogs and domains with high traffic. This suggests that attackers target these domains to reach more victims. We found links leading to GHS instances in posts hosted in *Jeuxvideo.com*, *Groups.Google.com*, *Pinterest.com*, *change.org*, *Youtube.com*, and *npm.runkit.com*.

3.4 Analysis

In this section, we discuss the results obtained from the analysis of the data. We first shed some light over the GHS instances, their similarities, and the games they targets. We then show that the scammer relies on pre-built templates to create new attacks without any technical knowledge. We also study the offers reached when interacting with the GHS instances. After that, we look at the domain names used by servers hosting the GHS instances and the offers. We show that public blacklists are mostly ineffective against GHS. Finally, we scan the android games APKs and related executable files. We report that all of the executable files are reported to be suspicious by virus total.

3.4.1 Page Contents

Scammers use specific words in the content of a GHS page, such as the name of the targeted game and words that advertise the generator's ability to hack the game and provide the victim with in-game resources. We have used these words to extract distinguishing features and use them as a pre-classification filtering step, which reduced the classification model execution time, and increased its accuracy (see Section 3.2.4). Figure 3.4 shows the most frequent words used in the GHS pages in the form of a word cloud, where the size of each word correlates with the number of times it appears in the corpus of GHS pages.

3.4.2 GHS Analysis

In this section, we present our analysis of the GHS instances. Here we provide an insight into the relationship between the different GHS instances. We first cluster GHS instances

Figure 3.4: Word cloud based on the text contents of the gathered technical GHS pages

into groups based on unique identifiers that we have found in the GHS instances. We then look at the set of games that are targeted by related scammers.

GHS Groups:

This analysis was conducted based on the finding that many GHS pages are built using similar templates. We found at least two different online advertisement websites that either provide GHS instance templates or provide tutorials on how to copy existing templates and deploy them in the scam. An example of GHS templates is presented in Figure 3.5.

We manually inspected the DOM of several GHS pages in search of identifiers that can be used to map the scam instances to the attacker publishing them, and identified eight such identifiers. Some of these identifiers relate to analytic collections. For example, we found links to the site *histats.com* in 26,756 of our GHS instances, about a third of them and *statcounter.com* in 5,244 of the pages. It does not mean that either *histats.com* or *statcounter.com* have any part in the scam, merely that scammers tend to use these sites for their analytics. Other identifiers commonly found in the DOM of the GHS instances relate to the sites that provide the GHS templates and offers at the end of the scam. Identifiers for these two websites appear in approximately a third of our GHS instances,

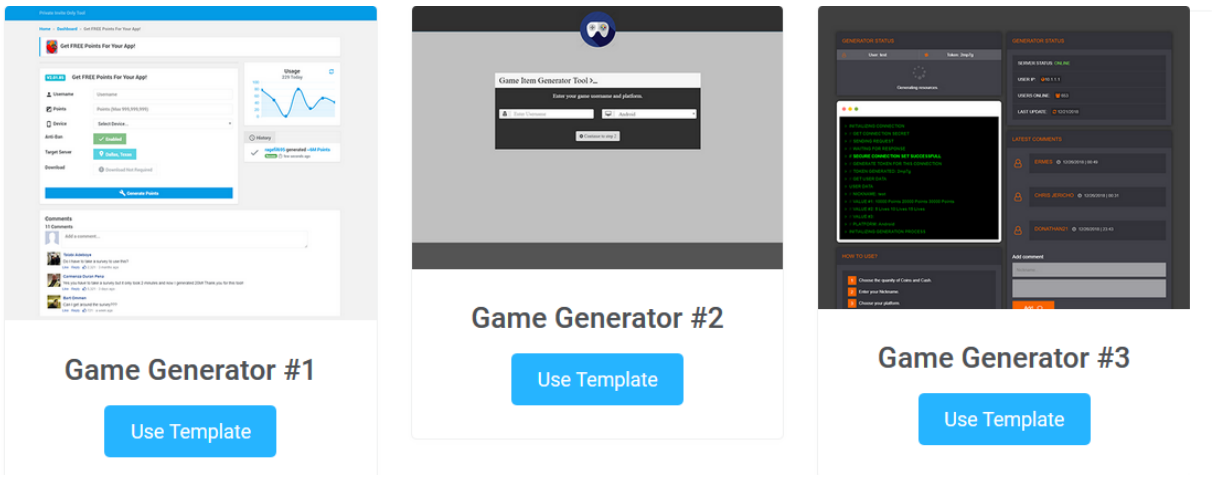

Figure 3.5: Examples of GHS templates

with 21,989 occurrences. To confirm our findings, we have created our own attack [9]. The attack can be reached at `https://dwnlds.co/3396a94`.

Some of the identifiers have unique ID values for each account, we assumed that each ID belongs to a different attacker, as suggested by our experiments. Some of the IDs appear in more than one GHS pages, which suggests that the set of pages containing the same ID belong to the same attacker. Overall, we have identified 8,450 unique ID values for the eight identifiers. 8,009 of them (94.7%) span less than five pages; thus, we excluded them to reduce the skew in our analysis. In the subsequent analysis, we used the 441 IDs that spans at least five pages. The breakdown of these IDs is shown in Figure 3.7.

Targeted Games:

Having identified clusters of attacks belonging to the same attackers, we then turn our attention to the targets of these related attacks. In particular, we wanted to understand why a given attacker would carry several attacks: was it to avoid detection, or was it to cast a wider net?

To answer this question, we looked at the actual games targeted by related GHS instances. We have extracted around 40k different game titles from our database of GHS instances. Some titles have a great number of occurrences. These are typically "generic" titles with no particular targeted game. We have identified 14 such titles. The top three are **"Generate Resources For Your Game!"** with 1,287 occurrences,**"Resource Generator"** with 590 occurrences and **"Generate Points For Your App!"** with 203 occurrences. We removed these pages since they provide no added value in this analysis. In this

[9]Of course; we did not deploy our attack, so no one was victimized by our tests

analysis, we only consider the attacker's IDs that span at least five pages in our database, since we are interested in trends among the attackers publishing several attacks.

Figure 3.6 shows our results. The x-axis represents the number of unique game title over the number of related pages, and the y-axis represents the fraction of unique attacker's IDs. We can see that around 2/3 of the attackers have at least 50% diversity in the game title they target, i.e., 50% of the games they target are unique. Moreover, 20% of the attackers target each game title only once. This clearly suggests that the attackers generate new attacks primarily to cover new games and increase the spread of their scams.

3.4.3 Offers

In this section, we provide two different analyses of the tasks that a victim needs to complete in order to obtain the claimed game resources at the "offers" stage of the scam. We first look at the spread of offers across different GHS instances. We then relate our attempts at contacting some of the "services" that are proposed by these scams.

The offers are the last stage in the GHS. Usually, they appear after the victim provides their game credentials and the fake hacking process starts. At this stage, a pop-up appears claiming that the hack was successful and the victim then invited to a "verification" step. During this verification process, some screen is shown to the user, asking to complete one or more tasks, called "offers". These offers are a dynamically loaded list of several tasks for the victim to complete.

In this research, we were able to identify and collect 375 different offers websites. Many of the identified offers are subscriptions for services advertising online libraries and video/-music streaming. All of these domains use very similar site templates and similar sign-up forms. Moreover, their second-level domain names tend to be created following similar patterns; the books sites contain "book" in the domain name and the streaming sites contain "music"/"play" in the domain name. In general, these sites claim to have a free trial period, but a valid credit card must be provided to enroll. It is very doubtful that any of these sites would provide any service at all. Other users reported their inability to get through these sites customer services[10][11]. As an example, subscription scam is the sixth-highest scam causing money loss in Canada with $2.9M in 2015[12].

[10] https://truecall-es.com/review/8889809787

[11] https://www.onlinethreatalerts.com/article/2018/6/3/beware-of-music-g8-at-musicg8-com-it-is-a-fraudulent-website/

[12] https://www.cbc.ca/news/canada/british-columbia/canada-s-top-10-scams-earned-crooks-1-2b-last-year-says-bbb-1.3471279

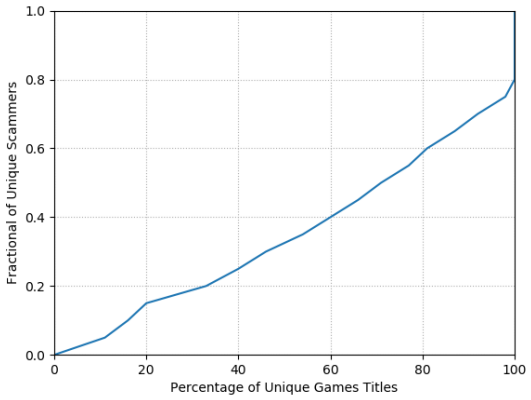

Figure 3.6: Number of games each scammer spans

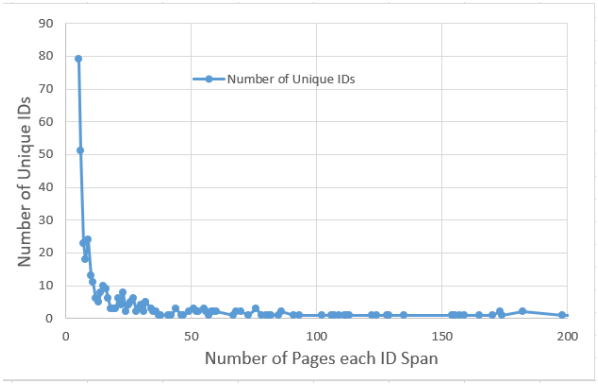

Figure 3.7: Breakdown of the number of GHS instances per ID.

Other websites ask the victim to download and install executable files. Unsurprisingly, these executable files are flagged as malware by sites such as virus total [116].

Finally, some of the offers are sites that promise free vouchers, gift cards, and free products in exchange for completing surveys. These websites are part of the survey scam which has recently been explored in prior work [43, 88]. For example, prize scams are the third-highest scam that caused money loss in Canada with $6.5M in 2015[12].

To collect the offers, we have interacted with more than 40 randomly selected GHS instances that are hosted on different domains. Of these 42 seemingly unrelated GHSis, we only got 14 different CLs, already showing convergence across attacks. This shows that many of these GHSis use the same CL, and thus will send the victim to the same offer sites. Moreover, as already noted, a third of our GHSis contain identifiers from *cpabuild.com* and/or *ogads.com*. This indicates that these GHSis use CLs provided by these two sites.

Each CL gives a choice of several offers (see e.g. Figure 3.8 and Figure 1.5 image 4). An initial crawling of the various CL links suggested that they would lead to different offers. However, we found that the CLs are actually dynamically loading the offers, and thus consecutive accesses to the same CL provides different offers to the victim. We therefore crawled continuously all of our 14 different CLs and found that overall, 115 different offers were presented across all the CLs. We saw a large overlap between the offers provided by the different CLs. Figure 3.9 provides some details. Almost 22% of the 115 offers were reached by all 14 CLs, and almost 75% of the 115 offers were reached by at least 12 of the 14 CLs. In other words, on our database, all the attacks basically lead to the same set of offers.

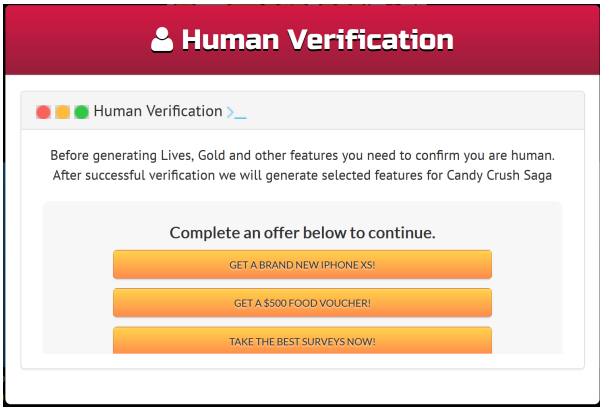

Figure 3.8: An example of the scam content locker.

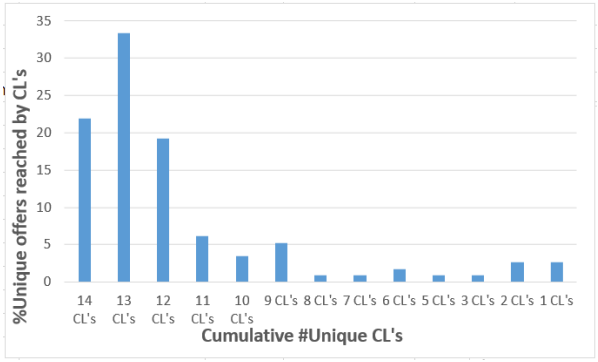

Figure 3.9: Percentage of Offers Reached per Number of CLs

Offers Reach-out

We attempted to contact ten of these offers using the email and phone listed on their websites. We targeted sites that claim to provide books and streaming subscriptions, inquiring about the content of their offerings. Nine of these sites replied back with the same auto-reply message, stating that a support ticket had been created and that a representative will follow-up as soon as possible. Only one of these sites reached out to us. Its representative simply suggested to sign-up to the service to check what the site offers. Moreover, the representative stated that we should use a credit card with at least 50$ to create a free trial account.

When we called the services and managed to get through, we ended up on automated answering systems with similar options across all sites. Most of the time, the system simply loops between options and repeats the same messages. Other users report their inability to get through at all[10]. In the rare cases in which we reached a human customer support, that person was answering for several sites and couldn't provide any useful information beside advising to register for an account. Many negative reviews can be found online for these sites, all suggesting that these sites are indeed scams.

In order to not provide financial support to the attackers, we had to stop our experiments when a valid credit card with available funds was necessary to proceed. We did not pursue our inquiries past that point.

3.4.4 Domains Analysis

In this section, we analyze the domains names of the servers hosting generators and offers. We first present the most abused TLDs in the GHS domains. We then compare these

TLD	%	Num Domains	TLD	%	Num Domains
com	39.37	2,284	org	3.86%	224
club	7.96%	462	us	3.05	177
xyz	5.84%	339	top	2.98%	173
online	4.71	273	win	2.88%	167
net	4.58	266	pro	2.64%	153

Table 3.4: Most common top-level domains (TLDs) for the final URLs of GHS instances.

domains with popular Blacklists as well as Google safe browsing [115] and virus total [116].

Most Abused TLDs:

For the generator domains, the most common TLD in our database is *.com* which appeared in 39.37% of the final-landing scam domain names. The second most common TLD is *.club* appearing in 7.96% of the domains names. *.xyz*, *.online* and *.net* each represent more than 4% of the domain names. Table 3.4 shows the details. In the case of the offer domains, we find that *.com* and *.net* are by far the most common TLDs, used in 71.2% and 17.87% of the time respectively.

Overlap with Blacklists:

We checked if the domains of the final URLs of our scam domains are flagged by some of the popular public blacklists (PBL), including malwaredomains[13], SANS[14], abuse.ch[15], Malc0de database[16], malwaredomainlist[17], and hpHosts[18]. For each domain, we check if blacklisted and if so, when it was first added to the list. Only 110 of the 5,930 domains hosting GHS instances are blacklisted by at least one PBL (1.85% of the domains).

Moreover, we have scanned our domains against Google Safe Browsing [115] and virus total [116]. We found that 336 (5.66%) of the domains are flagged by virus total, and 8 (0.13%) by Google Safe Browsing. Cumulatively, we have only 398 (6.71%) of the domains identified as a scam.

[13] `http://www.malwaredomains.com/,accessed`
[14] `https://isc.sans.edu/suspicious_domains.html`
[15] `https://abuse.ch/`
[16] `http://malc0de.com/database/`
[17] `https://www.malwaredomainlist.com/`
[18] `https://hosts-file.net/`

The PBLs fare better when it comes to the offer domains. Although 3 of the PBLs do not flag any of the offer domains, hpHosts flagged 189 (50.4%) of the offers domains. However, these domains were black-listed long after their registration date. On average, they were black-listed 918 days after the domain registration, and the earliest black-listing time was 34 days. However, we should note that we do not know when the domain started to host scams actively.

Similar to the generators domains, we scanned the offers domains using Safe Browsing [115] and virus total [116]. We found that 96 (25.6%) of the domains are flagged by virus total, and 5 (1.3%) by Google Safe Browsing. Cumulatively, we have only 233 (62.1%) of the domains identified as a scam.

These results suggest that the current PBLs are ineffective against GHS attacks, as they are against other scams such as the Technical Support Scam [134]. A system such as ours is much more effective at protecting end-users.

3.4.5 Executable Files and Modified APKs

In some cases, the generator is bypassed and the victim is directly presented with an alternate way to supposedly hack their favorite game. In this case, the attacker either provides an executable to download: either a modified version of the wanted game (an APK executable for Android), or an executable MS Windows file. These executable files are also sometimes provided as a payload by generators. We have collected 59 Windows executable files and 325 unique modified Android games APKs.

We scanned the 59 executable files using virus total [116]. Virus total scans any file or URL with over 70 antivirus scanners and URL/domain blacklisting services. All of the 59 files were flagged by at least two anti-virus scanners, and 54 (91.5%) of the files were flagged by at least 5 scanners. Moreover, many of these files were flagged by avast[19] , avg[20] , bitDefender[21] , and kaspersky[22]. Traces for Trojan, Malware, Bitcoin miner, Coin miner, Dropper, and Adware were reported.

We were able to scan 40 of the 325 APKs using the free API of virus total[23]. 26 (65%) of the APKs were flagged at least once, and 19 (47.5%) of the files were flagged by at least 5 scanners. Traces for Trojan, Coin miner, Coin hive, Bitcoin miner, Malware,

[19] `https://www.avast.com/en-ca/index`

[20] `https://www.avg.com/en-ca/homepage`

[21] `https://www.bitdefender.com/`

[22] `https://www.kaspersky.ca/`

[23] The maximum file size allowed by the free API of virus total is 32 MB, which is smaller than most of our APKs.

Adware, and Dropper were reported. Furthermore, we randomly selected 10 of the APKs for which we could find the original game on Google play. We were not able to run 3 of these APKs. Another one turned out to not be the game at all, but simply an instance of a GHS wrapped into an app. 5 of the remaining APKs seemed to be a working instance of the original game, in which the identifier used to display advertisements in the game had been modified, probably providing income to the hacker instead of to the genuine game developer. The last APK downloads and installs another APK, which is another game store.

3.5 Bitly Links Analysis

In our corpus, 2,708 of the GHS URLs were shortened using Bitly before being published. As pointed out in [96], Bitly provides a public API that can be used to collect metrics related to its URLs. In this section, we utilize this Bitly API to gain some insights about how successful GHS attacks are. We look at the lifespan of the links and at the number of clicks each link received. Then, we look at click through over time. Finally, we analyze the traffic, to find out the most common country of origin and referrer for the victims.

3.5.1 Click Through Analysis

Looking at the click-through activity seen on the GHS links, we see that 2,694 (99.48%) of the URLs received at least two clicks and 30% of the URLs receive at least 630 clicks. On average, we see an astonishing average of 2,274.68 clicks per link, accumulating a total of 6,127,995 clicks in our database of links. Our click-count analysis is presented in Figure 3.10. This shows that the scam attracts a large number of people. If we assume that in our database, the links that go through Bitly are reasonably representative of the other links, it suggests that our 65,905 URLs have generated around 150 million clicks. What is more, our method is certainly not exhaustive, and we are probably missing many GHS URLs, so the number of people clicking through the scam is perhaps even higher still.

As for the link click duration, our analysis shows that the links have a relatively long lifespan, and 40% of the links register clicks over a year or more. Moreover, around 10% of the URLs registered clicks over two years. This suggests that the links remain effective for a long time. Click-through-duration analysis is presented in Figure 3.11.

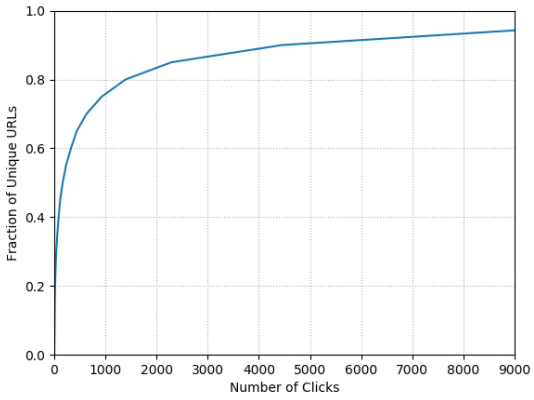

Figure 3.10: GHS click through analysis

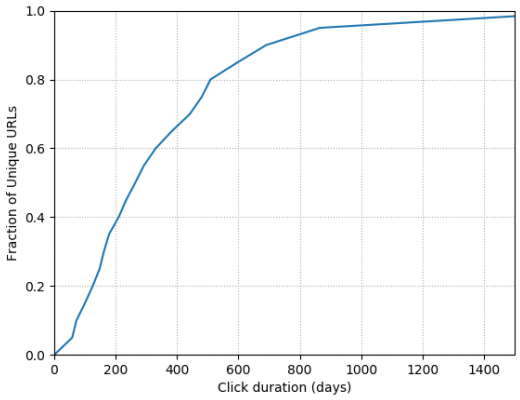

Figure 3.11: GHS click duration analysis

3.5.2 Monthly URL Clicks and Creation Analysis

In this analysis, we look at when the scam was most active. We also show that the URLs discovered in our previous analysis are still not blocked, and still have a high click rate. Perhaps the most telling metrics is shown in Figure 3.12. In this figure, we show the number of clicks received each month by our Bitly URLs since 2014. The line represents clicks count for the shortened URLs we collected as of May-2019. As shown in the figure, the number of clicks was on the rise, with a very sharp increase throughout 2018, reaching its maximum with more than 637k clicks in September-2018. We do not know the cause of this peak.

The analysis also shows that the URLs collected in the early stage of our analysis [18] stayed active for several months after their detection date, and no URL was blocked. Moreover, over the eight months following the first analysis, these URLs received around 1.35M clicks, 626.5 new clicks per URL on average. These results indicate that this scam is very active, and the number of victims is growing. Besides, this analysis suggests that no real actions are taken to stop this type of fraud. Awareness of GHS must be increased, and some suitable protection mechanisms are needed to stop it.

3.5.3 Country and HTTP Referrer Clicks

If we look at the countries from which the links have been clicked, we find a total of 245 countries, out of 254 possible country codes[24]. It shows that GHS attracts victims from nearly everywhere on earth. In terms of volume, victims in the US and India have generated

[24] `https://laendercode.net/en/2-letter-list.html`

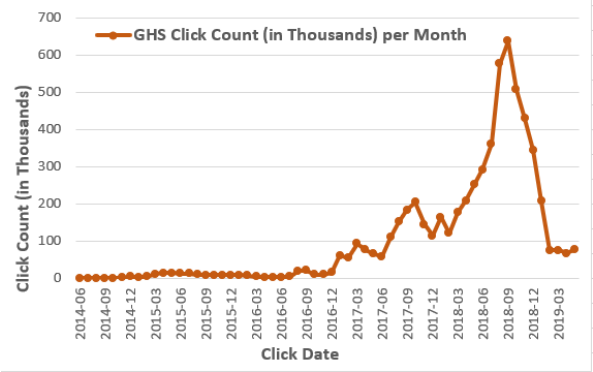

Figure 3.12: GHS clicks count per month

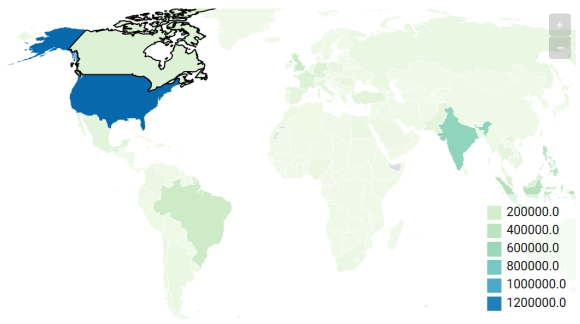

Figure 3.13: GHS clicks per country

the largest number of clicks, with 21.3% and 10.7% respectively. Figure 3.13 shows a break down of the number of clicks per country. If we consider the world population[25] to normalize the number of clicks per country per citizen; Singapore, Malaysia, and New Zealand have the highest number of clicks. To mitigate any bias that may occur from countries with low hits and a low number of citizens, we ignored any country with a population less than 1M or got less than 20,000 hits. Please refer to Table 3.5 for the top ten countries. In our future work, we will normalize our analysis using other factors, such as computer-literacy, bandwidth availability, and population's internet education/accessibility

As for the URLs referrers, we find that GHS URLs were accessed from 1,532 domains. "Direct" access is the most common source with 71.4%. Direct access includes sources like email clients, instant messages, and dedicated applications [18, 96]. Table 3.5 gives the breakdown of the top ten origin countries and referrers.

3.6 Study of two Templates Providers

In this case study, we analyze two of the largest GS campaigns we identified in Section 3.4.2. We look at *cpabuild.com* and *ogads.com* and highlight the templates and techniques they use to create and publish scam instances. We found the identifiers of these two sites in 13,393 of or GS generator pages (that's 41% of them), hosted by 1,378 different domains.

To better understand how these sites operate, we have created accounts on both systems. These two sites are (and advertise themselves as) Cost-per-Action (CPA) advertising networks. A CPA is a type of Affiliate Marketing that is used to drive traffic to a specific website [129, 133]. Both of these sites seek for other users, called "publishers", to publish their links. They provide the publisher with the URLs to publish, as well as pre-built

[25] `https://countrycode.org/`

	Top countries	Top referrals	
Rank	countries	Referrer	% Clicks
1	Singapore	direct	71.4
2	Malaysia	piktochart.com	4.6
3	New Zealand	jeuxvideo.com	3.33
4	Greece	google.com	1.48
5	United Kingdom	fliphtml5.com	1.43
6	United States	change.org	1.32
7	Canada	megatut.com	1.2
8	Australia	kabam.com	1
9	Philippines	t-adbar1.com	0.8
10	Romania	flasygames.com	0.8

Table 3.5: Referrers and countries with the highest number of clicks (countries analysis is normalized using the clicks-population ratio)

scam templates that publisher use to trick the victims. The publishers (scammers in our context) are told that they will receive a portion of the profit that the CPA site makes.

3.6.1 *CPABUILD.COM*

CPAbuild.com provides a variety of pre-built templates that can be used and customized to create generators for GS. These templates are very easy to use. An inexperienced user can easily create and publish generators for the game they want. The website also allows more experienced publishers to create and share their own templates. Figure 3.14 shows part of the GS generator creation process using pre-built templates. Many other properties can be customized. In a nutshell, a publisher only needs to fill out some provided fields, provide a game name and some pictures in order to obtain a perfectly working generator. That generator will automatically lead the victim to offers controlled by *CPAbuild.com*. This part is out of the hands of the publisher.

Once created, the generator must be hosted somewhere. If the publisher is interested, s/he can host it her/himself on one of her/his own servers, using one of her/his domain. It is, however, easier and faster to have *CPAbuild.com* to directly host the generator on one of its own servers. It provides seven different mirror websites for that purpose: `cbldc.io`, `dwnlds.co`, `swipebo.com`, `nextkon.com`, `ifreecards.com`, `cpbldi.com` and `bitsc.io`. If that option is selected, then the only task of the publisher/scammer is to publicize the generator's URL as widely as possible, in the hope that many victims will click on it and

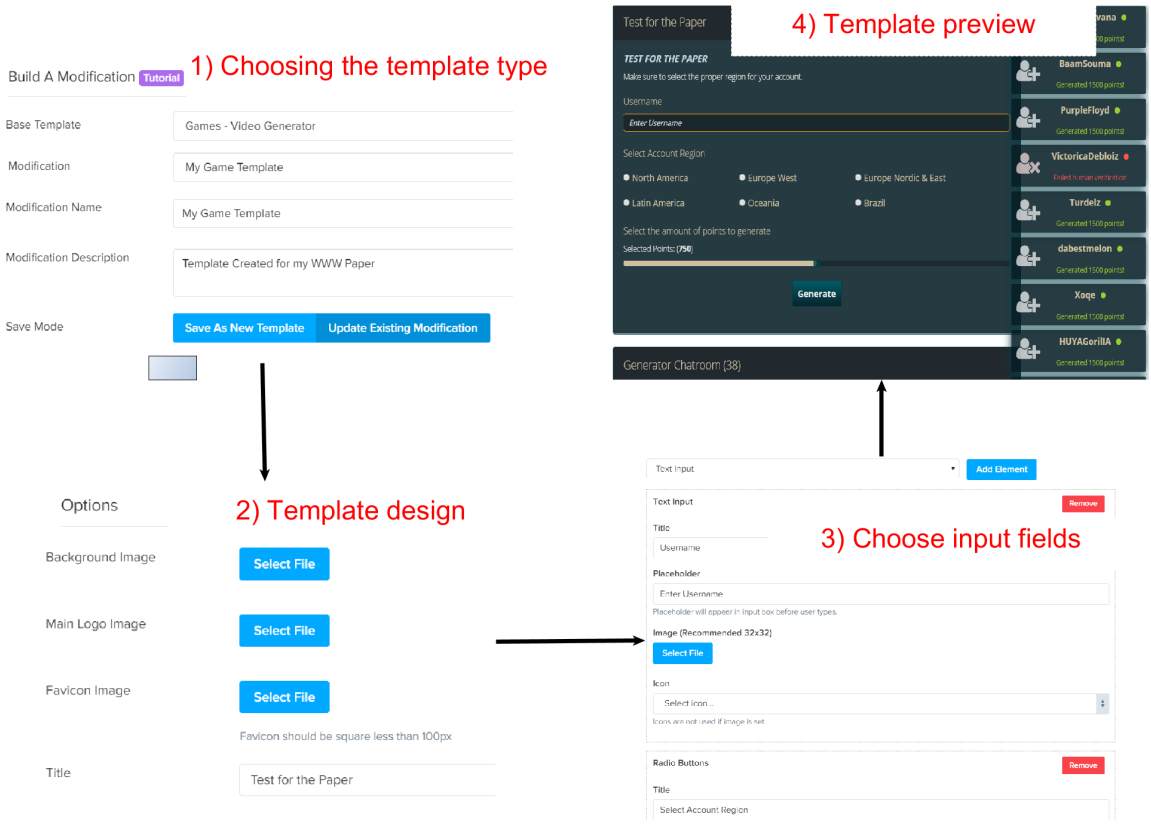

Figure 3.14: *Cpabuild.com* Template Creation Process

```
<script type="text/javascript">
    var CPABUILDSETTINGS={"it":500121,"key":"b2ebe"};
</script>
<script src="https://cpabuild.com/public/external/locker.js"></script>

<iframe src="https://cpabuild.com/public/offers/iframe.php?it=500121&key=b2ebe"
        height="200px" marginwidth="0" marginheight="0" align="middle"
        frameborder="0" width="100%" ></iframe>

http://cpabuild.com/public/locker.php?it=500121&key=b2ebe
```

Figure 3.15: *CPAbuild.com* Locker Types

follow through. If they do, *CPAbuild* claims to share some portion of the revenues with the scammer.

In addition to using templates to build generators, *CPAbuild.com* provides the option to publish the offer links directly without going through a generator. In this case, a "locker code" is provided and the publisher can simply embed that code within a controlled website. A victim accessing the website will be directly asked to complete one or more offers. Figure 3.15 shows an example.

3.6.2 *OGADS.COM*

Unlike *CPAbuild.com*, *ogads.com* does not provide pre-built templates or mirrors to host GS instances. They only provide GS content-lockers to publishers/scammers, as shown in Figure 3.16. There are however many tutorials on Youtube explaining how to copy existing templates from other scammers and simply hijack and republish them using OGADS's content-lockers instead. These tutorial are very simple and no technical background is required. These tutorials also explain how to purchase a domain, register it, configure it, and upload the GS generator to the domain. We were easily able to follow these tutorials, and within 30 minutes we created a complete functional GS instance. This generator can be seen at `https://ebadawi.github.io/candycrush/ghs/topgoodstuff.com/hack/candy-crush-saga/553834731.html`.

3.7 Limitations

One of the main limitations of our study is that we only look for GHS instances based on the ones we have already found. Thus, some of our current results may be biased by the

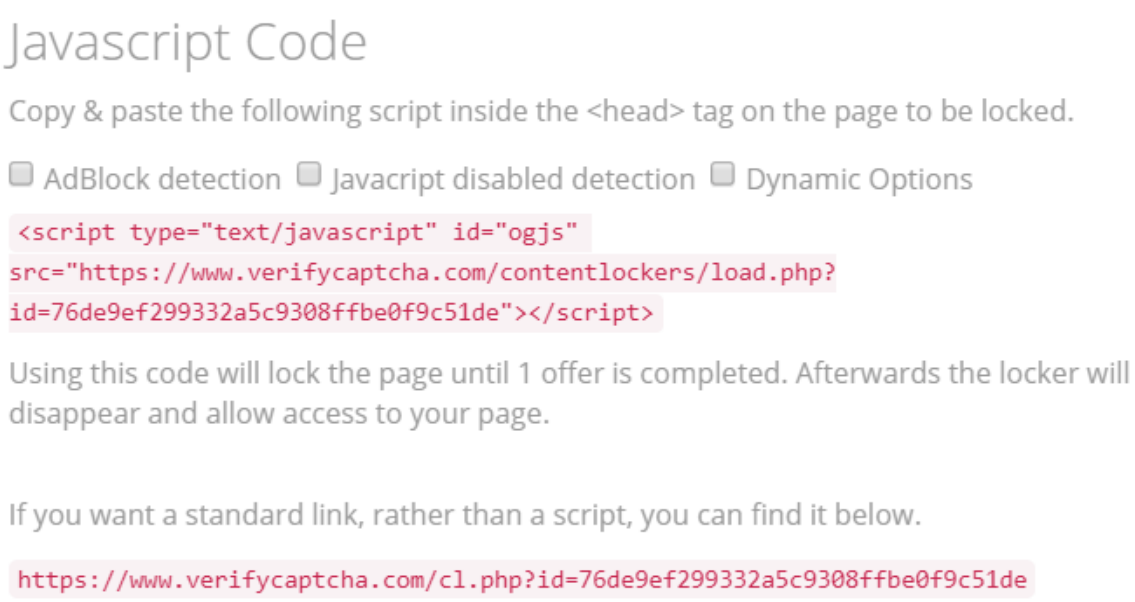

Figure 3.16: *OGADS.com* Locker Types

type of GHS instances we are looking for, and a more systematic search would shed new lights to the situation (for example, other template providers might come to light).

Another limitation is that we are studying the URLs distributed using social media and blogs. Bitly analysis suggests that 71% of the generated traffic is direct through emails and instant messages. This, in turn, suggests that we are missing a big source of URLs distribution. On the other hand, this 71% generated traffic comes from URLs we collected using websites crawling. Thus, although the web traffic is only 29%, it helps in discovering traffic from bigger sources of URLs distribution.

3.8 Conclusions

In this chapter, we investigated what we call the "Game Hack" Scam (GHS). we formulated GHS-related search queries, and used multiple search engines to collect data about the websites to which GHS victims are directed when they search online for various game hacks and tricks. We looked at the pages returned directly by the search engines, as well as the pages linked from these pages. We also investigated the modified APKs, and the executable files collected when searching online for the game hack.

Our data collection spanned a year; in that time, we uncovered 65,905 different GHS URLs, mapped onto over 5,900 unique domains. We were able to link several attacks to attackers and found that they routinely target a vast array of games. Furthermore, we

find that GHS instances are on the rise, and so is the number of victims. Our low-end estimation is that these attacks have been clicked at least 150 million times in the last five years. Additionally, in keeping with similar large-scale scam studies, we find that the current public blacklists are inadequate and suggest that our method is more effective at detecting these attacks.

Finally, we found that more than 90% of the GHS related executable files are flagged by at least five antivirus scanners in virus total. For the modified Android games APKs, 47.5% are flagged by at least five antivirus scanners in virus total. Furthermore, some of these games are not working, some of them have changed the in-game advertisements, and some of them have changed the game completely.

All the data used in our study is available at `http://ssrg.site.uottawa.ca/ghsjwe/`.

Chapter 4

The Bitcoin Generator Scam

4.1 Introduction

In this chapter, we extend our work and utilize the automated system -Chapter 3- to analyze the "Bitcoin Generator Scam" (BGS), which is another web-based cyberattack that promises to provide victims with free cryptocurrencies in exchange for a small mining fee. We have started our work by creating a BGS dataset from an initial manual search and by using blacklisted domains [76], `cutestat.com`[1], and the Internet archive[2]. We used this dataset to train a classifier to recognize BGS instances. We have then generated hundreds of search queries related to BGS and used them daily on popular search engines for 21 months. Once we identify a new BGS instance, we interact with it to extract the bitcoin address(es) used to accept money from the victims.

In the 21 months of our crawling, we have discovered more than 1,200 scam domains and more than 9,000 bitcoin addresses associated with them. These addresses have received more than 9.5 million dollars, with an average of 49.9 dollars per transaction. Since our approach is not based on existing transactions, we can detect scam addresses before receiving any money. Indeed, over 70% of the online addresses that have eventually received funds in our study were detected before receiving any transactions.

Our main contributions on studying the BGS are the following:

- We validated our model by utilizing it to give insight into a new type of scam that targets cryptocurrency users.

[1] https://cutestat.com/
[2] https://web.archive.org/

- We uncovered more than 1,200 BGS-related second-level domains and 9,000 cryptocurrency scam addresses.
- We were able to detect scam cryptocurrency addresses before receiving any funds from the victims.
- We exploited domain-related and addresses-related features to cluster BGS websites and their related addresses into campaigns operated by the same scammer.
- We compiled and disclosed a big dataset that contains scam domains and cryptocurrency addresses for public use.

The remainder of this chapter is structured as follows. After this introduction, in Section 4.2, we apply our methodology to detect the BGS. in Section 4.3, we report some basic numbers obtained during our crawling period. in Section 4.4, we carry out various analyses and discuss the results. in Section 4.5, we present other variations of the BGS attack. in Section 4.6, we present our multi-level clustering technique. in Section 4.8 we discuss some of the main limitations in our model. Finally, we conclude in Section 4.9.

4.2 Applying the Methodology

In this section, we adapt our methodology presented in Chapter 3 to detect, analyze and track BGS instances. Figure 4.1 presents our customized system to detect the BGS.

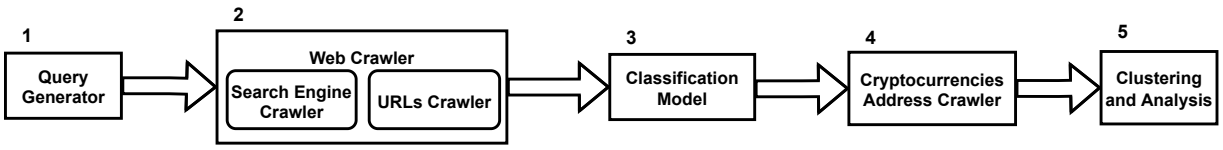

Figure 4.1: BGS detection and analysis model.

4.2.1 Training Dataset Creation

Similar to the GHS, there was no previous large-scale analysis of the BGS that we can use to create our dataset. Thus, we used various techniques to collect this initial dataset:

1. **Search Engines:** We have used the different search engines to collect and label an initial training dataset manually. We have manually searched for BGS instances on

Google. We have used several search queries related to the scam, such as "online bitcoin generator", "generator free bitcoin", and "online bitcoin hack tool". Our search identified an initial set of 52 BGS instances. We also obtained 30 new search queries using Google's automatic "related search" suggestions during this initial collection. This gave us our initial set of queries for starting our automated web crawl.

2. **Third-parties and Blacklists**: Many third-party companies and blacklists collect scam datasets that researchers can utilize in their analysis. For example, Yin and Vatrapu [168] used a dataset provided by `Chainalysis.com`, and Razali and Shariff [126] used Nocoin blacklist[3] in their analysis. In our work, we have used the site **Bitcoin.fr [76]** which contains a list of Bitcoin and cryptocurrency scam domains. The list is a collection of several scam lists, including adcfrance.fr, the House of Bitcoin, CryptoFR, badbitcoin.org, and scambitcoin.com. The list also contains testimonies of the site users. At the time of crawling, the list contained 6,230 domains.

3. **Customised historical search engines:** Many websites host an online service that scans and analyzes other websites and collects information related to domains, websites, usage reports, IP address, host, etc. Some of these websites provide customized search services that report previously scanned domains and URLs with content "related" to the performed search. The reported domains are either currently active or domains that were active in the past. For example, Phillips and Wilde [122] used `urlscan.io`[4] to expand their dataset with other domains related to their initial dataset. In our work, we have used `cuteStat.com`[5], which is a website that collects information related to websites, domains, host, IP address, usage reports, etc.. One of the services that this website provides is a list of up to 100 domains that have content "related" to the search we perform. We have utilized this service to collect 610 new domains that have content related to the search queries collected in step 1.

4. **Archived websites:** Many websites host online services that scan the internet and retain a snapshot of the scanned websites. These services enable web users to access archived and historical versions of the scanned websites. When a URL is submitted to these services, the service will automatically visit and collect data about the browsed URL, such as the domains and IPs contacted, the HTML content, a screenshot of the landing page, and the resources (JavaScript, CSS, etc.) requested from those domains. The **Internet Archive**[6] which is a digital library that provides an

[3]https://github.com/hoshsadiq/adblock-nocoin-list
[4]https://urlscan.io/
[5]https://www.cutestat.com/
[6]https://web.archive.org/

extensive collection of readily available digitized materials, including internet sites, games, music, and public-domain books for free, is an example of such websites. Another example is `urlscan.io` that saves snapshots of the scanned URLs and provides indicators of compromise; it tracks 400 popular brand domains and tries to make a verdict whether the scanned URL is suspicious or malicious if it targets any of the 400 brands. We have used the Internet Archive to collect thousands of snapshots for the set of domains we collected from *CuteStat.com* and *Bitcoin.fr*.

5. **Identifying BGS Instances:** Since the Internet Archive contains thousands of snapshots, and we could not manually check all of them; we filtered the snapshots and only considered the snapshots that contain a bitcoin address in the HTML. This reduced the number of possible BGS domains to only 307, a number we could handle manually. We inspected these domain snapshots one by one and verified that 252 of these domains were indeed BGS domains. The other 55 domains were different types of scams, such as HYIP and bogus charity.

Following these steps, we have collected 304 pages as our initial set of BGS instances. We then manually inspected 400 randomly selected pages that we had collected but not flagged during the first week of operation. Of these 400 pages, 374 were benign pages and 26 were new BGS instances. Therefore, our final dataset consists of 330 BGS pages (304 + 26), complemented with 330 benign pages randomly selected from the set of 374 pages we had.

4.2.2 Search Query Generator

To create BGS search queries, we have used the context-specific corpus technique we used to create GHS-related queries in Section 3.2.2. We also used two new techniques to increase our chances of discovering BGS instances:

- **Search Engines:** We started our work by collecting Google's automatic search suggestions as we manually searched for BGS. We then used these suggestions to create the first set of queries and perform an initial web crawl.
- **The "Keywords" meta tag:** As described in Section 4.2.1, we were able to collect and manually verify 330 BGS instances from our initial web crawling, as well as from a list of blacklisted domains [76], from the site `cutestat.com`, and from the Internet archive. We have extracted the "Keywords" meta tag contents from these instances to augment our original queries. The "Keywords" meta tag represents a

set of a comma-separated list of keywords that are relevant to the web page and used to inform the search engines about its content [8, 13].

- We further augmented our search queries by **utilizing the context-specific corpus.** We have inspected several BGS pages and found that scammers use specific words in the content of a BGS page, such as the name of the targeted currency and words that advertise the ability of the generator to hack the blockchain and provide the victim with the promised cryptocurrencies. For example, the words "bitcoin", "btc", "tool", and "mining" were widely used in the scam pages.

  We have utilized this fact to generate more scam-related queries. We extracted a bag of words from our corpus. We found 834 words that have a frequency greater than or equal to ten. We selected the 157 words with the highest frequency and have a direct connection to BGS. We then generated our queries using the Markov assumption [81] to approximate n-gram probabilities. We generated our n-grams for $n = 3$ to 7[7]. That gave us 527 n-grams, and we manually selected 207 search queries from them.

Our final query list contains 696 search queries[8].

4.2.3 Web Crawler

We use the web crawler to track and collect the BGS instances. We use the previously identified search queries as a seed to search daily for GHS pages using *Google.com*, *Bing.com*, and *search.yahoo.com*. For each query, we only consider the first 20 search results returned by each engine. We extract and crawl the URLs resulting from our searches. We also expand our crawler one more level, in which we fetch and crawl the URLs included in each non-BGS page returned by the search engine. We collect data about the crawled URLs, including URL redirections, HTML contents, a screenshot of the landing page, and the page resources (scripts, CSS files etc.).

4.2.4 Classification Module

To identify the BGS instances from the set of crawled pages, we have used a similar classification model to the one we used in Section 3.2.4. We tested five different classifiers

[7]Our experiments showed that 8-grams and up did not improve our results.

[8]The complete list is available at `http://ssrg.site.uottawa.ca/bgsextended/`.

Classifier	Page type	Classified clean	Classified BGS	F1
SVC	clean	327	3	98.92
	BGS	4	326	
MLP	clean	327	3	98.92
	BGS	4	326	
RF	clean	329	1	95.9
	BGS	25	305	
NB	clean	327	3	96.58
	BGS	19	311	
KN	clean	319	11	97.9
	BGS	3	327	

Table 4.1: Results of a 10-Fold cross-validation with five classifiers.

from Scikit-learn python library [120] on our training set: Linear SVC, NB, KNN, RF, and MLP[9].

To evaluate our classifiers, we used 10-fold cross-validation on the labeled dataset we prepared in Section 4.2.1. We used the five classifiers to classify the crawled pages based on the text as seen by the end-user. More precisely, we have used the TF-IDF of the words displayed to the users to extract the training features. Our classification model achieved good accuracy; we detected more than 98% of the scam pages while misclassifying less than 1% of the benign pages.

We show the results in Table 4.1. As can be seen, SVC and MLP achieved the highest F1 score, 98.92, followed by KN at 97.9. The other classifiers also performed fairly well, with RF having the lowest F1 score. Based on these results, we used the SVC classifier throughout our experiments. We have manually inspected the wrongly classified pages and found that most of these pages do not have enough text to extract the features from.

After using our classifier on newly found pages for a few days, we randomly selected 100 pages classified as benign and 100 pages classified as BGS instances for manual verification. Our model correctly classified 197 of these 200 pages. Two benign pages were misclassified as BGS, which yields a true positive rate of 98%, and one BGS instance was misclassified as benign, which yields a true negative rate of 99%.

[9]We have used the default parameters as described in Section 3.2.4

	Actually clean	**Actually BGS**
Classified clean	99	1
Classified BGS	2	98

Table 4.2: Classifier accuracy on pages that have not been observed in the training phase.

4.2.5 Cryptocurrencies Addresses Crawler

In this section, we use the scam operator crawler to collect the cryptocurrencies addresses that the scammers are using to collect funds from the victims. For this purpose, we interact with the BGS instances, provide the expected inputs, and follow the specific instructions in order to reach the final stage, when the scam address is provided (the fourth image of Figure 1.8). Usually, the fake hacking process requires 5 to 10 minutes on average. During this time, the attacker typically displays a detailed "log" of the hacking process, which is supposed to occur in real-time. This log displays server IP addresses supposedly being hacked, bogus proxy servers names, the ledger's block in which the transaction is supposed to be added, etc[10] (see, for example, Figure 1.8 image 3). However, in some cases, we find the scam address immediately in the HTML of the BGS instance. For these pages, we collect the scam address without further interaction with the BGS instance.

Furthermore, in addition to the "live" crawling, we also crawl the Internet Archive and `urlscan.io`[11] to collect addresses that the instance has used in the past. `urlscan.io` is an online service that scans and analyzes websites. When a URL is submitted to *urlscan.io*, the website will automatically visit and collect data about the browsed URL, including domains and IPs contacted, the HTML content, a screen-shot of the landing page, the resources (JavaScript, CSS, etc.) requested from those domains, technologies used, and cookies created by the page. Furthermore, `urlscan.io` provides indicators of compromise; it tracks 400 popular brand domains and tries to verdict whether the scanned URL is suspicious or malicious if it targets any of the 400 brands. Finally, Some scam websites provide a video tutorial for the scam in action, which we follow up and extract the addresses the scammer uses in the tutorial.

Feeding the BGS Addresses to the Anti-Phishing Work Group (APWG) data warehouse: Our analysis in Section 4.4.2 shows that our system can detect many scam addresses before they are recorded on the blockchain (i.e., before the victims transfer any funds to the scammers). This data is now sent automatically to the APWG[12] eCrime

[10] A complete example of one such log is presented in our public data repository.
[11] https://urlscan.io/
[12] https://apwg.org/

eXchange (eCX)[13] data warehouse in real-time. APWG is an international coalition that unifies the global response to cybercrime, such as phishing and online fraud across government, industry, NGO communities, and law-enforcement sectors. ECX represents a data warehouse containing cyber threat data modules, including thousands of phishing and malicious domains. It also contains more than 70K cryptocurrency addresses used in different types of cybercriminal activities. We hope that feeding the addresses to a blacklist in the early stages will reduce the number of victims.

4.2.6 Clustering and Analysis

The last stage of our model is clustering and analyzing the data from the previous stages to provide insight into the studied scam. For this reason, we conduct different analyses to give more insights into the BGS as follows:

- Evaluating our model effectiveness in detecting the scam bitcoin addresses before the victims transfer any funds.
- Estimate how many Bitcoins the BGS has received and the actual value of these Bitcoins in USD.
- Looking at the possibility of actual payback in the scam by studying a few of the Bitcoins addresses that have actually transferred back to the initial address they received funds from.
- Discussing two techniques that the scammers use to provide the scam deposit addresses that make our analysis harder.
- Studying the cryptocurrencies addresses reuse in our scam domains and other types of scams.
- Clustering the BGS domains and their related addresses to create campaigns operated by the same scammer.
- Investigating the BGS in other languages.

[13]https://apwg.org/ecx/

4.3 Scam Collection and Measurement

Our experiments were run on our university's server as well as on dedicated servers provided by Compute Canada[14].The results reported in this chapter come from data collected from November 2019 to July 2021. In this section, we present some basic numbers obtained directly from our crawler and classifier.

Over the course of our experiments, our system identified 9,140 bitcoin addresses involved in BGS. 3,357 of these addresses have at least one transaction. However, one particular BGS instance is responsible for most of the transaction-less addresses; the domain *bitmake.io* has a hard-coded list of 5,001 addresses, and one of these addresses is selected randomly when a payment is made. At the time of writing, on that particular BGS instance, only 39 of the 5,001 addresses have transactions, so that site alone is the source of 4,962 of the 5,765 transaction-less addresses in our database (that is 86.07% of them). Without that site, around 80% of the addresses have transactions. These addresses have been found on 1,216 unique scam domain names[15].

About half of the BGS domains (689 of them) contain a single payment address. At the other extreme, 71 of these domains (5.8%) are associated with at least ten addresses. We have found 183 addresses that belong to cryptocurrencies other than Bitcoin. 72 are Ethereum addresses, 30 are Litecoin (LTC) addresses, 18 are Bitcoin Cash (BCH) addresses, 18 are Dogecoin (Doge) addresses, and 45 addresses belong to other currencies such as dash and Zcoin. Since the vast majority of the addresses are Bitcoin addresses, we focus on that currency in the rest of our analysis. Finally, our analysis also showed that none of the Alexa top 1K domains[16][17] contains actual BGS instances. Therefore, we only report results for URLs hosted on domains outside Alexa top 1k.

In Figures 4.2 and 4.3, we present the number of BGS URLs and addresses detected per day. We do not include here the URLs and addresses found in the internet archive, in order to only count newly discovered and currently active BGS instances.

On average, our model detected about 2.1 new BGS instances and 3.7 new bitcoin addresses every day from November 2019 to July 2021[18]. These numbers are relatively

[14] `https://www.computecanada.ca/research-portal/`

[15] In general, we only consider second-level domain names when comparing scams URLs, excepted for hosting services, for which we consider the third-level domain name. So *generatorbitcoin.epizy.com* and *miningbtc.epizy.com* are counted as two separate attacks even though they are on the same second-level domain name because they are both using the hosting service *epizy.com*.

[16] `https://www.alexa.com/`

[17] However, we include the hosting domains and the public bloggers in our analysis

[18] Note that a new BGS instance does not necessarily mean a new address since there are some addressed that are shared among instances

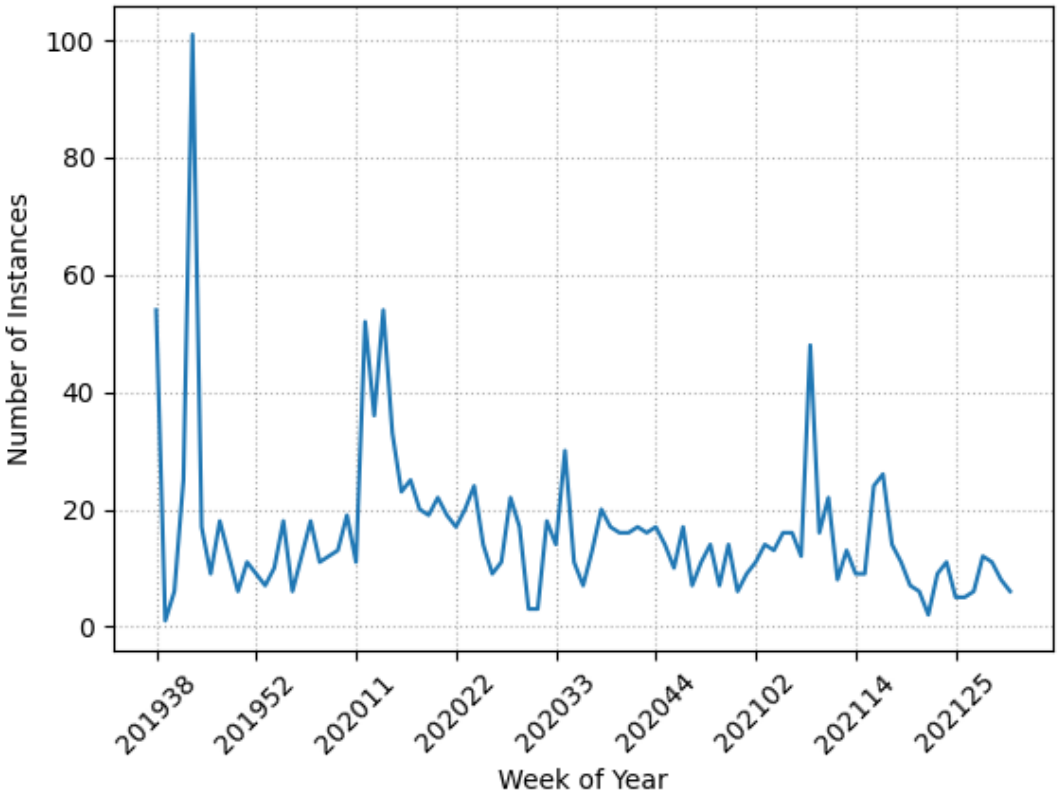

Figure 4.2: Number of BGS URLs detected per week.

stable throughout the period. Therefore, we can extrapolate that our system will identify more than 700 new BGS instances and more than 1,300 bitcoin addresses per year.

4.4 Analysis

In this section, we use our database of bitcoin addresses to estimate how much money was stolen through the BGS. We also look at the few cases in which bitcoins were actually transferred back to the initial address. We discuss a couple of techniques used by scammers that are making systems like ours less effective. We present basic statistics of the BGS domains and addresses. Finally, we discuss the addresses reuse in the BGS domains and other types of attacks.

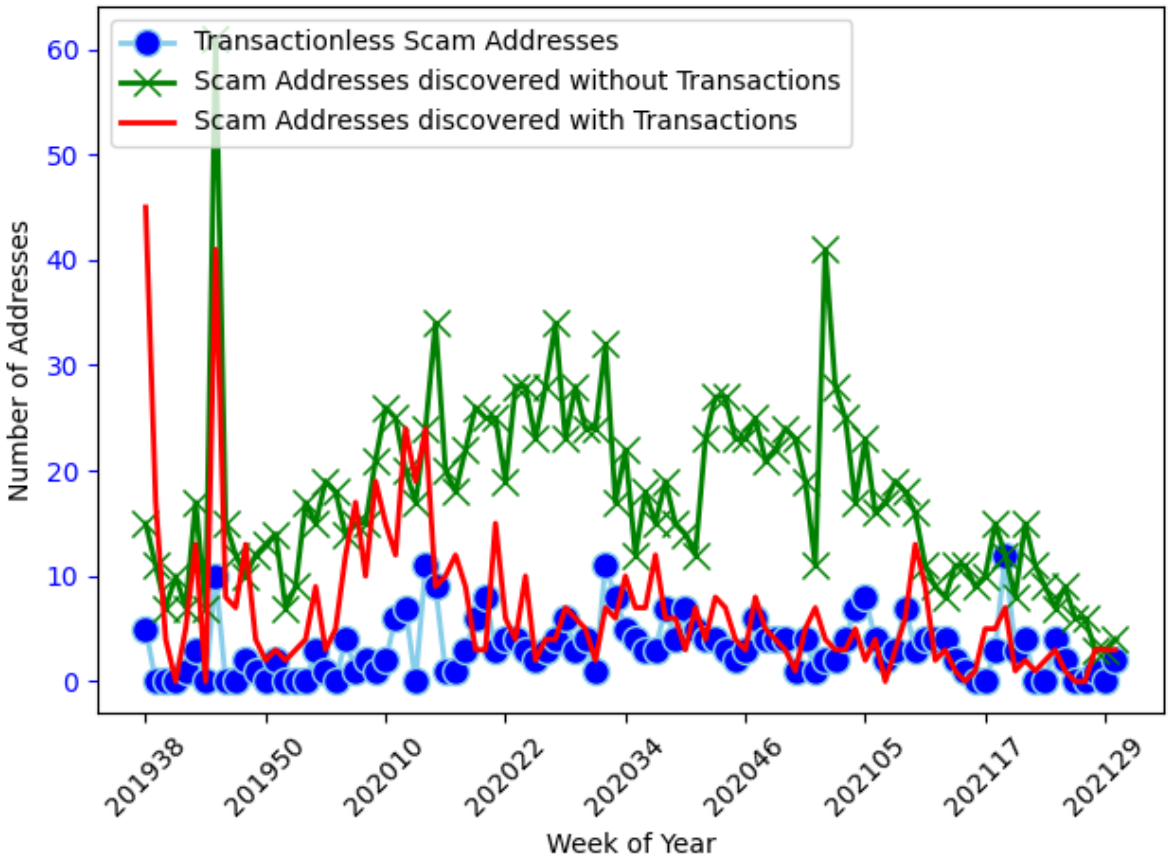

Figure 4.3: Number of Bitcoin addresses detected per week.

Figure 4.4: Word cloud based on the text contents of the gathered technical GHS pages

4.4.1 Page Contents

Scammers use specific words in the content of a BGS page, such as the name of the targeted currency and words that advertise the generator's ability to hack the blockchain and provide the victim with the promised cryptocurrencies. We will use these words to extract distinguishing features and use them as a pre-classification filtering step. We believe that the features will reduce the classification model execution time and increase its accuracy, as in the case of GHS (see Section 3.2.4). Figure 4.4 shows the most frequent words used in the BGS pages in the form of a word cloud, where the size of each word correlates with the number of times it appears in the corpus of BGS pages.

4.4.2 Crawler Effectiveness

In this section, we discuss the ability of our crawler to detect scam addresses before it receives any transactions. We collect scam addresses in two ways: first, we revisit daily all the BGS instances that we have previously discovered. Therefore, if an instance publishes new addresses, our system will pick them up within 24 hours. We also look at other sources, such as the Internet archive, data published by *urlscan.io* and tutorial videos published by scammers. That way, we collect some of the addresses that have been used in the past,

before we discovered the instance. Our database is thus a mix of currently active addresses and addresses that have been active months or years ago.

Overall, we have discovered 3,357 bitcoin addresses with at least one transaction. 2,253, of these addresses (67.11% of the total) were detected by the online crawler and did not exist in the other sources. 938 of these addresses (27.94% of the total) have been extracted from the other sources but were never found by our live crawler. Finally, the remaining 166 addresses (4.94% of the total) have been found both by our live crawler and on the other sources.

1,743 of the 2,419 addresses found by our live crawler were found before they had any transaction; transactions eventually arrived (recall that we are here only looking at addresses that have eventually received transactions), but only after the address has been flagged by us. That is one of the unique strengths of our model, the ability to detect a suspicious address before it receives any funds. The percentage of the addresses that we discover before receiving transactions increased with time. The current value is 72.05% of the eventually active addresses being discovered before any transaction were received, compared to 55% reported in our previous work.

4.4.3 Bitcoin Addresses Payment Analysis

We first measure the scale of BGS by analyzing the transactions involving the bitcoin addresses that we have found. Overall, we have collected 3,357 addresses with at least one transaction. These addresses have received 192,551 transactions from 313,856 unique addresses. On average, the addresses have received 0.017998 bitcoin per transaction, accumulating 3,465.38 bitcoins overall.

We then used the average exchange rate of the day of the transaction, obtained from *bitcoincharts.com*, to convert the value of the transactions to USD. In total, the addresses have received 9,610,109 USD. The transactions occurred between September 2011 and July 2021, when this analysis stops (the attack is meanwhile still active at the time of writing)

The total number of transactions and their corresponding total value in USD, compared to the exchange rate of bitcoin, are presented in Figures 4.5 and 4.6 respectively. As shown in the figure, there is a clear correlation between the success of BGS and the market value of bitcoin, which is certainly not surprising. Additionally, BGS is still going steady and continues deceiving victims daily. As can be seen in Figure 4.7, the scammers tend to transfer the funds away from the receiving address as soon as they are received.

Finally, if we try to estimate the accuracy of our numbers, in our analysis, we have reported the number of addresses and instances without extrapolation (e.g., without using

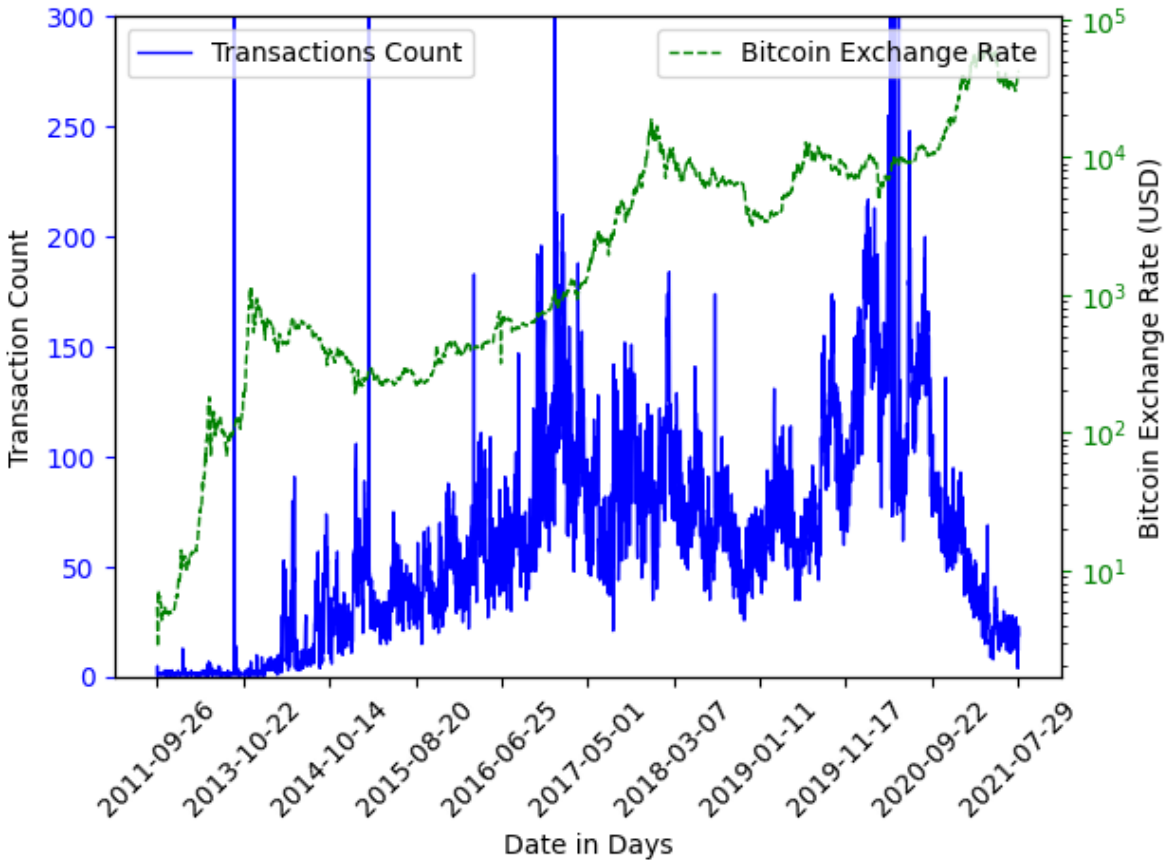

Figure 4.5: Daily incoming transactions to BGS addresses.

clustering techniques such as multi-input heuristic algorithm [127]). However, since we certainly did not detect all of the scam addresses, the actual number of addresses and instances are underestimated. As for the total value of received dollars, we cannot distinguish between the payments made to the scam and the payments made to the scam address for some other reasons; thus, our analysis might be overestimated, especially when it comes to early transactions.

4.4.4 Payback Analysis

In some types of attacks, such as the Ponzi scheme, scammers provide some pay back to some of the victims as part of the scheme. However, with the anonymity of the bitcoin, it is unclear if paying back to entice the victims to invest more in the scam or attract new victims to invest is effective. Although some of the scam addresses have sent some money back to addresses from which they have received payment, our analysis did not show evidence that there is an actual payback in the BGS attack.

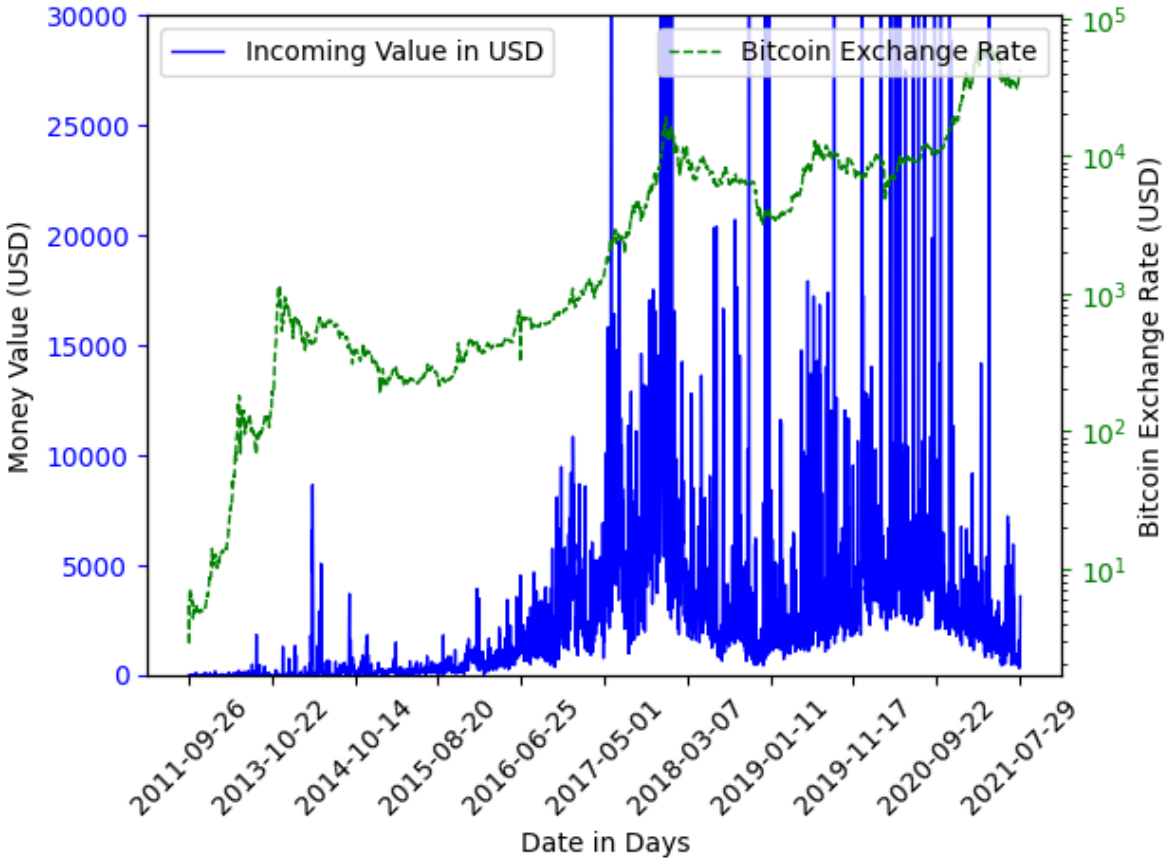

Figure 4.6: Daily deposited money to BGS addresses.

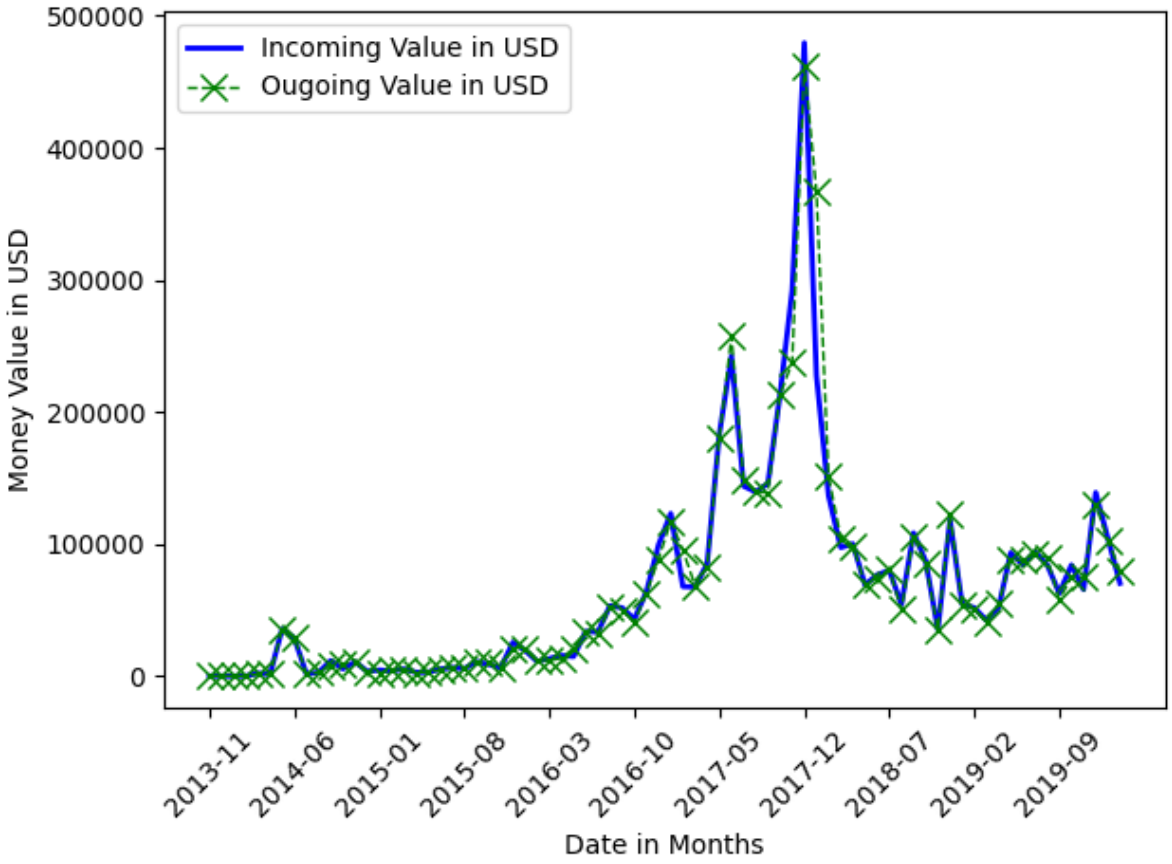

Figure 4.7: A comparison between the incoming and outgoing deposits in BGS addresses.

In this section, we look at the subset of the addresses that sent bitcoins to the scam addresses and received anything back from the same address[19], the way it would be if the scam was working as advertised.

Out of the 313,856 addresses that have sent bitcoins to the scam, 1,055 addresses have both sent and received bitcoins from at least one scam addresses. Overall, 152 scam addresses (4.52% of the 3,357 scam addresses) where implicated in these back transactions. Naturally, we would like to understand if those 1,055 addresses received money back from the scam or if in fact both addresses belong to the scammer. To do that, we look at the transaction history of the scam addressees and further divide the 1,019 addresses into two groups:

Scam Addresses: 52 of the 1,055 addresses are in this group; they are scam addresses that were already identified as scam addresses, and belong to our BGS dataset. Thus, they are internal transactions in the scam and do not represent payback.

Normal Addresses: 1,003 of the 1,055 addresses belong to this group. We have no concrete evidence that the scammers control these addresses (which certainly does not mean that they are not). These addresses have collectively sent 167.52 bitcoins (138,191 USD) to the scam and received 202.45 bitcoins (213,196 USD). However, four of these addresses has sent 0.026 bitcoin and received back 28.7 bitcoins, so excluding these outliers, on average, these addresses have received more or less what they have sent. Although at this time we cannot conclusively prove that these addresses are an integral part of the scam, we can at least state that overall, they do not impact or change our general results, as illustrated in Table 4.3.

4.4.5 Scam Addresses Delivery Techniques

In this section, we look at two techniques that the scammers use to provide the scam deposit addresses that make our analysis harder. First, many of the domains are regularly changing bitcoin addresses during their lifetime. Second, some of the domains generate a unique address for each victim. We are not sure of the underlying intent of these techniques. However, it limits automated systems like ours to find the BGS domains and extract their addresses. It is not surprising if the scammers use these methods to prevent detection and extend the lifespan of their attacks.

[19] A limitation of this analysis is that we consider the payback that is made to the same address from which the address received from. This is not necessarily the case in a bitcoin transaction [1], and we would miss the hypothetical transactions for which this is not the case. However, the scam instances that we inspected did not include any other way to get payments back.

	Total scam	Addresses that received some payment back		Addresses that did not receive anything
		Scam	Normal	
#addresses	313,856	52	1,003	312,801
#inTransac	192,551	290	2,374	189,887
#outTransac	97,357	371	4,106	92,880
#inBTC	3,465.38	17.29	167.52	3,280.56
#outBTC	3,428.43	19.87	202.45	3,206.11
#inUSD	$9,610,109	$15,857	$138,191	$9,456,060
#outUSD	$10,273,760	$16,968	$213,196	$10,043,595

Table 4.3: Detailed analysis for the scam addresses payback (Transac refers to transactions).

Regularly changing the bitcoin address. 527 of the BGS domains that we have found (that is 43.33% of the domains) have used at least two different addresses. Furthermore, we have found at least 10 addresses in 71 of these domains (5.83% of the total). The domains with the highest number of addresses have, respectively, 5,001 addresses, 236 addresses, 213 addresses, 143 addresses, and 124 addresses that we know of. In some cases, the scam address presented to the victim is selected randomly from an array of static choices (For example, see Figure 4.8). As previously explained, the most extreme case that we have detected is *bitmake.io*, which contains a list of 5,001 addresses, but only 39 of these addresses have any transactions. Of course, periodically changing the addresses reduces the number of transactions per address, making it more difficult to detect using techniques based on transaction history. In fact, as can be seen in Section 4.4.6 , around 50% of the addresses have received 1 or 2 transactions.

Distinct address per victim. In some cases, the BGS instance generates a unique scam address for each victim. We have detected 8 domains that use such a technique. We continuously crawled these domains and found that the attacker generates a unique address for each deposit address the victim is using. As a result, none of the addresses detected by these domains have any transactions. One consequence of this is making the attack hard to be detected by detection systems that depend on transaction history.

To further study these domains, we have manually searched on Google and YouTube to find addresses related to these domains that have transactions. We have only found 5 bitcoin addresses with funds related to the domain *doublebitcoin.win* from 3 YouTube re-

```
// ============================================
// Bitcoin Crypto Finder Address Database
// ============================================
var Address=new Array() // do not change this
Address[0] = "<input type='text' class='form-control' size='38' onClick='this.select();' class='depositAddy'
value='1NXDDChyj2qgRYFzCqJiXNkPixZdY1uXYz' readonly>";
Address[1] = "<input type='text' class='form-control' size='38' onClick='this.select();' class='depositAddy'
value='1Jk9HD76cFc4eMSJokm2DN7G3dH673qYke' readonly>";
Address[2] = "<input type='text' class='form-control' size='38' onClick='this.select();' class='depositAddy'
value='1DE9ZrWvqjWdyVAzQBxvaYBWQG5zcgGhjz' readonly>";
Address[3] = "<input type='text' class='form-control' size='38' onClick='this.select();' class='depositAddy'
value='1PKK6bsVFDimNiMSYkXThRkEdh7fL3DZwz' readonly>";
Address[4] = "<input type='text' class='form-control' size='38' onClick='this.select();' class='depositAddy'
value='172pjGKt6sbojVoJfvJ4Zv7QYRLagSWbTJ' readonly>";
// ====================================
// Do not change anything below this line
// ====================================
var Q = Address.length;
var whichAddress=Math.round(Math.random()*(Q-1));
function showAddress(){document.write(Address[whichAddress]);}
showAddress();
```

Figure 4.8: A real world example of a BGS instance in which the payment address is selected randomly from a list.

views published by a scam researcher[20]. Each of the addresses received a single transaction with a total of 0.026579 bitcoin. The transactions are related to the review provided by the researcher. We investigate the connections between the 5 addresses in Appendix C.

4.4.6 BGS Addresses Statistics

In this section, we report some basic numbers about our scam addresses transactions history:

- The addresses active lifetime, defined as the number of days between the first and last incoming transaction;
- The longest period during which an address was inactive, counted in days;
- The fraction over time of the total number of transactions received from the day of the first transaction to the day of the last transaction;
- The number of days a scam instance was active, defined as the number of days between the day of the first address discovered in the domain and the day of the domain becomes inactive. For active domains, we count until the day of the analysis.

Of course, these numbers are biased by the end of our experiment, since addresses and scam instances are still active afterward.

[20] https://www.youtube.com/watch?v=oic8YfMge2g

Fraction of data	Addresses active life time	Addresses longest inactive time	Transactions per address	Domains active life time
0.1	2	1	1	24
0.2	5	2	1	62
0.3	11	4	1	102
0.4	24	8	1	130
0.5	58	16	2	163
0.6	125	32	3	210
0.7	240	63	7	294
0.8	419	109	18	398
0.9	816	235	55	652
1	3733	1997	1688	2360

Table 4.4: General statistics

Table 4.4 shows our data. We can see that around 40% of the addresses have lived at most a month; on the other hand, 40% of them have lived at least a hundred days. For the number of days an address was idle without receiving any transaction, most of the addresses have a short inactive time; around 70% of the addresses were inactive for less than 63 days. On the other hand, around 10% of the addresses were idle for more than 235 days at some point. Finally, the majority of the addresses have received a low number of transactions. Around 90% of the addresses have received at most 55 transactions, and 50% of them have received less than 3 transactions.

In the case of BGS domains, they have a relatively long life span. Less than 20% of the domains have lived less than a month, and around 70% of the domains were active for more than hundred days. The average lifetime of BGS domains is 271 days, and the median lifetime is 163 days[21]. Comparing the BGS active time to other cryptocurrencies related scams such as HYIP schemes, we see that the active lifetime is much higher. Vasek *et al.* [153] have reported that the median lifetime of the bitcoin HYIP scheme is 37 days and the bridge HYIPs[22] is 125 days.

[21]Since some of the domains are still active, the average lifetime of the domains may be underestimated. On the other hand, as we are using historical data in our analysis, we can't verify if the domain was used for other purposes during it is active time or it was inactive for some period of time; thus, our analysis might be overestimated.

[22]A HYIP schemes that first start as traditional HYIP attack before being used in the bitcoin ecosystem through posts on `bitcointalk.org`

Reference	#addresses	#inCommon	Year	Crime type	URL
[148]	1,246	0	2018	HYIP	https://bit.ly/3nLcB9E
[21]	52	7	2018	HYIP	https://goo.gl/ToCho7
[26]	3	0	2018	Ransom	Hardcoded in the paper
[44]	126	0	2018	Ransom	Hardcoded in the paper
[150]	2,026	1	2018	General	https://goo.gl/sQJKdx
[146]	1,853	0	2019	Honeypot	https://honeybadger.uni.lu/
[149]	1,566	0	2019	HYIP	https://goo.gl/k5PCOZ
[20]	182	0	2020	HYIP	https://goo.gl/CvdxBp
[122]	3750	84	2020	General	https://cryptoscamdb.org/scams
[56]	2,179	0	2020	General	https://bit.ly/32pmC2A

Table 4.5: Crosschecking the BGS dataset with other public datasets

4.4.7 Addresses Reuse

In this section, we look at the addresses reuse. We first investigate BGS addresses reuse in our scam domains database. Second, we crosscheck our addresses with public datasets maintained by other authors.

Our analysis has shown that some addresses have appeared in different scam domains. Overall, we have identified 266 addresses that were used in more than one scam domain. Twelve of these addresses have been used in more than five domains, and the most reused address has appeared in ten domains. We cannot ascertain the underlying intent of reusing addresses, but it may help convince victims to transfer funds to the scam: since bitcoin transaction history is publicly available, an address with a history of receiving and sending transactions, or a large balance may convey more credibility. For example, in one of the BGS instances[23], the attacker advertised the ownership of a bitcoin address with a high fund. The attacker claimed that the address is maintained to pay back the received funds from the victims.

In our second analysis, we look at address used both for BGS and other types of scams. For this purpose, we crosschecked our addresses with 10 public datasets maintained by other authors, which we collected in [15]. None of these databases are about BGS. Yet, as shown in Table 4.5, 92 BGS addresses have been found on 3 datasets. This suggests some level of address reuse across different types of scam attacks

[23] `https://pastebin.com/sf0vMVAE`

4.5 Other BGS Cases

In this section, we discuss two other types of the BGS that we found through our analysis. In the first type, the victim is asked to install a mining executable file on their machine. In the second type, the victim is asked to complete one or more tasks instead of paying the mining fees.

4.5.1 Malicious Executables

In some cases, the victim is provided a software that can supposedly hack the blockchain. In this case, the attacker provides an executable file. During our analysis, we have collected 15 executable files all targeting the Windows OS.

We scanned the 15 files using virus total[24]. Virus total scans any file or URL with over 70 antivirus scanners and URL/domain blacklisting services. An example of the results returned by virus total is presented in Figure 4.9. All of the 15 files were flagged by at least one antivirus scanner, and 10 (66.66%) of the files were flagged by at least 5 scanners. Many of these files were flagged by avast[25], avg[26], bitDefender[27], and kaspersky[28]. Traces for Trojan, Malware, Bitcoin miner, Coin miner, Dropper, and Adware were reported.

4.5.2 Click per Action (CPA) Scam

In other cases, when the victim provides the information needed by the generator and the success message is displayed, a new screen is shown to the user, asking to complete one or more "offers" for verification purposes (as shown in Figure 4.10). This screen is called a "content-locker" (CL) by the creators of these scams. The "CL" with its set of offers is what the scammer ultimately wants the victim to see in this type of BGS attack. These so-called offers represent the final payload and include, but are not limited to, clicking through endless "surveys", filing out "market research" forms, collecting personal information, getting the victims to subscribe to questionable services, installing suspicious executable files on their machines, etc.. An example of a subscription offer is presented in Figure 4.11. This attack variation is similar to the "Game Hack" scam, which we investigated in Chapter 3. Both attacks use similar templates and lead to the same final payload.

[24]https://www.virustotal.com/gui/
[25]https://www.avast.com/en-ca/index#pc
[26]https://www.avg.com/en-ca/homepage#pc
[27]https://www.bitdefender.com/
[28]https://www.kaspersky.ca/

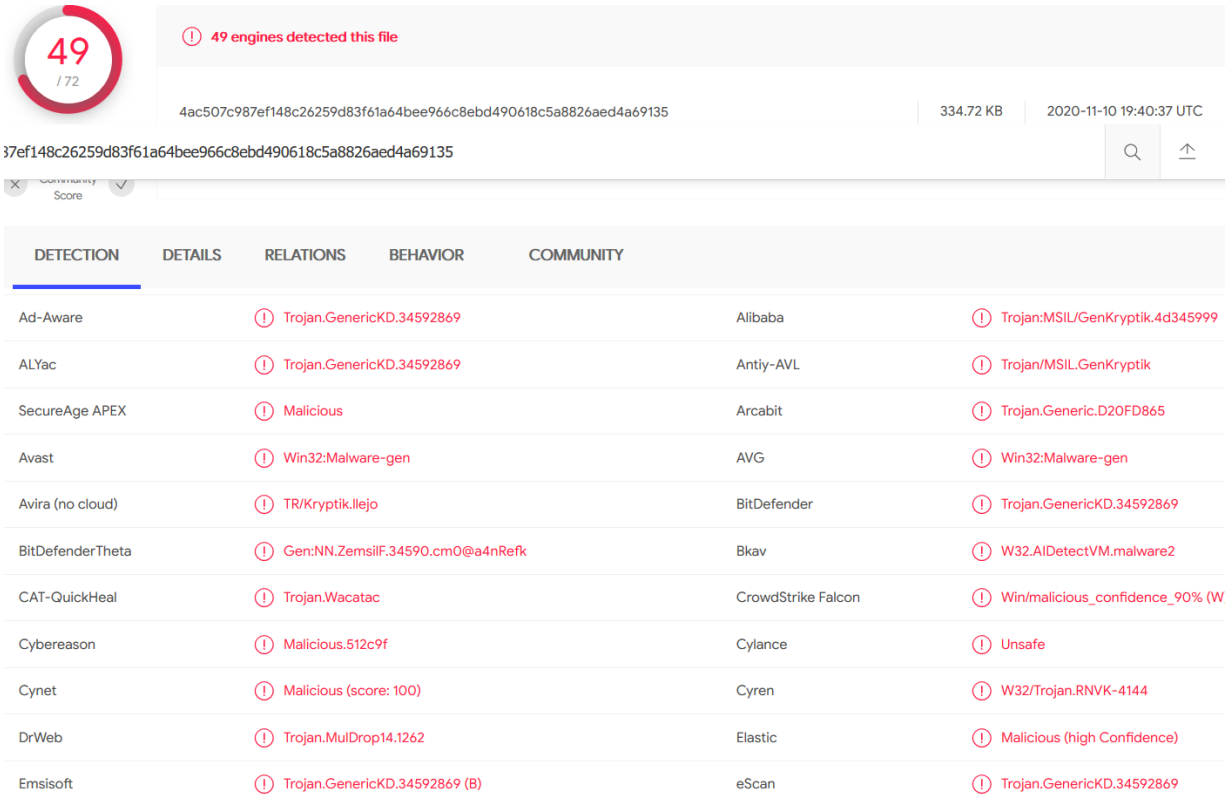

Figure 4.9: An example of virus total scan results.

Figure 4.10: An example of the scam content locker.

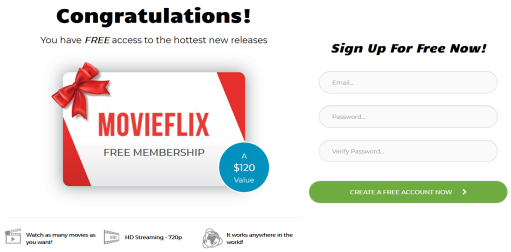

Figure 4.11: An example of the scam offers.

Our dataset contains 49 (4% out of the 1,216) domains that present this kind of offers as the final verification process. Five of these domains used a mix of offer verification and mining fee to collect the fund.

4.6 Scam Clustering

In this section, we attempt to cluster the BGS websites and their related addresses into campaigns operated by the same scammer. For this purpose, we use a variety of features and identifiers. We use website-related features that were deemed suitable for illicit websites clustering in previous work [122] and addresses-related features that we extract from the blockchain transactions history. Some of the website's features have been used individually (for example [157, 159]), and some are combined (for example [122]) to group websites. To the best of our knowledge, this is the first time all these features have been applied together to infer campaigns operated by the same scammer.

1. Level 1 **groups per domain**: this is a direct grouping feature in which we consider all the addresses within the same website to be controlled by the same scammer.

2. Level 2 **addresses reuse**: our analysis in Section 4.4.7 have shown that some attackers use the same address in different BGS domains to carry out their attacks (we have 266 addresses that have been detected on more than one BGS domain). In this level, we merge two clusters when they have some common addresses.

3. Level 3 **analytic/tracking ID**: In some of the scam instances, we found the signature of online advertisement and statistics websites. When using such services, identifiers have to be embedded in the DOM of the sites so that the service can track that particular site. In some cases, people reuse the same identifier across different site,

either on purpose to aggregate the results, or simply by mistake. Separated sites having the same identifier can be suspected to belong to and operated by the same entity [17, 18]. Some of these identifiers relate to third-party analytic services, such as the sites *histats.com* and *statcounter.com*. However, it does not mean that either *histats.com* or *statcounter.com* have any part in the scam, merely that scammers tend to use these sites for their analytics. Other identifiers commonly found in the DOM of the scam instances are related to the sites that provide the scam templates and offers at the end of the scam. Other researchers [122, 136], have shown that google analytic ID can be used to cluster separate illicit websites into campaigns.

These identifiers often require a user account ID to be placed within the DOM of scam instances. Finding matching identifiers account IDs in the DOM of seemingly unrelated websites suggests that the same owner owns them. In this level, we merge two clusters if they have domains that contain the same identifier ID.

4. Level 4 **IP address**: the same IP address can serve the content of numerous domain names. Being hosted on the same IP address has been used as a feature to link illicit websites to the same scammer [122]. In this level, we merge together clusters if they have domains hosted on the same IP.

5. Level 5 **fund transfer between scam instances**: In this level, we merge two clusters A and B together if addresses from A appeared in the input side and addresses from B appeared in the output side of the same transaction.

We provide an overview of each clustering level's outcome in Figure 4.12. In the figure, we present the number of clusters at each level and the two clusters with the highest value in USD. For each of the two clusters, we show the number of domains, number of cryptocurrencies addresses, the incoming value in USD, what caused the merge between the clusters from previous levels, and a sample of the domains that caused the merge and what type of connections between them. The cluster number in the class represents a numeric value to distinguish between the different clusters.

Using our clustering method, we could connect different scam attacks and link them to the same scammer. Our results show that a small group of scammers controls the majority of the received funds. The top two clusters[29] have received around 6.2M USD, which is more than half of the total funds received by the scam addresses.

In order to view the relations between domains and addresses, we build what we call the domain/addresses connection graph. Specifically, to connect the nodes we use the features

[29]We suspect that one or two groups of scammers control these clusters

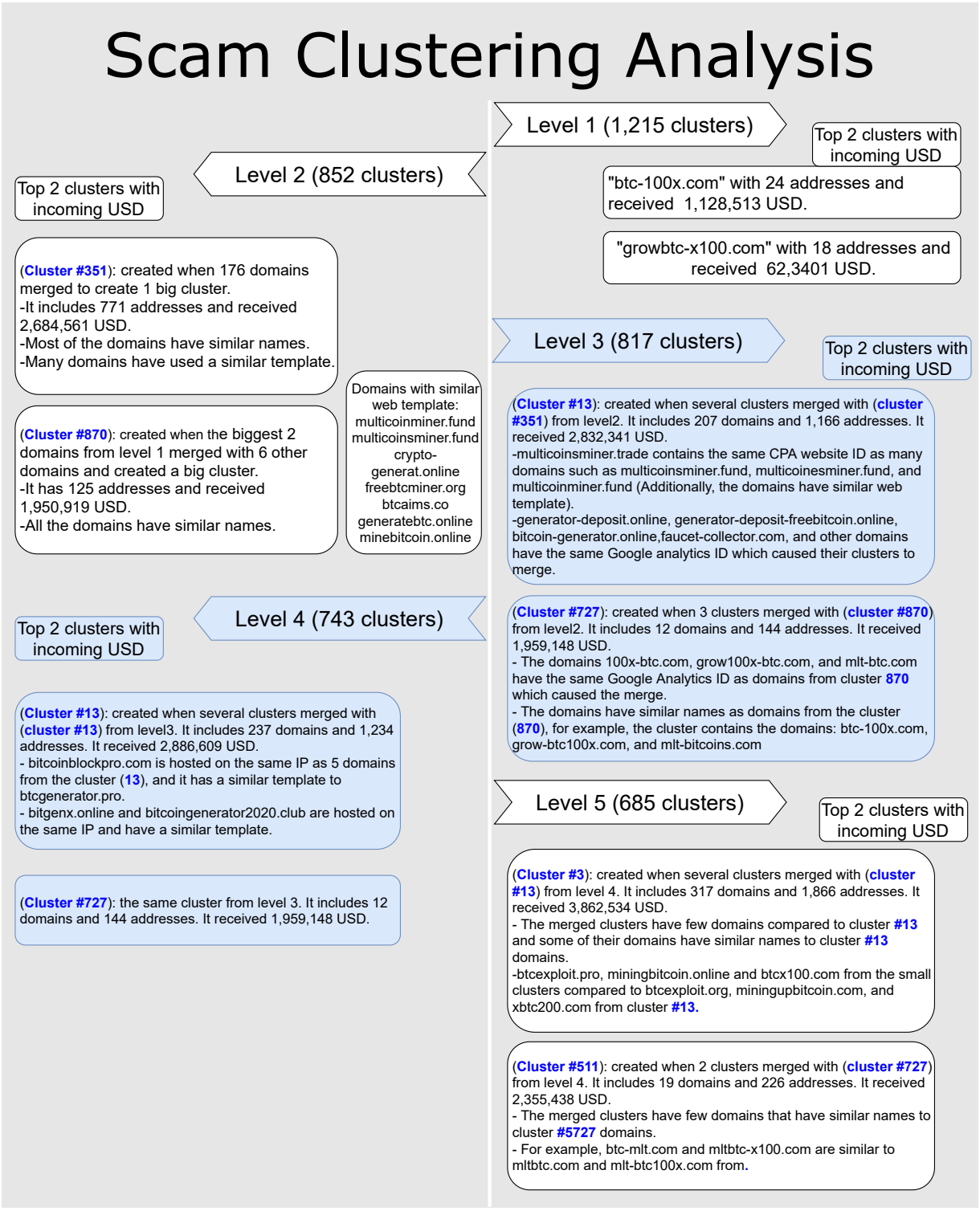

Figure 4.12: BGS addresses clustering analysis.

that caused the merge at different levels. The domain/addresses connection graph of one of the top two clusters from level 5 is shown in Figure 4.13. The red nodes represent the domains, the black node represents the addresses, and the edges represent the connections created during the clustering process. In the graph, an edge is created between a domain and an address if the address is found in the domain DOM. The green edges connect between the domains that have the same analytic/tracking ID (generated from level 3). The red edges connect the domains hosted on the same IP (generated from level 4). The blue edges connect the addresses that transferred funds to each other (generated from level 5). The edge size correlates to the number of transfers between the addresses. An interactive domain/addresses connection graph of the top 2 clusters can be accessed at `https://ebadawi.github.io/level5_1/` and `https://ebadawi.github.io/level5_2/` respectively.

4.7 Investigating BGS in Other Languages

In our analysis, we have trained our classifier on pages with English text only. Thus, we focused our research on pages with English text. In this section, we use a text-independent classifier to investigate if we can find any evidence of significant BGS attacks in other languages.

For this purpose, we have expanded the 696 search queries generated in Section 4.2.2 to include non-English queries. We have used Google translator[30] to translate the 696 queries into different languages, which are, English, Hindi, Spanish, French, Ukrainian, Russian, Chinese, and Swahili. We have targeted the 5-most spoken languages[31] and the languages spoken in the top 5 countries with the highest cryptocurrency adoption index[32]. We have used our model to crawl the extended search queries for 2 days, in which we have collected 14,825 pages identified as non-English pages.

To detect the presence of BGS instances in these pages, we have identified 4 non-language dependent features from the BGS instances to train a classifier:

- **The presence of cryptocurrency address:** This feature checks the existence of a cryptocurrency address within the HTML page content. We look for the pattern of 16 cryptocurrencies that we have observed during the first part of our analysis.

[30]https://translate.google.ca/
[31]https://www.visualcapitalist.com/100-most-spoken-languages/
[32]https://markets.chainalysis.com/#geography

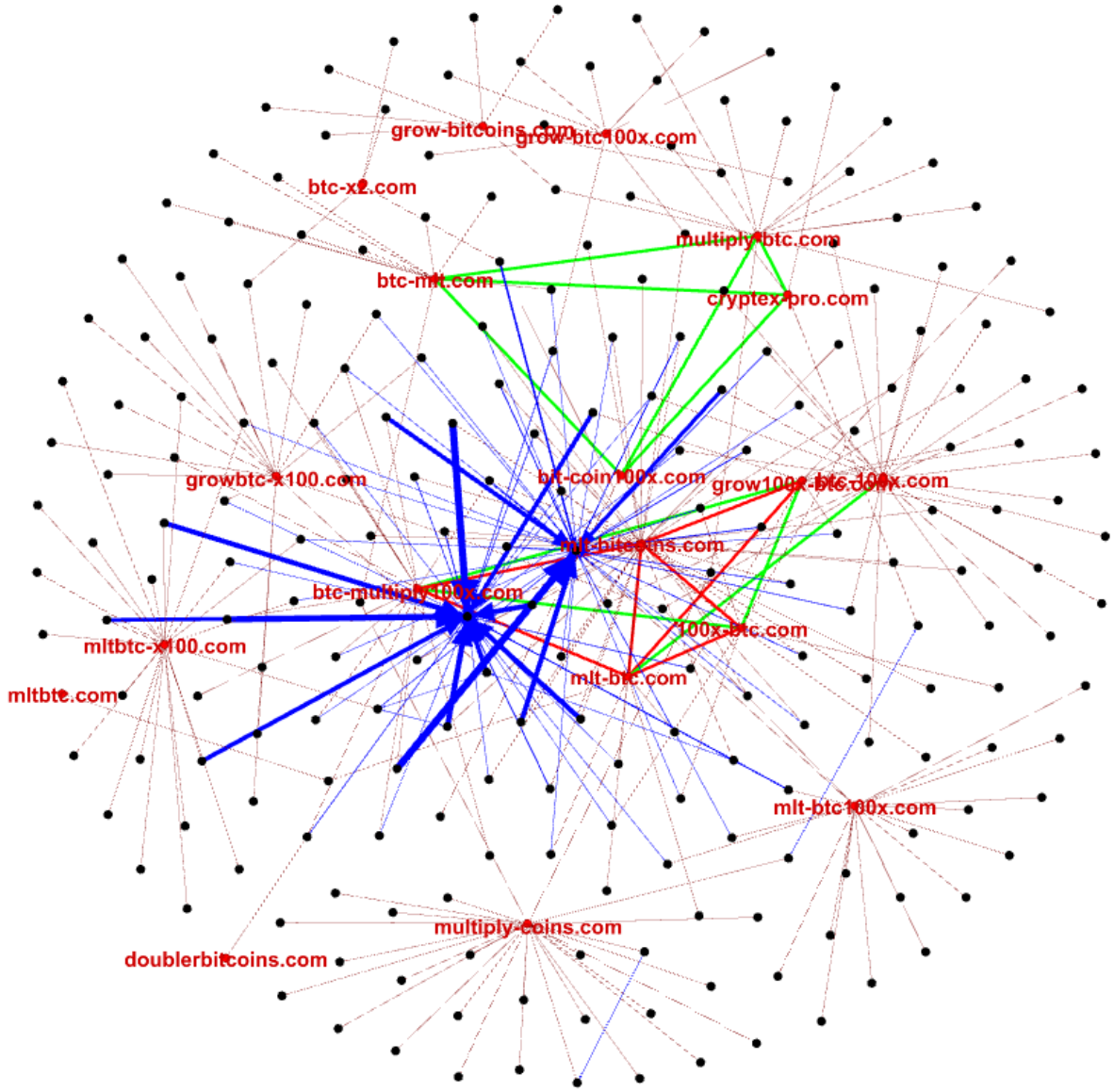

Figure 4.13: Cluster #511 BGS domains and addresses connection graphs (generated by Gephi using layout Fruchterman Reingold).

- **Domain name:** This feature checks for the existence of terms related to cryptocurrency or the scam in the domain name. For example, we have observed the terms "btc", "bitcoin", "generate", and "invest" in many of the scam domains.
- **The presence of input fields:** BGS instances usually contain an input field to accept the victim's address to deposit the proclaimed generated coins. For this feature, we simply look for the tags related to buttons. We include the tags <input>.
- **The presence of buttons:** BGS instances usually contain a button to initiate the pretend generation process. For this feature, we simply look for the tags related to buttons. We include the tags <button>, the tag <input> when the type is "button", and any other tag with "class" or "id" related to buttons.

We have used these features to train five machine learning algorithms from the Scikit-learn python library [120]: KN, Neural Networks (NN), SVC, RF, and NB. To evaluate the classifiers, we used 10-fold cross-validation on the labeled dataset we prepared in Section 4.2.1. We evaluate the performance using the Area Under The Curve (AUC): the greater, the better. The results are shown in Figure 4.14, we can see that all the classifiers perform well, with RF having the greater area.

We then used the RF classifier to classify the set of 14,825 non-English pages we had gathered. 14,770 of the pages were classified as clean, while 55 pages were classified as BGS instances. We manually inspected the pages classified as BGS instances and verified that 9 of them are scam pages. 6 of these pages are English pages that were wrongly identified as non-English by our language detector[33]. The other 3 pages are BGS instances with non-English text. We have also inspected 50 randomly selected pages that were classified as clean and verified they were classified correctly.

Our features classifier is not perfect, and in particular is not as effective as our text classifier. However, this experiment seems to conclusively indicate that the bitcoin generator scam is carried out mostly in English[34]. Therefore, we are confident that our English-only study is in fact very representative of the attack at a whole, at least at the time of writing.

4.8 Limitations

One of the main limitations of our study is that we only look for BGS instances based on the ones we have already found. Thus, some of our current results may be biased by the

[33]We have used our text classifier on these pages and they were identified as scam.

[34]More precisely, the BGS instances that can be found through our search queries are almost exclusively English.

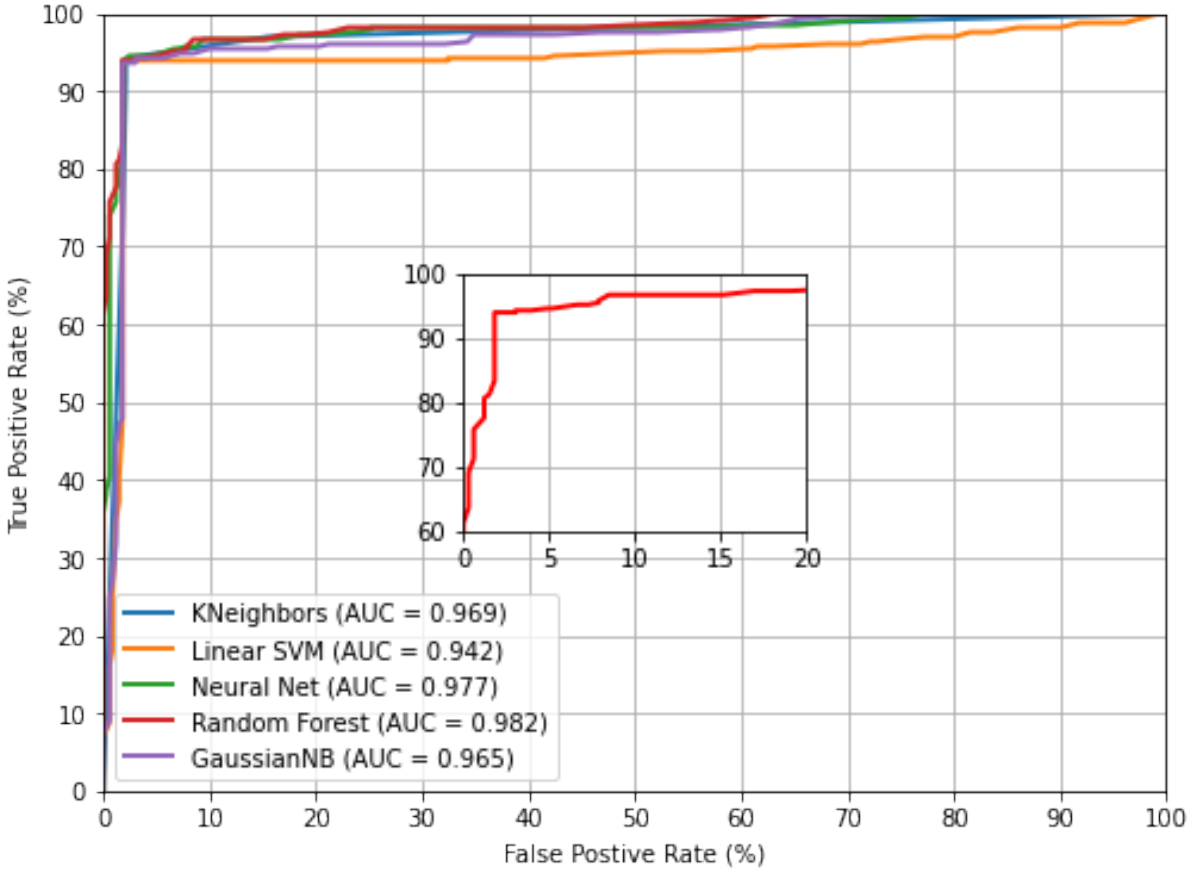

Figure 4.14: Features classifiers area under the curve (AUC).

type of BGS instances we are looking for, and a more systematic search would shed new light on the situation. For example, by improving our search queries, new and different BGS instances might come to light.

Another limitation is that we have trained our classifier on pages with English text. Thus, our crawler and our results only deal with English instances of BGS. That certainly doesn't mean that the scam is not active in other languages, and we would have overlooked these instances in that case.

Finally, we depend on text classification to detect BGS instances. However, this type of classification can be evaded relatively easily. We could enhance our feature set to be less dependant on the text that is being presented to the user. In our future work, we will build a more accurate text classification model and we will add some non-text-based features. Additionally, we would like to further explore the relations between the scam addressees. For this purpose, clustering techniques such as multi-input heuristic clustering can be used [21, 147].

4.9 Conclusion

In this chapter, we investigated what we call the "Bitcoin Generator Scam". In BGS, the scammer promises to generate free bitcoin using dubious methods such as owning a high-speed mining device or the ability to hack the blockchain. The attack is being advertised through webpages and targets the victims who are looking for an easy profit using cryptocurrency. We have used our model to automatically search the internet for scam pages, monitors their behavior, and collects the cryptocurrencies addresses used by the scammer.

Identifying a scam addresses by analyzing the blockchain history is typically difficult, error-prone, and only works on addresses with a good transaction history. However, our system proactively looks for the source of the scam, which enabled us to detect transaction-less addresses or addresses with a low number of transactions. Finally, we also innovate with the source of information we use; in addition to using traditional search engines, we showed that services such as the Internet Archive, `urlscan.io`, and `CuteStat.com` can be used to increase the number of addressees found significantly.

Our data collection spanned 21 months; in that time, we uncovered 9,140 cryptocurrency addresses extracted from 1,216 unique domains. These addresses have received $9,610,109 USD, with an average of $49.9 USD per transaction. We also used several features that we extracted from the scam websites and the addresses transactions history

to link scam instances and create groups of scams controlled by the same scammer. Our system has been integrated as an additional "feed" to the Anti-Phishing Working Group Cryptocurrency eCrime Exchange database.

Finally, we believe that our main contribution is targeting the scam source directly. By actively looking for the source of the scam instances, we discovered 9,140 addresses directly advertised by the scam. This is a much greater number of addresses than usually found in state-of-the-art research, where typically the scam instances are manually collected, and the bulk of the addresses come from clustering techniques such as the multi-input heuristic algorithm [127].

All the data used in our study is freely available at `http://ssrg.site.uottawa.ca/bgsextended/`.

Chapter 5

Generalizing our Model and Increasing Automation

5.1 Introduction

In this thesis, we started our work by proposing a data-driven model to detect and track the game hack scam (GHS) in chapter 3. We manually crawled the web and collected the GHS training dataset to train our classifier and run our model. We then used these instances as a seed to search for and detect more scam instances.

During our investigation of the GHS, the process of generating the search query, crawling the web, and detecting new scam instances were completely automated. This encouraged us to expand our investigation and target other web-based scams with a web presence. Thus, in Chapter 4, we have investigated the bitcoin generator scam (BGS) scam. In this analysis, we followed the same approach we used to study GHS. We manually searched the web and collected a training dataset to train a classifier and run our model.

Using our model, we have investigated and analyzed both GHS and BGS. In GHS, we have detected more than 5.9k GHS domains, and showed that the attackers routinely target a vast array of games. In BGS, we have detected more than 1,200 scam domains and more than 9k bitcoin addresses associated with them. We also found that a small group of scammers controls the majority of the received funds. Additionally, we developed an early BGS detection system where we discovered more than 70% of the online addresses before receiving funds. However, a big challenge that significantly hindered our automation was the manual collection of the training dataset which was very time-consuming.

In this chapter, we leverage the expertise and insights gained, to generalize our model, increase automation, and significantly reduce the manual efforts required to setup such a

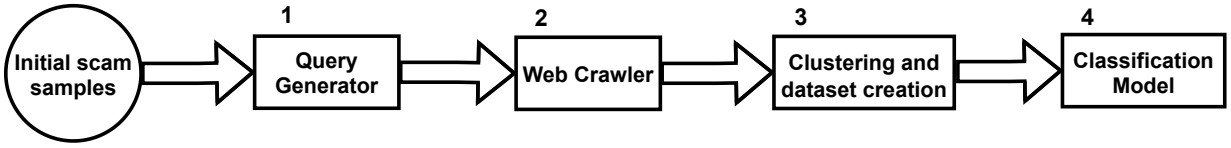

Figure 5.1: A Generic Model for Scam Detection and Analysis.

study. Given a small set of scam pages as initial input, our model generates scam-related search queries and uses them as a seed to search for more scam pages. We cluster the collected pages and automatically select the clusters with a high probability of containing scam pages. We then use our model to create a benign training dataset. In order to validate this generalized model, we have successfully generated training datasets for BGS and GHS. This required less effort and time than our initial studies. We used the datasets to train new classification models and used them to detect new scam instances. Our classifiers detected the BGS and GHS scam instances with good accuracy.

Figure 5.1 describes our system, which includes four modules:

1. **Search query generator.** This module generates keywords that are likely to be used in the scam pages.

2. **Web crawler.** This module uses the previous queries to search for scam pages using search engines, such as *Google.com*, and customized historical search engines such as `cutestat.com`. We crawl the resulting pages and recursively crawl once all the links that they contain. We collect the pages HTML content, URI redirections, as well as screen-shots.

3. **Clustering and dataset creation.** This module uses the previously collected web pages to create our labeled training dataset.

4. **Classification model.** This module categorizes the crawled pages as either "scam" or "clean" pages based on their text.

5.2 Query Generator

Finding good search queries that are highly likely to lead to scam pages is an important task. In this section, we discuss three techniques that we used to generate our search queries:

1. We **utilized the context-specific corpus** to generate search phrases that are highly likely leading to the scam. To successfully lure the victim to the scam website, the attackers create a website with content, words, and language related to the original service or product. For example, Miramirkhani *et al.* [103] have shown that technical support scammers use specific words, such as "call", "technicians" and "virus", in the content of a scam page to convince the users that their devices are infected with a virus.

   Our analysis of the GHS and BGS in Chapters 3 and 4 had shown the same result. For example, the words "hack", "tool", and "online" were widely used in the GHS, while the words "bitcoin", "btc", and "mining" were widely used in the BGS. Generating relevant search queries from a context-specific corpus has been used effectively in the past for TSS detection and analysis [134]. We have also followed a similar approach to create GHS search queries in Section 3.2.2.

2. **The "Keywords" meta tag:** it represents a set of a comma-separated list of keywords that are relevant to the web page, and used to inform the search engines about its content [8, 13]. The contents of the "Keywords" meta tag can be extracted from the scam dataset to generate more representative search queries. We have used this approach to create BGS search queries in Section 4.2.2.

3. **Google trends service:** Google trend reflects the popularity of search queries as normal web users use it. We can use the prepared queries created in step 1 and step 2 to crawl Google trend API[1] and generate more queries related to the scam. For example, we have used Google trend service to generate 1,604 unique queries related to GHS as described in Section 3.2.2.

5.3 Web Crawler

The primary purpose of this module is to browse the web and collect web pages to use in our clustering. Here, we use the previously identified search queries as a seed to search for and collect scam pages. We can use two sources to search for and collect our pages, which are:

- Using **search engines** such as *Google.com*, *Bing.com*, and *search.yahoo.com*. For each query, the crawler can visit a predefined number of pages returned by each engine. For example, we considered the first two pages (that is, 20 search results).

[1] https://trends.google.com/trends/?geo=US

- Using the **customized historical search engines** to collect domains and URLs with content related to the performed search. For example, `urlscan.io`[2] custom search reports previously scanned domains and URLs with DOM structure similar to the performed research. Other websites that can be used are `website.informer.com`[3] and `cutestat.com`[4], which are websites that gather detailed information on other websites. These websites have custom search services that report domains and URLs with text content similar to the performed search. This search aims to collect many scam instances with similar content when compared to the initial scam samples.

  In case the domain is no longer available, we use the **Internet Archive**[5] and `urlscan.io` to collect up to 5 previous snapshots of the domain. The **Internet Archive** is a digital library that provides a collection of readily available digitized materials, including music, public-domain books, internet sites, and games for free.

After collecting the URLs, we filter out the URLs hosted on the Alexa top 1K domains[6]. Our analysis of the BGS and GHS has shown that the URLs hosted on these domains do not contain scam instances.

The crawler can be built and customized using different technologies and libraries. In our work, we build our crawler based on ChromeDriver[7] and Python Selenium[8]. We then use Python beautifulsoup[9] and the CSS selectors to collect and crawl the URLs we gathered from the search results. We use a lightweight scripted headless browser built using python by integrating Selenium, ChromeDriver, and BeautifulSoup to implement our crawler.

5.4 Clustering and Dataset Creation

The goal of the clustering step is to group in the same cluster the scam pages based on some common features. We then semi-automatically label the true positive clusters and select the pages within as our scam dataset. For this purpose, we propose applying a two-step clustering process. An initial clustering on a small dataset of web pages, where we manually inspect a small set of clusters that have a high likelihood to contain mostly scam pages and label the true positive clusters. We then use the true positive clusters to

[2] https://urlscan.io/
[3] https://website.informer.com/
[4] https://www.cutestat.com/
[5] https://web.archive.org/
[6] `https://www.alexa.com/`
[7] `http://chromedriver.chromium.org/`
[8] `https://selenium-python.readthedocs.io`
[9] `https://pypi.org/project/beautifulsoup4/`

automatically label a bigger dataset. To select the benign samples, we randomly select pages from the clusters that we do not label as true positive clusters.

Our analysis of the BGS and GHS have shown that when scammers target a specific online service, the scam instances share common criteria that we can use to group them in the same cluster, such as:

- The presence of specific keywords: attackers create a website with content, words, and language related to the original service or product to successfully lure the victim to the scam website. As a result, the scam instances have content highly similar to the targeted service [103, 134]. For example, Miramirkhani *et al.* [103] have shown that technical support scammers (TSS) use the words "call", "technicians" and "virus", in the content of a scam page to convince the users that their devices are infected with a virus. The same case applies to BGS and GHS; for example, the words "hack", "tool", and "online" were widely used in GHS -Section 3.4.1-, and the words "bitcoin", "btc", and "mining" were widely used in BGS -Section 4.4.1-.

- Attackers use pre-built templates to create their attacks. Our analysis of GHS in Section 3.4.2 has shown the existence of online tutorials on how to copy, customize, and deploy existing templates to target online games. These templates are so easily usable that a nontechnical scammer can customize and publish them without effort. Furthermore, the majority of the scam instances we have found have similar templates. We have also found that many BGS instances use similar templates.

- The presence of the scam payload: in some cases, the attackers provide unique identifiers that link the scammer to the scam instance. These identifiers are used for different purposes, such as collecting the victim's information, receiving payments, or communicating phone numbers or emails. For example, in the TSS [134], the scam instance contains a phone number that can be used to contact the scammer. In BGS, the scam instance contains a cryptocurrency address to receive the victim's payment.

5.5 Classification Model

Since we aim to build an automated system and deploy it to protect the internet users, we need to continuously monitor the web and detect scam pages as soon as possible. Thus, we use the training dataset we created in the previous section to train a classifier that automatically distinguishes scam instances from genuine URLs at the crawling time.

	BGS	GHS
#tokens (word)	77	107
# queries from the context-specific corpus	148	127
#queries from meta tag	214	70
#queries from Google trend	520	391
#total queries	882	588
#unique queries	779	582

Table 5.1: Summary of the Generated Queries for BGS and GHS

To identify the scam instances from the set of crawled pages, we propose using a text-based classification model to classify the crawled pages based on the text as seen by the end-user. More precisely, we propose using the TF-IDF of the words displayed to the users to extract the training features. Our results in Chapters 3 and 4 have shown that the linear SVC text classifier has high accuracy in detecting the GHS and BGS instances. Using a text classification enables us to detect scam instances with scam words in common regardless of the template they are using.

5.6 Creating Training Datasets Automatically for BGS and GHS

In this section, we use our approach to create training datasets for BGS and GHS. We then train a text-based classifier on the training datasets and validate our results on real-world datasets that we crawled during our investigation of the scams. Finally, we compare the time we spent preparing the dataset manually and using the automated approach.

In our work, our goal is to ensure that our approach is generic and can be used to create a good training dataset starting from a few known scam pages. Thus, we have randomly selected 5 pages from each of the BGS and GHS scam datasets we identified during our research of the two scams. We then applied our approach on these pages to create the training datasets.

We used the techniques mentioned above to generate our search queries. Our results are presented in Table 5.1[10].

[10]The complete list is available at `http://ssrg.site.uottawa.ca/datasetWWW/`.

	BGS		GHS	
	Dataset A	**Dataset B**	**Dataset A**	**Dataset B**
#domains	7,175	6,267	5,145	5,483
#pages	16,162	21,664	8,208	13,245

Table 5.2: Summary of the datasets obtained through the crawler

5.6.1 Collecting the Corpus of Web Pages

We then used our web crawler to collect a corpus of web pages that will be used in our clustering process. We used the previously identified search queries as a seed to search for BGS and GHS pages using *Bing.com*, *Google.com*, and *search.yahoo.com* for a month. We searched daily and extracted the first 20 URLs returned by the search engines. We have also used `website.informer.com` to collect domains related to the search queries.

In this step, we divide the corpus of pages into two datasets, namely dataset A and dataset B. We will use dataset A to label a set of pages as scam in a semi-automatic way. We then use these labeled pages to infer the label of the entire dataset B.

We can create the two datasets in several ways. In our work, we used the pages we collected from the search engines as dataset A and the pages we collected from `website.informer.com` as dataset B.

We present the results of these steps in Table 5.2[11].

5.6.2 Clustering and Dataset Creation

In our work, we cluster the pages based on structural similarity. This helps to catch the different variations of the same attack when scammers mass-produce scam instances using the same template and changing the content. For clustering, we use the method proposed by Cui *et al.* [47], where "proportional distance" is used to measure the similarity between the pages tag-vectors. The proportional distance is a straightforward distance metric based on the HTML tags vector of web pages. We use the HTML elements provided by the World Wide Web Consortium[12] to create a vector that contains a predefined corpus of HTML tags. We then calculate the proportional distance between the generated vectors by dividing the count of indexes of the tag vectors that have a different value over the count of indexes of the tag vectors that are not equal to zero in at least one of the vectors.

[11]The complete domain names and URLs are available at `http://ssrg.site.uottawa.ca/datasetWWW/`.

[12]`https://www.w3.org/TR/html-markup/elements.html`

In our work, we consider all of the tags to create our tags vector except the common tags such as <html>, <head>, and <body>. We then assign a fixed ordering of the tags in the corpus, which we use to create a "vector" of the size of the corpus. We generate the corresponding vector for each page by counting how many times each corpus tag occurs in the page DOM. For example, consider Figure 5.2, where two simple pages DOMs are provided. If the corpus consists of the HTML tags <b> <form> <p> <h1> <button> <h2> <iframe> <input> and , in that order, then the tag vector for the page $p1$ is <0 ,1 ,2 ,3, 1, 1, 0, 2, 4>, and the tag vector for the page $p2$ is <0, 1 ,0 ,4, 0, 0, 0, 0, 6>.

```
<html>
<head>
</head>
<body>
    <div>
        <h1>data1</h1>
        <h1>data2</h1>
        <h1>data3</h1>
        <h2>data4</h2>
    </div>
    <div>
        <div>data5</div>
        <div>
            <p>data6</p>
            <p>data7</p>
            <form>
                <input>
                <input>
                <button>
            </form>
        </div>

    </div>

</body>

</html>
```

(a) DOM of the page $p1$

```
<html>
<head>
</head>
<body>
    <div>
        <div>data1</div>
        <div>
            <p>data2</p>
            <p>data3</p>
            <form>
                <h1>data4</h1>
            </form>
        </div>
    </div>
    <div>
        <div>data1</div>
        <div>
            <h1>data4</h1>
            <h1>data5</h1>
            <h1>data6</h1>
        </div>
    </div>
</body>

</html>
```

(b) DOM of the page $p2$

Figure 5.2: Tag vectors

We first computed the clustering threshold that yields clusters that are both dense and far away from each other. The optimal thresholds of BGS and GHS are presented in Figures 5.3 and 5.4, respectively. The x-axis represents the proportional distance between the vectors belonging to the cluster. The y-axis represents the average proportional distance of

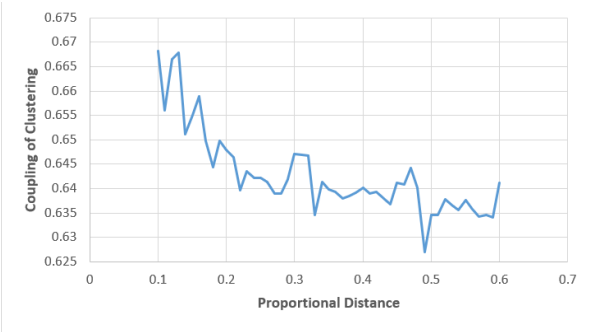

Figure 5.3: Optimal threshold of BGS clusters

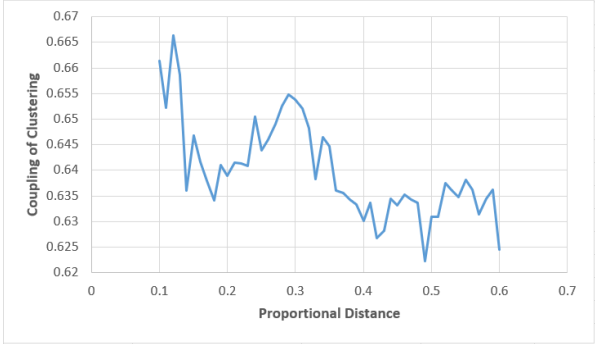

Figure 5.4: Optimal threshold of GHS clusters

vectors inside a cluster divided by the smallest proportional distance between two vectors in the cluster. The smaller the y value, the more related the pages within the clusters. The optimal identified threshold value for the BGS and GHS was found to be 0.49 in both cases, which we used in our analysis.

After identifying the optimal threshold, we have applied our initial clustering on dataset A. Overall, the algorithm generated 986 and 699 clusters with more than one scam page for BGS and GHS, respectively. We then need to find a set of clusters that; 1) are small enough to be manually inspected, and 2) have high likelihood to contain scam instances.

In our case, we decided that we would not inspect more than 70 clusters for each scam[13]. We first order the clusters based on how similar their pages are to the initial set of scam pages. To do that, we extract the words that are frequently used in these pages[14], and then count the number of these words appearing in every page of each cluster. We order the clusters based on the number of occurrences of the most frequent word. We then keep increasing the minimum number of occurrences until we select 70 clusters or less.

Figure 5.5 shows the number of clusters that contain a given number of these words. As can be seen, to select fewer than 70 clusters, we need 14 words for GHS and 15 for BGS. We then randomly select 2 to 3 pages from each cluster for visual analysis in order to label the clusters. If the randomly selected pages are a scam, then we label the complete cluster as a scam. Table 5.3 presents our results. Overall, we found 19 BGS clusters containing 232 pages and 41 GHS clusters containing 352 pages.

After we manually labeled the scam pages in the initial clustering round, we used these pages to automatically label the pages we had in dataset B. We merged the labeled scam pages of dataset A with dataset B and applied our clustering algorithm on the combined

[13]This threshold of 70 is arbitrary and depends on the time the researcher is willing to spend.

[14]In our case, we extract the words with at least 5 occurrences.

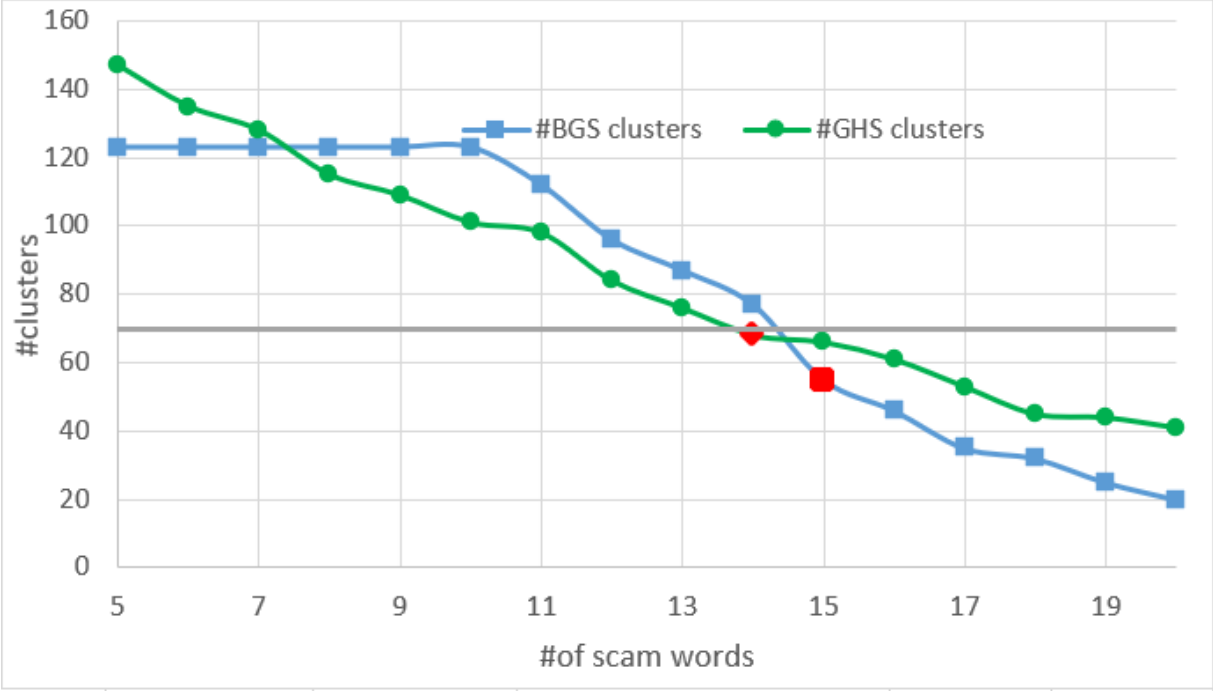

Figure 5.5: Number of clusters vs number of scam words in the web pages.

	Threshold	**Scam**		**Clean**	
		#clusters	**#pages**	**#clusters**	**#pages**
BGS	15	19	232	36	165
GHS	14	41	352	26	176

Table 5.3: Clustering results

dataset. We then labeled the clusters that contain the scam pages of dataset A scam clusters. Overall, we have labeled 151 new BGS scam pages and 6 new GHS scam pages.

To create a benign dataset, we used the clusters that were not labeled as scam, excluding any cluster containing any URL hosted on the same domains as a scam page. We randomly selected one page from each such cluster.

Our manual interaction is limited to labeling 70 clusters or less that we select in the initial clustering step in our model. We manually inspect and label the true positive cluster to reduce the number of mislabeled pages in our training dataset. We can reduce or eliminate this manual process by tightening our filtering process in our initial clustering step. However, in this case, we will have a trade-off between automation and model accuracy. We may increase automation, but we will reduce the true positive rate and increase the false positive rate.

5.6.3 Validating the Training Datasets

In this section, we use the datasets we generated in Section 5.6.2 to train text classifiers and validate our results on the datasets obtained during our investigation of the BGS and GHS.

Classification Process

To evaluate our datasets, we have used the same text-based classification model that we used in Sections 3.2.4 and 4.2.4. We tested five different classifiers from the Scikit-learn python library [120] on our training set: Linear SVC, NB, KNN, RF, and MLP[15]. We have extracted our features from the text as seen by the end-user. More precisely, we have used the TF-IDF of the words displayed to the users.

Our classification model achieved a good accuracy, with an F1-score of more than 98% on both datasets. We show the results in Tables 5.4 and 5.5. As can be seen, SVC achieved the highest F1 scores on both datasets. The other classifiers also performed fairly well, with Kneighbors and NB having the lowest F1 score. Based on these results, we used the SVC classifier to validate our methodology on pages that were not used in the training phase.

[15]We have used the default parameters as described in Section 3.2.4

Classifier	Page type	Classified clean	Classified scam	Precision	Recall	F1 Score
SVC	clean	351	3	99.14	98.3	98.71
	scam	6	348			
MLP	clean	350	4	98.86	98.02	98.43
	scam	7	347			
RF	clean	351	3	99.14	98.3	98.71
	scam	6	348			
NB	clean	349	5	98.33	83.61	90.37
	scam	58	296			
Kneighbors	clean	349	5	98.27	80.5	88.5
	scam	69	285			

Table 5.4: Results of 10-Fold cross-validation of the five classifiers on BGS dataset

Classifier	Page type	Classified clean	Classified scam	Precision	Recall	F1 Score
SVC	clean	378	2	99.47	99.47	99.47
	scam	2	380			
MLP	clean	378	2	99.47	99.21	99.33
	scam	3	379			
RF	clean	377	3	99.21	98.95	99.07
	scam	4	378			
NB	clean	358	22	94.43	97.64	96
	scam	9	373			
Kneighbors	clean	380	0	100	93.71	96.75
	scam	24	358			

Table 5.5: Results of 10-Fold cross-validation of the five classifiers on GHS dataset

	BGS	BGS original work	GHS	GHS original work
True negative	99.86%	99.88%	99.90%	99.77%
True positive	87.26%	93.91%	88.03%	97.49%
False negative	12.74%	6.09%	11.97%	2.51%
False positive	0.14%	0.12%	0.10%	0.23%

Table 5.6: Classifier results on pages that have not been used in the training phase

Classifier Validation on Testing Dataset

In this section, we validate our classifiers using real-world testing datasets that were not used in the training phase. To create the testing datasets, we used the corpus of pages we collected during our analysis of the BGS and GHS. For the GHS dataset, we randomly selected 100k pages out of 679k pages we collected during our crawling. For the BGS dataset, we have used the whole BGS dataset, which contains 77,214 pages.

We present our classification results in Table 5.6. As shown in the table, our methodology achieved good results for both types of scams. Our classifiers successfully detected more than 87% of the scam pages while maintaining a false positive rate as low as 0.23%[16].

A Comparison between the Manual and Automated Approach

To prepare the GHS training dataset, we have manually searched the web for around a week to collect our initial scam samples. On average, we have spent 4 hours a day searching for and understanding the scam, adding up to a total of 28 hours of work. We then used the collected samples to create search queries to run our crawler. After a month of crawling, we manually inspected around 600 pages to create the benign dataset. On average, we have spent 10 seconds visually inspecting each page, adding up to 1.5 hours. Overall, we have spent around 31 hours on manual search and inspection.

We have followed a similar approach to prepare our BGS training dataset. On average, we have spent 4 hours per day searching and collecting scam samples, adding up to a total of 28 hours. We have also manually inspected around 1,500 snapshots that belong to 307 possible scam domains[17] that we collected from the Internet archive. Finally, we manually

[16]We manually inspected 1,000 randomly selected pages classified as benign and approximated the total number of false negatives based on the dataset size.

[17]We collected the domain by crawling `cutestat.com` search engine and a blacklist maintained by `Bitcoin.fr`.

inspected 400 randomly selected pages collected during the first week of crawling to create our benign dataset. On average, we have spent 10 seconds visually inspecting each page, adding up to 5 hours. Overall, we have spent around 33 hours on manual search and inspection.

In our automated approach, our manual efforts are kept as low as possible. In our initial clustering, we automatically select a maximum of 70 clusters and only inspect 2 to 3 pages from each cluster for visual analysis. On average, we have spent 10 seconds visually inspecting each page, adding up to half an hour on manual inspection.

Finally, although the automated approach results are not as good as the results from our manual work in Chapters 3 and 4, our main gain is significantly reducing the manual effort while maintaining good accuracy. Ultimately, we saved more than 98% of the manual effort we spent to create the datasets manually[18].

5.7 Discussion

5.7.1 Model Persistence

Search engines have become a fundamental part of our daily lives and one of the most powerful tools on the internet, given that 68% of all website traffic comes from search engines [19]. Our work and model utilizes the popularity and widespread use of search engines to search for and detect scam instances. In our search, we depend on common features that are shared between the scam instances to identify and detect scam instances. However, since we work in an antagonistic environment, scammers may change their tactics to evade our model.

Moreover, technology may change with time, and more tools can be used by attackers to create more sophisticated attacks. For example, attackers may use AI to customize their attack instances and obfuscate the traces of the attack. Attackers can also evade search engines and target a specific audience using social engineering attacks. Note, however, that scammers do not have unlimited freedom in the techniques that they can use. Evading our model will make it harder for scammers to spread their scams instances widely, thus reducing the overall profit that they will gain. Scammers will also need to spend more effort and time creating customized attacks and targeting their victims. Therefore, while

[18]In our analysis, for both the manual and automated approach, we did not include any automated process, such as crawling time. We only included the time we manually spent searching, inspecting, and labeling the pages

[19]https://websitesetup.org/news/internet-facts-stats/, accessed in 2021

an attacker could evade our model, we will still be successful in making the scammer's life harder and more costly, and simultaneously reducing the number of victims.

Furthermore, in Section 3.6, our analysis showed that template providers facilitate the GHS scam instances. Since the providers facilitate the creation and hosting of the scam instances, this means that the scammers can focus their time on publishing and advertising the instances. Thus, it will not be a time consuming task for attackers to completely change the current attack scheme in the hope of evading our detection model. Although the attack vector and the features for detection may slightly change, we believe that the main concept of the attack will still be valid in which the attackers need a communication channel to reach out to the victims. Furthermore, rather than creating an entirely new scam instance every time, attackers will be reusing/modifying their attacks and redeploying them. Thus, our general idea of proactively searching for scam instances and detecting them using a classification model will still be valid. Indeed, over time, we will need to adapt it and tune it to deal with the modified and upgraded attacks.

Our analysis in Section 2.7 showed that, in the majority of the fraud and scam activities such as GHS and BGS, attackers use the web as their communication medium. However, in the future, the attackers may divert from using web pages as their communication channel into using other means such as directed social engineering attacks, emails, or even a completely new technology that does not exist today. In this case, we will have to continue monitoring the evolution of the attacks and adapt our methodology accordingly. Ultimately, we are working on making the attackers' lives more complex and less profitable, thus deterring them from choosing the career of an attacker.

5.7.2 User Education

Given our findings in Chapters 3 and 4 we argue that GHS and BGS are a real and dangerous threat to web users. For example, in the case of BGS, the scam does not need any monetization efforts since scammers only need to create and publish the scam instance and wait for the victims to access these pages. If the scam is successful, the victims will willingly transfer money to scammers. In some other types of attacks, such as ransom and TSS, attackers need to contact the victims throughout the attack.

Even though automated systems like our model are essential to automatically discover these scams as soon as they arise, we anticipate that the threat of GHS and BGS can be restrained with the education of the public. User education has been a long-lasting obstacle of security systems, and attackers have often abused the lack of user education to launch attacks through social engineering. We argue that, to some extent, it is easy to

explain the concepts of GHS and BGS to users. This is because, in both scams, there are no complex concepts that the user must remember. A users' long-lived dream of accessing easy profits or of accessing shortcuts to get free services is not possible as advertised by these scammers. Thus, educating the public not to trust these pages is one of the first mitigation steps against these scams. The idea of educating the public has been suggested as efficient mitigation for different types of scams, such as TSS [103], romance [163], and telephone [106, 151] scams.

Multiple countries are already working on raising awareness about the safety and health issues through public service announcements [103]. This type of awareness through public announcements would be an ideal approach for educating users about the dangers and characteristic signs of GHS and BGS. Even though we have raised awareness and warnings about BGS through public media and through the uOttawa website[20], the announcement is far from reaching a wider general audience, as the announcements are only available on specific websites. Additionally, even though we can educate non-technical people to recognize GHS and BGS, we must provide a simple way of informing the web users when they encounter these web pages.

5.8 Limitation and Future Work

One of the main limitations of our study is that we validated our methodology using two types of web-based scams only. Furthermore, we did not run complete experiments to collect our testing corpus of pages. We used the corpus of pages we collected in our previous work as our testing dataset. In our future work, we will further validate our methodology by studying other web-based scams where we depend entirely on the automated process to collect the scam dataset.

Another limitation is that our result is biased by the five initial pages we used as a starting point in our experiments. It is not guaranteed to achieve the same results if we start with different scam samples. However, we believe that the overall results will not significantly vary when changing the initial scam samples because we followed a completely random selection process to pick our initial 5 pages. We will further validate our methodology by repeating our experiments using different initial pages in our future work.

Finally, in our experiments, we have used the DOM structural similarity between the scam instances in our clustering step. However, our approach will not work if the scam instances do not have high structural similarities. In our future work, we will consider other

[20] `https://media.uottawa.ca/news/catch-cryptocurrency-thief`

types of clustering, such as content-based clustering. As described above, when the scam targets a specific service, the scam pages host highly similar content, words, and languages. This similarity can be used to group the scam instances in the same cluster. In such cases, the HDBSCAN [34] which is a hierarchical density-based spatial clustering of application with noise, can be used to cluster the web pages into related groups. Alternatively, we can use the proportional distance to measure the similarity between the page's word vectors. We can generate the initial vector from the bag of words (BoW) of the most used words in the initial scam sample.

5.9 Conclusion

In this chapter, we generalized our data-driven model and leveraged the expertise and insights gained from studying the BGS and GHS to increase automation and drastically reduce the manual efforts required to setup such a study. Given a small set of scam samples, our model formulates scam-related search queries and uses them on different sources, such as search engines and customized historical search engines to search for and collect potential scam pages. After collecting a sufficient corpus of web pages, our model semi-automatically clusters the search results and creates a labeled training dataset with minimal human interaction. Our system provides a continuous tracking and detecting mechanism in which we proactively look for scam web pages and monitor their evolution over time.

We have used our model to create training datasets for BGS and GHS by utilizing initial scam samples of only five web pages. Our analysis showed that we could create a good labeled training dataset that can detect the scam pages with good accuracy while significantly reducing the manual effort. The classifiers successfully detected more than 87% of the scam pages while wrongly classifying less than 0.24% of the benign pages.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The problem of scam attacks is a continued threat to society. Despite both academia and industry's efforts to prevent scam attacks, scams continue to cause an online threat. Furthermore, with the people's lifestyles being shifted toward being online, several scam attacks are emerging and causing many losses. For example, the "game hack scam" (GHS) has been visited millions of times, and the "bitcoin generator scam" (BGS) has received millions of USD without being investigated and analyzed adequately.

In this thesis, we propose a data-driven approach that proactively searches the web and looks for scam web pages to detect, investigate, and prevent them. This, in turn, reduces the number of victims by providing an early detection and prevention system. Given a small set of initial scam pages, our system depends on formulating scam-related search queries and use them on multiple search engines to collect data about the websites to which victims are directed when they search online for pages with content related to the scam. We then cluster the collected corpus of pages to create a labeled training dataset with 98% less manual effort. We have used our model to report on the first systematic investigation of both GHS and BGS.

Our research of the GHS showed that GHS attackers use popular websites to publish links leading to this type of scam. A variety of sites are used to disseminate these links: social media, streaming sites, blogs, and even unrelated sites such as *change.org*, *jeuxvideo.com*, or *researchgate.net*. Our data collection spanned a year, during which we uncovered 65,905 different GHS URLs, mapped onto over 5,900 unique domains. We were able to link attacks to attackers and found that they routinely target a vast array of games. Analyzing our data showed that the attackers use pre-built templates to create their at-

tacks. We also found that they tend to target different games. Furthermore, we found that GHS instances are on the rise, and so is the number of victims. Despite its low profile, the click traffic generated by the scam is in the hundreds of millions. Our low-end estimation is that these attacks have been clicked at least 150 million times in the last five years. Additionally, in keeping with similar large-scale scam studies, we found that the current domain-based public blacklists are inadequate and suggest that our method is more effective at detecting these attacks. Finally, we found that more than 90% of the GHS related executable files are flagged by at least five antivirus scanners in virus total.

Our analysis of the BGS showed that it is a simple scam that targets victims looking for a quick and easy way to make money through cryptocurrencies. The attackers trick the victims by promising to generate new bitcoins using the ones deposited by them. During 21 months of crawling, we collected 9,140 Bitcoin addresses mapped to more than 1,200 unique scam domains. On average, these addresses have received $49.9 USD per transaction, accumulated to $9,610,109 USD. Once a scam instance is identified, we monitor it to trace payments and bitcoin addresses that are being used over time. Our detection system has two significant contributions; First, unlike most bitcoin-based scam monitoring systems, we do not rely on analyzing transactions on the blockchain to find scam instances. Instead, we proactively find these instances through the web pages advertising the scam. Thus our system was able to find addresses with very few transactions, or even none at all. Indeed, over half of the addresses that have eventually received funds were detected before receiving any transactions. Second, we collected a large number of bitcoin scam addresses through automatic crawling compared to the state-of-the-art research, where typically scam addresses are manually collected, and the bulk of the addresses come from "multiplier" techniques such as the multi-input heuristic clustering algorithm [127].

6.2 Future Work

6.2.1 Study of the GHS Templates Providers

Analyzing our GHS data showed that the attackers use pre-built templates to create their attacks. We found that these templates are powered by online advertisement websites that either provide GHS instance templates or provide tutorials on how to copy existing templates and deploy them in the scam. We found that these websites embed unique signatures in the DOM of the GHS instances to publish the scam offers, which represents the final payload of the scam. Our initial analysis in Section 3.4.2 showed that two online advertisements websites control around 22k GHS instances (around one-third of all the

detected instances).

In future work, we aim to study the online advertisement websites templates providers in more detail and try to understand how they operate. We will also highlight the templates and techniques they use to create and publish scam instances. Finally, we will investigate the possibility of these websites publishing other types of scams using their advertisement system.

6.2.2 Validate our Approach Using other Types of Web-based Scams

In our work, we have used our data driven model to investigate and give insights into two types of web-based scams; the GHS -Chapter 3- and the BGS -Chapter 4-. However, there are other types of web-based scams. For example, HYIP attacks are widely advertised on online fora and blogs, such as bitcointalk.org and Reddit [21, 28, 147–149, 154]. In future work, we aim to incorporate new scam types into our system and detect these scams in their early stages.

6.3 Improving the Automation Aspect of our System

In our work, we aim to automate scam detection, tracking, and analysis. However, the final payload differs from one scam type to another, forcing us to manually study the BGS and GHS scams and determine the appropriate analysis that we can apply to each of them. For example, in Section 3.4 we have used unique identifiers found in the scam pages to detect similarities and infer common ownership of the GHS instances. In Section 4.4, we have analyzed possible evasion techniques used by the attackers to make it harder to track and detect BGS instances using automation detection systems.

In our future work, we will work on automating the scam analysis step. For example, we can automatically study the scam lifetime and the effectiveness of the current public blacklists against the scam by analyzing the lifespan of the scam domains. Another automated analysis is identifying the most frequent words used in the scam pages. The most frequent words analysis can be done in the form of a word cloud, where the size of each word correlates with the number of times it appears in the collected corpus of scam pages.

Appendix A

Cryptocurrencies Emerging Threats and Defensive Mechanisms: A Systematic Literature Review (Research Method)

In our review, we adopted the standard systematic literature review (SLR) guidelines of Kitchenham and Charters [89], which is "a means of evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest". The review strategy consists of six steps: 1) research questions, 2) search strategy, 3) study exclusion & inclusion criteria, 4) quality assessment criteria, 5) document retrieval and data extraction, and 6) data synthesis.

Figure A.1 describes the steps of the research method and review protocol.

A.1 Research Questions

In the SLR, our aim was to explore the threats that emerged with cryptocurrencies and identify the proposed defensive mechanisms that were developed to prevent these new threats. Moreover, we aimed to provide easy access to the publicly available datasets in the literature. In particular, we addressed the following research questions:

RQ1: With the introduction of cryptocurrencies, what are the types and scales of cybercriminal activities reported by researchers?

RQ2: What are the proposed defensive mechanisms available to detect cybercriminal activities, and what is the reported effectiveness of these mechanisms?

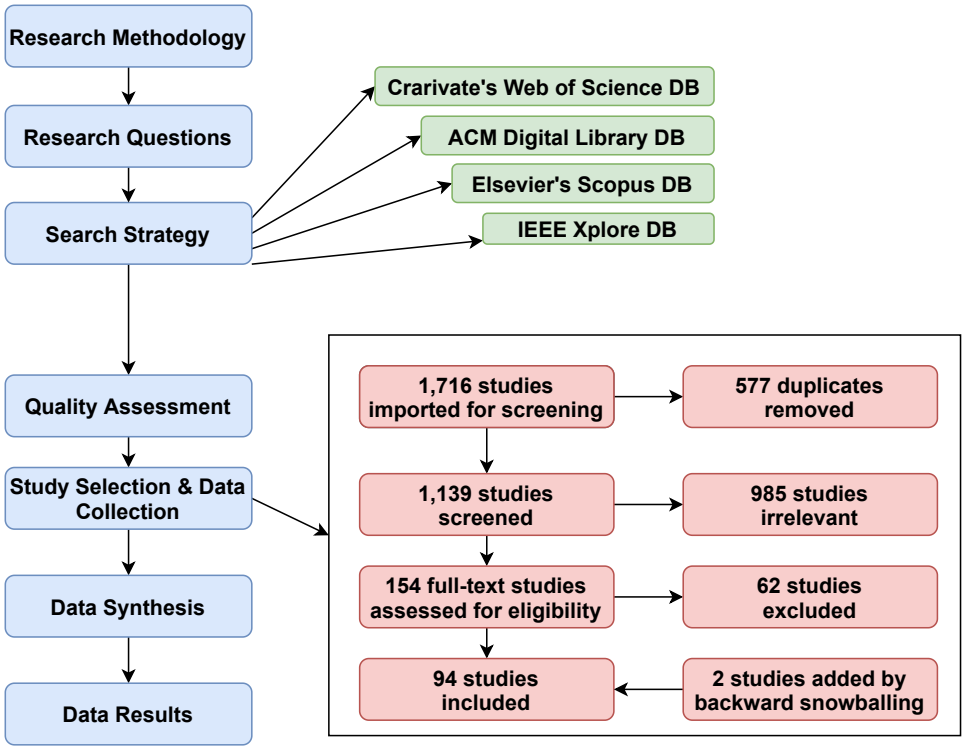

Figure A.1: Review methodology.

RQ3: For cryptocurrency cybercrimes detection and prevention, what are the public datasets provided in the literature, and how have these datasets been collected?

A.2 Search Strategy

Our search strategy was developed by identifying the two main concepts related to our research questions. The first is the concept of cryptocurrency and its related terms and synonyms. Our second concept refers to the cybercriminal activities that use cryptocurrencies and their synonyms. To increase the effectiveness of our search query, we manually searched on Google Scholar for articles that discuss cybercriminal attacks that target cryptocurrencies and extracted the synonyms of cryptocurrency and cybercriminal activities as used by other researchers. We further included the names of the most used cryptocurrencies in 2019[1][2], which often represent the primary target for scammers.

Overall, we have collected ten different terms related to cryptocurrencies and sixteen terms related to cybercriminal activities. We then translated the different terms into

[1] https://www.statista.com/topics/4495/cryptocurrencies/
[2] https://leftronic.com/cryptocurrency-statistics/

Concept	Synonyms
Cryptocurrency	bitcoin, ledger, blockchain, cryptocurrenc*, “crypto-currenc*”, “coin mining”, Ethereum, litecoin, XRP, and tether
Cybercriminal activities	scam, hyip, “yield* investment program*”, ponzi, pyramid, fraud, abuse, “money laundering”, ransomware, phishing,“pump & dump”, pump-and-dump, *jacking, DoS, ”Denial of service”, and ”Denial-of-service”

Table A.1: Search query related terms

Boolean logical queries that we executed on four different databases (see Section A.2.1) to create our initial dataset of papers. The complete list of terms and the search query are presented in Section A.2.2.

The search results on the four databases gave us the list of articles that we used to extract the different synonyms related to our two concepts. These results indicate that our query has a high possibility of returning other articles that contain any of the synonyms included in our query.

A.2.1 Source Databases

For the systematic review, we used four different scientific and engineering databases and libraries. These databases are the top four databases suggested by our university library for conducting research in Computer Science.

These databases are:

- Elsevier’s Scopus database (`scopus.com`).
- ACM Digital Library database (`dl.acm.org`).
- Crarivate’s Web of Science database (`apps.webofknowledge.com`).
- IEEE Xplore database (`ieeexplore-ieee-org`).

A.2.2 Abstract Search Query

The keywords used to construct the search query are listed Table A.1.

The finalized search query is the following:

```
( bitcoin OR ledger OR blockchain OR cryptocurrenc* OR
 ‘‘crypto-currenc*’’ OR ‘‘coin mining’’ OR Ethereum OR litecoin
 OR xrp, OR tether )
```

```
AND
( scam OR hyip OR ‘‘yield* investment program*’’ OR
 ponzi OR pyramid OR fraud OR abuse OR ‘‘money laundering’’
 OR ransomware OR phishing OR ‘‘pump \& dump’’ OR
 pump-and-dump OR *jacking OR DoS OR "Denial of service" OR "Denial-of-
    ↪ service")
```

Initially, we ran the query on the full text of the papers. However, that returned hundreds of irrelevant papers. In order to reduce the results to meaningful, manageable, and relevant results, the search was ultimately limited to the title, abstract, and keyword metadata.

A.3 Inclusion Criteria

Although our search query is comprehensive and includes popular synonyms related to our research question, other researchers may use other synonyms that we do not know. Accordingly, our query will not detect these papers. Moreover, in our selection process, we may reject some related articles if neither the title, abstract, or keywords contained terms related to our research question. Therefore, we peruse the reference sections of the selected papers in search for additional relevant papers our search might have missed, a technique called “backward snowballing” [75].

A.4 Exclusion Criteria

In our search, we excluded non-peer-reviewed journals and conferences. Some of the researchers publish early results of their articles on `https://arxiv.org/`; we only considered the final versions published in the journals or conferences for such cases in the SLR. We limited our database search to papers written in English. We did not consider an article if the title, abstract, or keywords did not contain the keywords related to our research questions. We limited our search to the papers published after 2009, as the first successful cryptocurrency coin was introduced in 2009 [109].

Finally, We did not include articles submitted to conferences in unrelated fields, such as medical or commerce conferences. Including these conferences adds a large number of mostly unrelated papers, in particular because some of our terms such as “scam” and “fraud” are used in different contexts.

A.5 Quality Assessment Criteria

Identifying quality assessment criteria (QAC) improve SLRs in different ways, such as providing a more detailed inclusion/exclusion criteria and advising recommendations for further research [89]. In our SLR, we considered all work that meets the following assessment criteria:

- The paper has a clear, reproducible methodology.
- The paper presents and discusses cybercriminal attacks that target cryptocurrencies.

A.6 Study Selection & Data Collection

As shown in Figure A.1, the previous search returned 806 unique results. This was reduced to 114 unique and relevant articles after a manual selection process based on reading the papers' titles and abstracts. It was then further reduced to 64 papers once the full text of the papers was read. Finally, two papers were added thanks to the backward snowballing technique, creating a total of 66 unique and relevant papers used in our SLR. Our papers screening and selection were carried out with the aid of Covidence[3], a web-based software platform that simplifies the production of systematic reviews. It provides an interface to import articles, screen title and abstract, screen the articles full text and export the study results in different formats.

Our data extraction approach was motivated by our research questions. The following pieces of information were manually extracted, assessed, and synthesized:

- (D1) The type of crime(s) being discussed.
- (D2) The analysis evaluation criteria.
- (D3) The cryptocurrency in the study.
- (D4) The dataset source.
- (D5) The dataset availability for public use.
- (D6) The detection algorithm.
- (D7) The detection accuracy and efficiency.

[3]https://www.covidence.org/home

- (D8) The crime effectiveness (based on the USD value and scale).
- (D9) The evaluation of the crime effectiveness.
- (D10) Cryptocurrencies address clustering algorithm (when used).
- (D11) The features used in the classification process (if any).

Our dataset is made publicly available on our team's website and can be reused by other researchers or reproduced if necessary[4]. Our raw data includes the 11 data records mentioned above, the SLR protocol, the list of articles, the features and the classifiers used in each article. Our full analysis is presented in section 2.2.

[4]http://ssrg.site.uottawa.ca/slr/

Appendix B

Cryptocurrencies Emerging Threats and Defensive Mechanisms: A Systematic Literature Review (Breakdown per Article)

In this Appendix, we provide a breakdown of the data used to conduct our analysis in Section 2.2, as reported in the literature. We provide the reported cybercrimes scale in Table B.2, the resources used to prepare the training datasets in Table B.3, and a breakdown of the classifiers used in the papers and the results achieved in Table B.1.

Table B.1: The reported detection results in the literature

Ref.	**Crime Type(D1)**	**Detection Algorithm (D6)**	**Achieved Results (D7)**
[168]	Service Detection	Bagging and XGBoost	80.76% accuracy (XGBoost) and 78.46% accuracy (Bagging)
[150]	Service Detection	Random forest	72% accuracy (owner-based scheme) and 70% accuracy (address-based scheme)
[117]	Service Detection	Random forest	23.67% TPR and 0.02% FPR
[149]	HYIP (Bitcoin)	Random forest	83% TPR and 4.4% FPR

(Continued on next page)

Ref.	Crime Type (D1)	Detection Algorithm (D6)	Achieved Results (D7)
[21]	HYIP (Bitcoin)	Random forest	96.8% TPR and 96.9% Recall
[147]	HYIP (Bitcoin)	Random forest	95% TPR and 4.9% FP
[40]	HYIP (Ethereum)	XGBoost	94% precision and 81% recall
[80]	HYIP (Ethereum)	Random forest	99% precision and 97% recall (full data), and 98% precision and 96% recal (0 day detection)
[41]	HYIP (Ethereum)	Random forest	95% precision and 69% recall
[138]	Cryptojacking	Random forest	99.7% TPR and FPR less than 0.25%
[58]	Cryptojacking	K-Nearest Neighbors	88% precision and 87% F1 score
[87]	Cryptojacking	Support vector machine	97.9% TPR and 1.1% FPR
[113]	Cryptojacking	Capsule Network	Detect 87% of the instance instantly and 99% of the instances within a window of 11 seconds.
[174]	Cryptojacking	Proximity-based	99.7% TPR and 46.1% FPR
[156]	P&D	XGBoost	99.5% AUC, 85.5% sensitivity and 99.7% specificity
[167]	P&D	Random forest	Predicts the likelihood of a currency being pumped with an area under curve of over 90%
[56]	Service Detection	XGBoost	Accuracy of 96.3%
[6]	ML	Graph Convolutional Networks (GCN)	Accuracy of 97.4%
[5]	ML	Ensemble learning	Accuracy of 98.13%

(Continued on next page)

Ref.	Crime Type (D1)	Detection Algorithm (D6)	Achieved Results (D7)
[71]	General	RF	Accuracy of 99.55%
[61]	Ransom	Bayesian belief network (BBN)	Accuracy of 97.5%
[93]	General	XGBoost	Accuracy of more than 96%
[155]	DDoS	word-based classifier	Accuracy of 75%
[19]	DDoS	Multilayer perceptron (MLP)	High accuracy with 12 layers and higher training epochs.
[4]	Ransom	Random Forest	An accuracy of more than 80%
[49]	CryptoJacking	LSTM, Attention-based LSTM, and CNN	An accuracy rate of 95% in the static analysis and and 99% in the dynamic analysis.
[60]	CryptoJacking	TLC and Two-Level Classification	A precision and recall close to 1, but decrease with more programs run simultaneously.
[59]	CryptoJacking	Random Forest and Support vector machine	Achieves a near-perfect classification with samples of length as low as five seconds.
[94]	P&D	Random Forest and Logistic Regression	91% F1 score in the first 25s of the scam
[170]	Phishing	Support vector machine	84.6% F1 score
[171]	Phishing	Improved Graph Classification	73% F1 score
[38]	Phishing	Dual-sampling Ensemble algorithm	83% F1 score

(Continued on next page)

Ref.	Crime Type (D1)	Detection Algorithm (D6)	Achieved Results (D7)
[37]	Phishing	Graph Convolutional Network (GCN)	57.25% AUC with cluster size of 30k nodes, increasing the size will decrease the results
[121]	HYIP (Ethereum)	Extremely randomized trees	98% precision, 93% recall, and 95% F1 score
[54]	HYIP (Ethereum)	Ordered boosting	98% F score on the real-world dataset
[99]	HYIP (Ethereum)	Convolutional neural network	98.2% precision and 93.8% recall
[55]	HYIP (Ethereum)	XGBoost	96.55% F1 score
[140]	CryptoJacking	Heruestic algorithm	93% accuracy
[85]	CryptoJacking	Decision tree	97.1% accurrecy
[35]	CryptoJacking	Random Forest	97.1% AUC
[112]	P&D	Different classifiers	Around 30% F1 score
[160]	Phishing	AdaBoost	92% AUC
[53]	HYIP (Ethereum)	Ordered boosting	96% F1 score
[72]	HYIP (Ethereum)	Support vector machine	99% accuracy

Table B.2: Reported cybercrimes scale estimation in the literature.

Ref.	Crime Type(D1)	Currency (D2)	Scale (D8)
Analyzing the transaction history of the collected addresses			
[153]	Services detection	Bitcoin	Scam addresses received 11 million USD from 13,000 distinct victims, and returned back 4 million USD.
[98]	Ransom	Bitcoin	Scam addresses received 1,128.40 Bitcoin (310,472.38 USD) in the period from September 2013 through January 2014.
[70]	Ransom	Bitcoin	Scam addresses received 16 million USD from 19,750 victims.
[21]	HYIP	Bitcoin	scam addresses received around 10 million USD.

(Continued on next page)

Ref.	Crime Type (D1)	Currency (D2)	Scale (D8)
[68]	Phishing	Bitcoin	Scam addresses received over 50 million USD in 3 years.
[26]	Ransom	Bitcoin	Scam addresses received 169 bitcoins.
[44]	Ransom	Bitcoin	Scam addresses received 7059.9 Bitcoin (2,834,468 USD).
[146]	Honeypot	Ethereum	690 honeypot smart contracts that accumulated profit of more than 90,000 USD from 240 victims.
[20]	HYIP	Ethereum	scam addresses received almost 0.5 million USD.
[155]	DDoS	Bitcoin	7.4% of Bitcoin-related services and 60% of large mining pools have been DDoSed
[57]	DDoS	Bitcoin	Reduce the daily number of big transactions
[3]	DDoS	Bitcoin	Reduce the average trading volume during the attack
[42]	HYIP	Ethereum	Found 835 Ponzi scheme contracts that have over 17 million US Dollars invested by victims
Inferred an Estimation based on the average P&D events and the currency price variation during the event			
[82]	P&D	Multiple	2,150 P&D schemes over 20 days of crawling witn an average of 1.6 P&D events per currency per day.
[39]	P&D	Bitcoin	Found 471,899 (0.04% the full dataset) abnormal price records in Mt. Gox leaked dataset. The abnormal price transactions involved 16,660 (13.09%) of the users.
[156]	P&D	Multiple	612 P&D schemes.
[167]	P&D	Multiple	Found 100 organized Telegram P&D channels that coordinates 2 P&D events per day on average. These events generates an aggregate artificial trading volume of 6 million USD a month. Furthermore, the authors reported that some online exchanges are active participants in the P&D schemes

(Continued on next page)

Ref.	Crime Type (D1)	Currency (D2)	Scale (D8)
[94]	P&D	BTC	The scam events generated a volume of transactions of 5,176 BTC in a single operation (more than 36M USD)
[64]	P&D	Multiple	The scam was able to create a price variance of around 15%
Inferred an estimation by applying mathematical analysis on cryptojacking campaigns extracted data			
[69]	CryptoJacking	Monero	2,770 unique cryptojacking domain, including 868 among Alexa top 100K. Estimates that the cryptojacking affects 10 million web users per month and generate over $59K daily by consuming 278K kWh extra power.
[90]	CryptoJacking	Monero	Estimates that each cryptoJacking campaign profit from 14.36 USD to 31,060.80 USD per month on average.
[135]	CryptoJacking	Multiple	3,487 mining domains, including 1,295 among Alexa top 1M. Furthermore, the authors reported that many mining domains have lived more than four years and received more than tens of millions of DNS resolutions.
[87]	CryptoJacking	Multiple	6,302 unique cryptojacking domains, including 828 among Alexa top 1M.
[107]	CryptoJacking	Monero	Estimates that 0.2% of Alexa top 1M domains contains mining scripts and it generates up to 340 USD per day.
Estimation based on extrapolating classification model results			
[23]	Mining/Jacking	Monero	Estimates that 0.2% of Alexa top 1M domains contains mining scripts and it generates up to 340 USD per day.
[168]	Services detection	Bitcoin	Estimates that the percentage of cybercrime-related addresses is 29.81% according to Bagging classifier, and 10.95% according to Gradient Boosting classifier.

(Continued on next page)

Ref.	Crime Type (D1)	Currency (D2)	Scale (D8)
[40]	HYIP	Ethereum	Estimates that 434 (0.15%) of the contracts on Ethereum platform before May 7, 2017 are Ponzi.
[41]	HYIP	Ethereum	Estimates that 507 (0.03%) of all the contracts before May 7, 2017 are Ponzi.

Table B.3: Sources used in the literature for dataset collection.

Ref.	Crime Type(D1)	Currency (D2)	Dataset Source (D4)
[153]	Services detection	Bitcoin	Bitcointalk.org and Cryptohyips.com.
[98]	Ransom	Bitcoin	Online fora.
[168]	Services detection	Bitcoin	Dataset provided by Chainalysis.com
[149]	HYIP	Bitcoin	Bitcointalk.org and Blockchain.info/tags.
[150]	Services detection	Bitcoin	Blockchain.info/tags, WalletExplorer.com, and BitcoinTalk.org.
[70]	Ransom	Bitcoin	Executing ransomware binaries and collected the ransom addresses from the memory dump, created files, and screenshots resulted from the ransomware. They further used search engines to find screenshots with ransom addresses provided by previous victims.
[21]	HYIP	Bitcoin	Bitcointalk.com
[68]	Phishing	Bitcoin	isco Systems, Inc. and Ukraine Cyberpolice.
[26]	Ransom	Bitcoin	Previously reported wannacry ransom addresses.
[82]	P&D	Multiple	Cryptocurrencies market data from online exchanges (Binance, Bittrex, Kraken, Kucoin and Lbank) using CCXT python library.

(Continued on next page)

Ref.	Crime Type (D1)	Currency (D2)	Dataset Source (D4)
[69]	Cryptojacking	Monero	Alexa top 100k domains. regularly visits the websites to Collect traces using Hash Based Profiler (search for hashing traces in the websites) and Stack Structure Based Profiler (search for heavy workloads with repeated behavioral patterns in the stack execution).
[40]	HYIP	Ethereum	Previous study.
[44]	Ransom	Bitcoin	Ransomware knowledge base, ransomware removal guides, online blogs, and available ransomware screenshots in different search engines image databases.
[154]	HYIP	Bitcoin	Bitcointalk.org subforums (Scam accusations, Games and Rounds, and Investment Games).
[148]	HYIP	Bitcoin	Pirate\@40 scheme addresses accessed through Bitcointalk.com and/or WalletExplorer.com
[80]	HYIP	Ethereum	Previous study.
[41]	HYIP	Ethereum	Open source smart contracts on the Ethereum platform.
[147]	HYIP	Bitcoin	Bitcointalk.org and WalletExplorer.com
[126]	Cryptojacking	Multiple	Nocoin blacklist.
[58]	Cryptojacking	Multiple	Device side-channel magnetic field signals generated from cryptocurrencies mining algorithms.
[146]	Honeypot	Ethereum	Previous study.
[117]	General	Ethereum	Etherscan.io.
[135]	Cryptojacking	Multiple	Monitoring Daily feed of suspicious URLs visited by Palo Alto Networks customers and Alexa top 1M domains.
[39]	P&D	Bitcoin	Mt. Gox leaked addresses transaction history.

(Continued on next page)

Ref.	Crime Type (D1)	Currency (D2)	Dataset Source (D4)
[87]	Cryptojacking	Multiple	Applying existing cryptojacking detection tools to scan Alexa top 1M domains.
[107]	Cryptojacking	Monero	Previously published mining script.
[113]	CryptoJacking	Monero	System runtime parameters of the studied computer/phone.
[156]	P&D	Multiple	Ccurrency price and volume (Binance exchange), fraud ads (Telegram P&D groups), and currency capitalization data (coinmarketcap.com).
[167]	P&D	Multiple	Telegeram channels.
[20]	HYIP	Ethereum	Open source smart contracts on the Ethereum platform.
[56]	General	Ethereum	Etherscamdb.
[6]	ML	Bitcoin	Elliptic dataset, a publicly available data set.
[5]	ML	Bitcoin	Elliptic dataset, a publicly available data set.
[71]	General	EOS	PeckShield and bot index.
[23]	Mining/Jacking	General	Alexa top 1M websites.
[139]	Mining/Jacking	General	Previous reports.
[46]	ML	Bitcoin	Online fora.
[122]	General	Bitcoin, Ethereum	CryptoScamDB and URLScan.io
[61]	Ransom	Bitcoin	Online repositories.
[93]	General	Ethereum	etherscan, cryptoscamdb, and GitHub.
[155]	DDoS	Bitcoin	Bitcointalk.org
[57]	DDoS	Bitcoin	Mt. Gox exchange leaked data, bitcoincharts.com, and bitcoinity.org
[3]	DDoS	Bitcoin	Bitfinex twitter feed, Bitfinex status page, and Google news search.
[19]	DDoS	Bitcoin	Previous study.
[49]	Cryptojacking	General	Windows Portable Executable (PE32) cryptominer samples registered with virustotal.com in 2018

(Continued on next page)

Ref.	Crime Type (D1)	Currency (D2)	Dataset Source (D4)
[50]	Cryptojacking	General	Collected and manually verified a large dataset of Android mining apps and found 728 mining apps
[169]	Cryptojacking	General	Crawl alexa top 1M and match the existing blacklist (uBlock, NoCoin and CoinBlockerLists) based on the source URL of WebAssembly/asm.js
[60]	Cryptojacking	General	Running a miner on their private server
[59]	Cryptojacking	Multiple	Profiling running miners
[94]	P&D	Bitcoin	Collect data manually from telegram and discord groups
[170]	Phishing	Ethereum	EtherScamDB and Etherscan.io
[171]	Phishing	Ethereum	1660 verified phishing addresses from etherscan.io
[38]	Phishing	Ethereum	1,683 phishing addresses from etherscan.io
[37]	Phishing	Ethereum	etherscan.io
[28]	HYIP	Bitcoin	15736 scam address by crawling BitcoinTalk
[121]	HYIP	Ethereum	Previous study
[54]	HYIP	Ethereum	Previous study
[99]	HYIP	Ethereum	Previous study
[55]	HYIP	Ethereum	Previous study
[140]	Cryptojacking	General	VirusShare
[97]	Ransom	Bitcoin	Black lists and google trend service
[85]	Cryptojacking	General	Extract features from the system calls and the CPU usage
[35]	Cryptojacking	Bitcoin, Monero, and Bytecoin	Collected network traffic
[64]	P&D	General	Discord and telegram channels
[112]	P&D	General	PumpOlymp, a website that collects and hosts comprehensive historical pump events on Telegram channels
[160]	Phishing	Ethereum	3,135 phishing accounts from etherscan

(Continued on next page)

Ref.	Crime Type (D1)	Currency (D2)	Dataset Source (D4)
[53]	HYIP	Ethereum	Previous study
[42]	HYIP	Ethereum	Previous study

Appendix C

doublebitcoin.win Addresses Graph Representation

In this section, we investigate whether the 5 bitcoin addresses related to the website `doublebitcoin.win` are related or not. For this purpose, we have built what we call the addresses/transactions connection graph. Specifically, we use the addresses transactions history to connect the 5 addresses. We have used the multi-input heuristic twice to infer more addresses related to the scam. The first time was applied on the 5 addresses and inferred 144 new addresses (level1). The second time was applied on the 144 addresses and inferred and 554 addresses (level2).

Our graph is shown in Figure C.1, the nodes represent bitcoin address and the edges represent coins flow. We have used different node colors to distinguish between the addresses as follows:

- **Red nodes** represents the initial 5 address.
- **Blue nodes** represents the addresses identified in multi-input heuristic first level.
- **Green nodes** represents the addresses identified in multi-input heuristic second level.
- **Black nodes** represents other addresses that appeared in the transactions history.

As shown in the graph, the 5 addresses and those identified using the multi-input heuristic are connected through different series of transactions which suggests they are related to the same scammer.

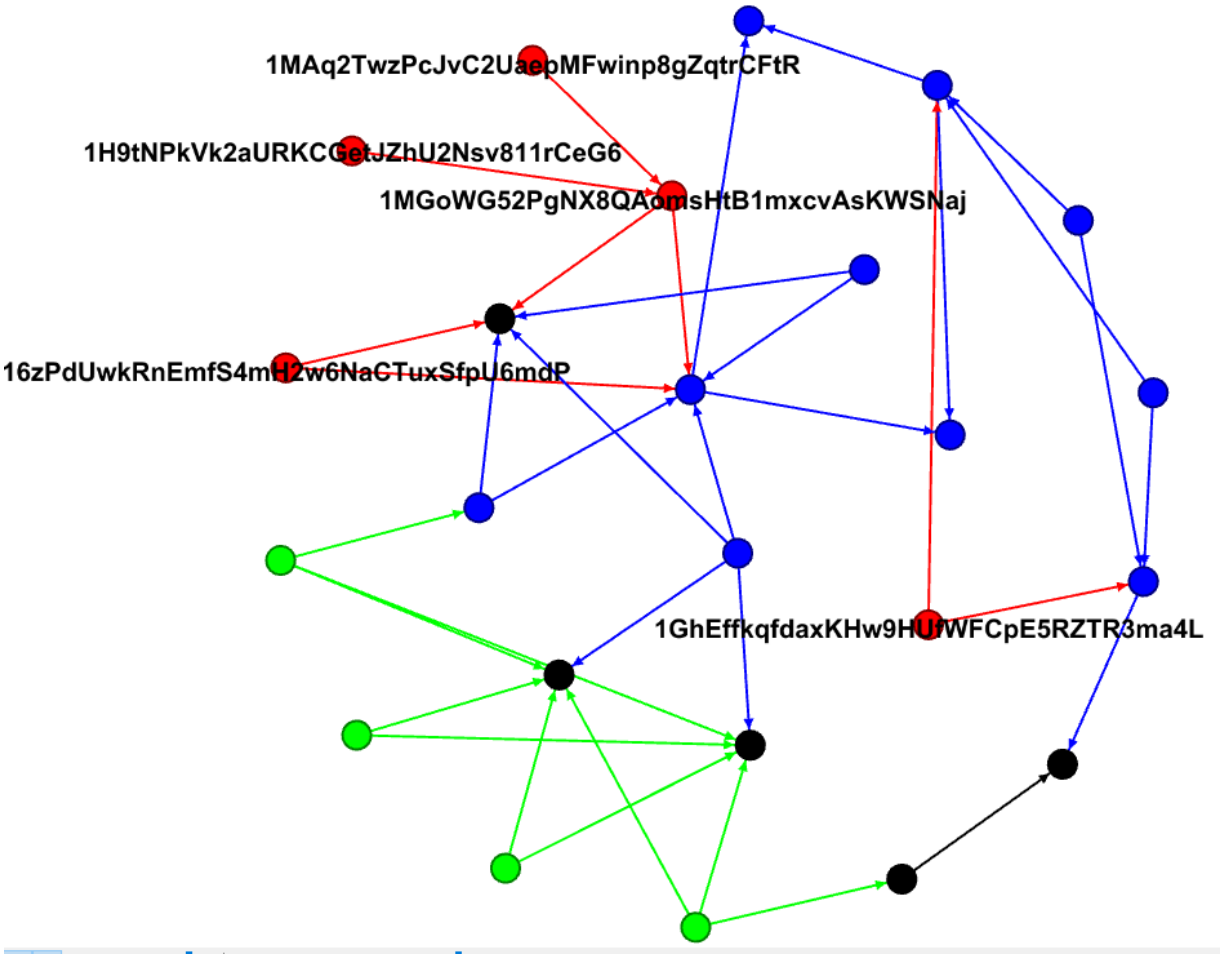

Figure C.1: A graph representation of the doublebitcoin.win BGS domain addresses.

References

[1] Bitcoin wiki. `https://en.bitcoin.it/wiki/From_address`. Last accessed 2021.

[2] Frank W Abagnale. *Catch Me If You Can: The Amazing True Story of the Youngest and Most Daring Con Man [most Extraordinary Liar] in the History of Fun and Profit*. Broadway, 2000.

[3] Abhishta Abhishta, Reinoud Joosten, Sergey Dragomiretskiy, and Lambert JM Nieuwenhuis. Impact of successful ddos attacks on a major crypto-currency exchange. In *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 379–384. IEEE, 2019.

[4] Cuneyt G Akcora, Yitao Li, Yulia R Gel, and Murat Kantarcioglu. Bitcoinheist: Topological data analysis for ransomware prediction on the bitcoin blockchain. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence*, 2020.

[5] Ismail Alarab, Simant Prakoonwit, and Mohamed Ikbal Nacer. Comparative analysis using supervised learning methods for anti-money laundering in bitcoin. In *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*, pages 11–17, 2020.

[6] Ismail Alarab, Simant Prakoonwit, and Mohamed Ikbal Nacer. Competence of graph convolutional networks for anti-money laundering in bitcoin blockchain. In *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*, pages 23–27, 2020.

[7] Chad Albrecht, Kristopher McKay Duffin, Steven Hawkins, and Victor Manuel Morales Rocha. The use of cryptocurrencies in the money laundering process. *Journal of Money Laundering Control*, 2019.

[8] Dariush Alimohammadi. Meta-tag: a means to control the process of web indexing. *Online Information Review*, 2003.

[9] Diego Raphael Amancio, Cesar Henrique Comin, Dalcimar Casanova, Gonzalo Travieso, Odemir Martinez Bruno, Francisco Aparecido Rodrigues, and Luciano da Fontoura Costa. A systematic comparison of supervised classifiers. *PloS one*, 9(4):e94137, 2014.

[10] Leila A Amineddoleh. Are you faux real: An examination of art forgery and the legal tools protecting art collectors. *Cardozo Arts & Ent. LJ*, 34:59, 2016.

[11] Alan Appelbaum. Another look at the assassination of pertinax and the accession of julianus. *Classical Philology*, 102(2):198–207, 2007.

[12] Daniel Arp, Spreitzenbarth Michael, Hubner Malte, Gascon Hugo, Rieck Konrad, and Siemens C. E. R. T. Drebin: Effective and explainable detection of android malware in your pocket. *Ndss*, 14:23–26, 2014.

[13] Ahmet ARSLAN. On the usefulness of html meta elements for web retrieval. *Eskişehir Technical University Journal of Science and Technology A-Applied Sciences and Engineering*, 21(1):182–198, 2020.

[14] A. M. Aswini and P. Vinod. Droid permission miner: Mining prominent permissions for android malware analysis. In *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, pages 81–86, Feb 2014.

[15] Emad Badawi and Guy-Vincent Jourdan. Cryptocurrencies Emerging Threats and Defensive Mechanisms: A Systematic Literature Review. *IEEE Access*, 8, 2020.

[16] Emad Badawi, Guy-Vincent Jourdan, Gregor Bochmann, and Iosif-Viorel Onut. An automatic detection and analysis of the bitcoin generator scam. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 407–416, Los Alamitos, CA, USA, sep 2020. IEEE Computer Society.

[17] Emad Badawi, Guy-Vincent Jourdan, Gregor Bochmann, and Iosif-Viorel Onut. Automatic Detection and Analysis of the "Game Hack" Scam. *Journal of Web Engineering*, 18(8), 2020.

[18] Emad Badawi, Guy-Vincent Jourdan, Gregor Bochmann, Iosif-Viorel Onut, and Jason Flood. The "game hack" scam. In *ICWE 2019. Springer LNCS 11496*, pages 280–295, 2019.

[19] Ui-Jun Baek, Se-Hyun Ji, Jee Tae Park, Min-Seob Lee, Jun-Sang Park, and Myung-Sup Kim. Ddos attack detection on bitcoin ecosystem using deep-learning. In *2019*

20th Asia-Pacific Network Operations and Management Symposium (APNOMS), pages 1–4. IEEE, 2019.

[20] Massimo Bartoletti, Salvatore Carta, Tiziana Cimoli, and Roberto Saia. Dissecting ponzi schemes on ethereum: identification, analysis, and impact. *Future Generation Computer Systems*, 102:259–277, 2020.

[21] Massimo Bartoletti, Barbara Pes, and Sergio Serusi. Data mining for detecting bitcoin ponzi schemes. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 75–84. IEEE, 2018.

[22] Mark William Becker. *Greek culture and the ideology of Roman Empire in Cicero's" Verrine Orations"*. PhD thesis, Princeton University, 1996.

[23] Weikang Bian, Wei Meng, and Mingxue Zhang. Minethrottle: Defending against wasm in-browser cryptojacking. In *Proceedings of The Web Conference 2020*, pages 3112–3118, 2020.

[24] Morvareed Bidgoli and Jens Grossklags. "hello. this is the irs calling.": A case study on scams, extortion, impersonation, and phone spoofing. In *Electronic Crime Research (eCrime), 2017 APWG Symposium on*, pages 57–69. IEEE, 2017.

[25] G Martin Bingisser. Data privacy and breach reporting: Compliance with various state laws. *Washington Journal of Law, Technology & Arts*, 4(3):9, 2008.

[26] Stefano Bistarelli, Matteo Parroccini, and Francesco Santini. Visualizing bitcoin flows of ransomware: Wannacry one week later. In *ITASEC*, 2018.

[27] Marzieh Bitaab, Haehyun Cho, Adam Oest, Penghui Zhang, Zhibo Sun, Rana Pourmohamad, Doowon Kim, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, et al. Scam pandemic: How attackers exploit public fear through phishing. In *2020 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–10. IEEE, 2020.

[28] Yazan Boshmaf, Charitha Elvitigala, Husam Al Jawaheri, Primal Wijesekera, and Mashael Al Sabah. Investigating mmm ponzi scheme on bitcoin. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pages 519–530, 2020.

[29] Christian Brenig, Rafael Accorsi, and Günter Müller. Economic analysis of cryptocurrency backed money laundering. In *ECIS*, 2015.

[30] Danton Bryans. Bitcoin and money laundering: Mining for an effective solution. 89 lnd, 2014.

[31] Tom Buchanan and Monica T Whitty. The online dating romance scam: causes and consequences of victimhood. *Psychology, Crime & Law*, 20(3):261–283, 2014.

[32] Carolyn Budd and Jessica Anderson. *Consumer fraud in Australasia: Results of the Australasian consumer fraud taskforce online Australia surveys 2008 and 2009*. Australian Institute of Criminology, 2011.

[33] Malcolm Campbell-Verduyn. Bitcoin, crypto-coins, and global anti-money laundering governance. *Crime, Law and Social Change*, 69(2):283–305, 2018.

[34] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[35] Maurantonio Caprolu, Simone Raponi, Gabriele Oligeri, and Roberto Di Pietro. Cryptomining makes noise: Detecting cryptojacking via machine learning. *Computer Communications*, 171:126–139, 2021.

[36] Dennis Challinger. Refund fraud in retail stores. *Security Journal*, 7(1):27–35, 1996.

[37] Liang Chen, Jiaying Peng, Yang Liu, Jintang Li, Fenfang Xie, and Zibin Zheng. Phishing scams detection in ethereum transaction network. *ACM Transactions on Internet Technology (TOIT)*, 21(1):1–16, 2020.

[38] Weili Chen, Xiongfeng Guo, Zhiguang Chen, Zibin Zheng, and Yutong Lu. Phishing scam detection on ethereum: Towards financial security for blockchain ecosystem. In *IJCAI*, pages 4506–4512, 2020.

[39] Weili Chen, YueJin Xu, Zibin Zheng, Yuren Zhou, Jianxun Eileen Yang, and Jing Bian. Detecting" pump & dump schemes" on cryptocurrency market using an improved apriori algorithm. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 293–2935. IEEE, 2019.

[40] Weili Chen, Zibin Zheng, Jiahui Cui, Edith Ngai, Peilin Zheng, and Yuren Zhou. Detecting ponzi schemes on ethereum: Towards healthier blockchain technology. In *Proceedings of the 2018 World Wide Web Conference*, pages 1409–1418, 2018.

[41] Weili Chen, Zibin Zheng, Edith C-H Ngai, Peilin Zheng, and Yuren Zhou. Exploiting blockchain data to detect smart ponzi schemes on ethereum. *IEEE Access*, 7:37575–37586, 2019.

[42] Weimin Chen, Xinran Li, Yuting Sui, Ningyu He, Haoyu Wang, Lei Wu, and Xiapu Luo. Sadponzi: Detecting and characterizing ponzi schemes in ethereum smart contracts. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(2):1–30, 2021.

[43] Jason W. Clark and Damon McCoy. There are no free ipads: An analysis of survey scams as a business. In *Presented as part of the 6th USENIX Workshop on Large-Scale Exploits and Emergent Threats*, Washington, D.C., 2013. USENIX.

[44] Mauro Conti, Ankit Gangwal, and Sushmita Ruj. On the economic significance of ransomware campaigns: A bitcoin transactions perspective. *Computers & Security*, 79:162–189, 2018.

[45] C COSH. The immigrant sponsorship scam. *ALBERTA REPORT/NEWSMAGAZINE*, 22(8):30–31, 1995.

[46] Jesse Crawford and Yong Guan. Knowing your bitcoin customer: Money laundering in the bitcoin economy. In *2020 13th International Conference on Systematic Approaches to Digital Forensic Engineering (SADFE)*, pages 38–45. IEEE, 2020.

[47] Qian Cui, Guy-Vincent Jourdan, Gregor V. Bochmann, Russell Couturier, and Iosif-Viorel Onut. Tracking phishing attacks over time. In *International World Wide Web Conferences Steering Committee*, pages 667–676, 2017.

[48] Bart Custers, Jan-Jaap Oerlemans, and Ronald Pool. Laundering the profits of ransomware: Money laundering methods for vouchers and cryptocurrencies. *European Journal of Crime, Criminal Law and Criminal Justice*, 28(2):121–152, 2020.

[49] Hamid Darabian, Sajad Homayounoot, Ali Dehghantanha, Sattar Hashemi, Hadis Karimipour, Reza M Parizi, and Kim-Kwang Raymond Choo. Detecting cryptomining malware: a deep learning approach for static and dynamic analysis. *Journal of Grid Computing*, pages 1–11, 2020.

[50] Stanislav Dashevskyi, Yury Zhauniarovich, Olga Gadyatskaya, Aleksandr Pilgun, and Hamza Ouhssain. Dissecting android cryptocurrency miners. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pages 191–202, 2020.

[51] Oscar Delgado-Mohatar, José María Sierra-Cámara, and Eloy Anguiano. Blockchain-based semi-autonomous ransomware. *Future Generation Computer Systems*, 2020.

[52] Stephen Ellis. *This present darkness: A history of Nigerian organized crime*. Oxford University Press, USA, 2016.

[53] Shuhui Fan, Shaojing Fu, Haoran Xu, and Xiaochun Cheng. Al-spsd: Anti-leakage smart ponzi schemes detection in blockchain. *Information Processing & Management*, 58(4):102587, 2021.

[54] Shuhui Fan, Shaojing Fu, Haoran Xu, and Chengzhang Zhu. Expose your mask: Smart ponzi schemes detection on blockchain. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2020.

[55] Shuhui Fan, Haoran Xu, Shaojing Fu, and Ming Xu. Smart ponzi scheme detection using federated learning. In *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 881–888, 2020.

[56] Steven Farrugia, Joshua Ellul, and George Azzopardi. Detection of illicit accounts over the ethereum blockchain. *Expert Systems with Applications*, 150:113318, 2020.

[57] Amir Feder, Neil Gandal, JT Hamrick, and Tyler Moore. The impact of ddos and other security shocks on bitcoin currency exchanges: Evidence from mt. gox. *Journal of Cybersecurity*, 3(2):137–144, 2017.

[58] Ankit Gangwal and Mauro Conti. Cryptomining cannot change its spots: Detecting covert cryptomining using magnetic side-channel. *IEEE Transactions on Information Forensics and Security*, 2019.

[59] Ankit Gangwal, Samuele Giuliano Piazzetta, Gianluca Lain, and Mauro Conti. Detecting covert cryptomining using hpc. In *International Conference on Cryptology and Network Security*, pages 344–364. Springer, 2020.

[60] Fábio Gomes and Miguel Correia. Cryptojacking detection with cpu usage metrics. In *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pages 1–10, 2020.

[61] Parth S Goyal, Akshat Kakkar, Gopika Vinod, and Gigi Joseph. Crypto-ransomware detection using behavioural analysis. In *Reliability, Safety and Hazard Assessment for Risk-Based Technologies*, pages 239–251. Springer, 2020.

[62] Lars Haffke, Mathias Fromberger, and Patrick Zimmermann. Cryptocurrencies and anti-money laundering: the shortcomings of the fifth aml directive (eu) and how to address them. *Journal of Banking Regulation*, pages 1–14, 2019.

[63] Saqib Hakak, Wazir Zada Khan, Muhammad Imran, Kim-Kwang Raymond Choo, and Muhammad Shoaib. Have you been a victim of covid-19-related cyber incidents? survey, taxonomy, and mitigation strategies. *Ieee Access*, 8:124134–124144, 2020.

[64] JT Hamrick, Farhang Rouhi, Arghya Mukherjee, Amir Feder, Neil Gandal, Tyler Moore, and Marie Vasek. An examination of the cryptocurrency pump-and-dump ecosystem. *Information Processing & Management*, 58(4):102506, 2021.

[65] David Harley, Martijn Grooten, Steven Burn, and Craig Johnston. My pc has 32,539 errors: how telephone support scams really work. *Virus Bulletin*, 2012.

[66] Elina Hartikainen. The nigerian scam: easy money on the internet, but for whom. In *Unpublished paper presented at Michicagoan Conference and blogged online at http://www. antropologi. info/blog/anthropology*, 2006.

[67] Alfred Hasbrouck. Gregor mcgregor and the colonization of poyais, between 1820 and 1824. *The Hispanic American Historical Review*, 7(4):438–459, 1927.

[68] Artsiom Holub and Jeremiah O'Connor. Coinhoarder: Tracking a ukrainian bitcoin phishing ring dns style. In *2018 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–5. IEEE, 2018.

[69] Geng Hong, Zhemin Yang, Sen Yang, Lei Zhang, Yuhong Nan, Zhibo Zhang, Min Yang, Yuan Zhang, Zhiyun Qian, and Haixin Duan. How you get shot in the back: A systematical study about cryptojacking in the real world. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1701–1713, 2018.

[70] Danny Yuxing Huang, Maxwell Matthaios Aliapoulios, Vector Guo Li, Luca Invernizzi, Elie Bursztein, Kylie McRoberts, Jonathan Levin, Kirill Levchenko, Alex C Snoeren, and Damon McCoy. Tracking ransomware end-to-end. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 618–631. IEEE, 2018.

[71] Yuheng Huang, Haoyu Wang, Lei Wu, Gareth Tyson, Xiapu Luo, Run Zhang, Xuanzhe Liu, Gang Huang, and Xuxian Jiang. Understanding (mis) behavior on the eosio blockchain. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(2):1–28, 2020.

[72] Giacomo Ibba, Giuseppe Antonio Pierro, and Marco Di Francesco. Evaluating machine-learning techniques for detecting smart ponzi schemes. In *2021 IEEE/ACM 4th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, pages 34–40, 2021.

[73] F. Idrees and M. Rajarajan. Investigating the android intents and permissions for malware detection. In *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 354–358, Oct 2014.

[74] Jelena Isacenkova, Olivier Thonnard, Andrei Costin, Aurélien Francillon, and David Balzarotti. Inside the scam jungle: A closer look at 419 scam email operations. *EURASIP Journal on Information Security*, 2014(1):1–18, 2014.

[75] Samireh Jalali and Claes Wohlin. Systematic literature studies: database searches vs. backward snowballing. In *Proceedings of the 2012 ACM-IEEE international symposium on empirical software engineering and measurement*, pages 29–38. IEEE, 2012.

[76] Jean-Luc. Liste d'escroqueries liées à bitcoin et aux cryptomonnaies - bitcoin.fr. `http://bit.ly/2Pi5YN7`, 2020.

[77] L. Jing. Mobile internet malicious application detection method based on support vector machine. In *2017 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, pages 260–263, May 2017.

[78] Benjamin Johnson, Aron Laszka, Jens Grossklags, Marie Vasek, and Tyler Moore. Game-theoretic analysis of ddos attacks against bitcoin mining pools. In *International Conference on Financial Cryptography and Data Security*, pages 72–86. Springer, 2014.

[79] Timothy Johnson. The financial revolution of the late seventeenth century. In *Ethics in Quantitative Finance*, pages 103–125. Springer, 2017.

[80] Eunjin Jung, Marion Le Tilly, Ashish Gehani, and Yunjie Ge. Data mining-based ethereum fraud detection. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 266–273. IEEE, 2019.

[81] Daniel Jurafsky and James H. Martin. Markov assumption. stanford.io/29zsjAy, 2014.

[82] Josh Kamps and Bennett Kleinberg. To the moon: defining and detecting cryptocurrency pump-and-dumps. *Crime Science*, 7(1):18, 2018.

[83] Ilker KARA and Murat AYDOS. Cyber fraud: Detection and analysis of the crypto-ransomware. In *2020 11th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 0764–0769, 2020.

[84] Christos Karapapas, Iakovos Pittaras, Nikos Fotiou, and George C Polyzos. Ransomware as a service using smart contracts and ipfs. *arXiv preprint arXiv:2003.04426*, 2020.

[85] Rupesh Raj Karn, Prabhakar Kudva, Hai Huang, Sahil Suneja, and Ibrahim M. Elfadel. Cryptomining detection in container clouds using system calls and explainable machine learning. *IEEE Transactions on Parallel and Distributed Systems*, 32(3):674–691, 2021.

[86] Vittal Katikireddi. Food scam. *BMJ*, 326(Suppl S3), 2003.

[87] Amin Kharraz, Zane Ma, Paul Murley, Charles Lever, Joshua Mason, Andrew Miller, Nikita Borisov, Manos Antonakakis, and Michael Bailey. Outguard: Detecting in-browser covert cryptocurrency mining in the wild. In *The World Wide Web Conference*, pages 840–852, 2019.

[88] Amin Kharraz, William Robertson, and Engin Kirda. Surveylance: Automatically detecting online survey scams. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 70–86. IEEE, 2018.

[89] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.

[90] Radhesh Krishnan Konoth, Emanuele Vineti, Veelasha Moonsamy, Martina Lindorfer, Christopher Kruegel, Herbert Bos, and Giovanni Vigna. Minesweeper: An in-depth look into drive-by cryptocurrency mining and its defense. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1714–1730, 2018.

[91] Christian Kopp, James Sillitoe, Iqbal Gondal, and Robert Layton. THE ONLINE ROMANCE SCAM: A COMPLEX TWO-LAYER SCAM. *Journal of Psychological & Educational Research*, 24(2):144–161, 2016.

[92] Nir Kshetri and Jeffrey Voas. Do crypto-currencies fuel ransomware? *IT professional*, 19(5):11–15, 2017.

[93] Nitesh Kumar, Ajay Singh, Anand Handa, and Sandeep Kumar Shukla. Detecting malicious accounts on the ethereum blockchain with supervised learning. In *International Symposium on Cyber Security Cryptography and Machine Learning*, pages 94–109. Springer, 2020.

[94] Massimo La Morgia, Alessandro Mei, Francesco Sassi, and Julinda Stefa. Pump and dumps in the bitcoin era: Real time detection of cryptocurrency market manipulations. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–9, 2020.

[95] Jonathan Larson, Bryan Tower, Duane Hadfield, Darren Edge, and Christopher White. Using web-scale graph analytics to counter technical support scams. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3968–3971. IEEE, 2018.

[96] Sophie Le Page, Guy-Vincent Jourdan, Gregor V Bochmann, Jason Flood, and Iosif-Viorel Onut. Using url shorteners to compare phishing and malware attacks. In *In APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–13. IEEE, 2018.

[97] Hannarae Lee and Kyung-Shick Choi. Interrelationship between bitcoin, ransomware, and terrorist activities: Criminal opportunity assessment via cyber-routine activities theoretical framework. *Victims & Offenders*, 16(3):363–384, 2021.

[98] Kevin Liao, Ziming Zhao, Adam Doupé, and Gail-Joon Ahn. Behind closed doors: measurement and analysis of cryptolocker ransoms in bitcoin. In *2016 APWG eCrime*, pages 1–13. IEEE, 2016.

[99] Yincheng Lou, Yanmei Zhang, and Shiping Chen. Ponzi contracts detection based on improved convolutional neural network. In *2020 IEEE International Conference on Services Computing (SCC)*, pages 353–360, 2020.

[100] Sagwadi Mabunda. Cryptocurrency: The new face of cyber money laundering. In *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–6. IEEE, 2018.

[101] Artem A Maksutov, Maxim S Alexeev, Natalia O Fedorova, and Daniil A Andreev. Detection of blockchain transactions used in blockchain mixer of coin join type. In *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 274–277. IEEE, 2019.

[102] Manzhi Yang and QiaoYan Wen. Detecting android malware with intensive feature engineering. In *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 157–161, Aug 2016.

[103] Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. Dial one for scam: A large-scale analysis of technical support scams. *arXiv preprint arXiv:1607.06891*, 2016.

[104] Tyler Moore, Jie Han, and Richard Clayton. The postmodern ponzi scheme: Empirical analysis of high-yield investment programs. In Angelos D. Keromytis, editor, *Financial Cryptography and Data Security*, pages 41–56, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[105] Malte Möser, Rainer Böhme, and Dominic Breuker. An inquiry into money laundering tools in the bitcoin ecosystem. In *2013 APWG eCrime Researchers Summit*, pages 1–14. Ieee, 2013.

[106] Mohd Faizal Mubarak, Saadiah Yahya, and Ahmad Faisal Adham Shaazi. A review of phone scam activities in malaysia. In *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, pages 441–446. IEEE, 2019.

[107] Marius Musch, Christian Wressnegger, Martin Johns, and Konrad Rieck. Thieves in the browser: Web-based cryptojacking in the wild. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–10, 2019.

[108] Rennie Naidoo. A multi-level influence model of covid-19 themed cybercrime. *European Journal of Information Systems*, 29(3):306–321, 2020.

[109] Satoshi Nakamoto and A Bitcoin. A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, 2008.

[110] Ewan Nettleton. End of the line for phone scams? *Journal of Database Marketing & Customer Strategy Management*, 13(3):231–235, 2006.

[111] Graeme R Newman, Megan M McNally, et al. Identity theft literature review. 2005.

[112] Huy Nghiem, Goran Muric, Fred Morstatter, and Emilio Ferrara. Detecting cryptocurrency pump-and-dump frauds using market and social signals. *Expert Systems with Applications*, page 115284, 2021.

[113] Rui Ning, Cong Wang, ChunSheng Xin, Jiang Li, Liuwan Zhu, and Hongyi Wu. Capjack: Capture in-browser crypto-jacking by deep capsule network through behavioral analysis. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 1873–1881. IEEE, 2019.

[114] Andrew Odlyzko. Newton's financial misadventures in the south sea bubble. *Notes and Records: the Royal Society journal of the history of science*, 73(1):29–59, 2019.

[115] online. Google safe browsing api. https://goo.gl/4yAFyQ, 2018. Last accessed 2021.

[116] online. Virustotal. https://www.virustotal.com/, 2018. Last accessed 2021.

[117] Michał Ostapowicz and Kamil Żbikowski. Detecting fraudulent accounts on blockchain: A supervised approach. In *International Conference on Web Information Systems Engineering*, pages 18–31. Springer, 2019.

[118] Oxford English Oxford. *Oxford English Dictionary*. Oxford: Oxford University Press, 2009.

[119] Helen Paul. *The South Sea Bubble: an economic history of its origins and consequences*. Routledge, 2010.

[120] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[121] Jianxi Peng and Guijiao Xiao. Detection of smart ponzi schemes using opcode. In Zibin Zheng, Hong-Ning Dai, Xiaodong Fu, and Benhui Chen, editors, *Blockchain and Trustworthy Systems*, pages 192–204, Singapore, 2020. Springer Singapore.

[122] Ross Phillips and Heidi Wilder. Tracing cryptocurrency scams: Clustering replicated advance-fee and phishing websites. *arXiv preprint arXiv:2005.14440*, 2020.

[123] Nadia Pocher. The open legal challenges of pursuing aml/cft accountability within privacy-enhanced iom ecosystems. In *DLT@ ITASEC*, 2020.

[124] Zachée Pouga Tinhaga. From avoiding 'double taxation'yesterday to avoiding 'double non-taxation'today: The urgent need for an international tax regime based on unitary tax principles. 2016.

[125] Sampsa Rauti and Ville Leppänen. "you have a potential hacker's infection": A study on technical support scams. In *2017 IEEE International Conference on Computer and Information Technology (CIT)*, pages 197–203. IEEE, 2017.

[126] Muhammad Amirrudin Razali and Shafiza Mohd Shariff. Cmblock: In-browser detection and prevention cryptojacking tool using blacklist and behavior-based detection method. In *International Visual Informatics Conference*, pages 404–414. Springer, 2019.

[127] Fergal Reid and Martin Harrigan. An analysis of anonymity in the bitcoin system. In *Security and privacy in social networks*, pages 197–223. Springer, 2013.

[128] Diana Mergenovna Sat, AB Kasatkin, IA Kornev, GO Krylov, K Evgenyevich, et al. Investigation of money laundering methods through cryptocurrency. *Journal of theoretical and applied information technology*, 83(2):244–254, 2016.

[129] Jennifer Selleck. What is a cpa network? cost per action explained. bit.ly/2uHRWsl, 2013.

[130] Junwoo Seo, Mookyu Park, Haengrok Oh, and Kyungho Lee. Money laundering in the bitcoin network: Perspective of mixing services. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1403–1405. IEEE, 2018.

[131] OLIVIA Solon. Frank abagnale on the death of the con artist and the rise of cybercrime, 2017.

[132] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. Bitiodine: Extracting intelligence from the bitcoin network. In *International Conference on Financial Cryptography and Data Security*, pages 457–468. Springer, 2014.

[133] Kevin Springborn and Paul Barford. Impression fraud in on-line advertising via pay-per-view networks. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*, pages 211–226, Washington, D.C., 2013. USENIX.

[134] Bharat Srinivasan, Athanasios Kountouras, Najmeh Miramirkhani, Monjur Alam, Nick Nikiforakis, Manos Antonakakis, and Mustaque Ahamad. Exposing search and advertisement abuse tactics and infrastructure of technical support scammers. In *WWW'18*, pages 319–328, 2018.

[135] Oleksii Starov, Yuchen Zhou, and Jun Wang. Detecting malicious campaigns in obfuscated javascript with scalable behavioral analysis. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 218–223. IEEE, 2019.

[136] Oleksii Starov, Yuchen Zhou, Xiao Zhang, Najmeh Miramirkhani, and Nick Nikiforakis. Betrayed by your dashboard: Discovering malicious campaigns via web analytics. In *Proceedings of the 2018 World Wide Web Conference*, pages 227–236, 2018.

[137] Karthika Subramani, Xingzi Yuan, Omid Setayeshfar, Phani Vadrevu, Kyu Hyung Lee, and Roberto Perdisci. Measuring abuse in web push advertising. *arXiv preprint arXiv:2002.06448*, 2020.

[138] Rashid Tahir, Muhammad Huzaifa, Anupam Das, Mohammad Ahmad, Carl Gunter, Fareed Zaffar, Matthew Caesar, and Nikita Borisov. Mining on someone else's dime: Mitigating covert mining operations in clouds and enterprises. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 287–310. Springer, 2017.

[139] Dmitry Tanana. Behavior-based detection of cryptojacking malware. In *2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*, pages 0543–0545. IEEE, 2020.

[140] Dmitry Tanana and Galina Tanana. Advanced behavior-based technique for cryptojacking malware detection. In *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–4. IEEE, 2020.

[141] Fabian Teichmann and Marie-Christin Falker. Blockchain: Implications of the impending token economy. In *Institute of Scientific Communications Conference*, pages 1551–1565. Springer, 2019.

[142] Fabian Teichmann and Marie-Christin Falker. Money laundering through cryptocurrencies. In *13th International Scientific and Practical Conference-Artificial Intelligence Anthropogenic nature Vs. Social Origin*, pages 500–511. Springer, 2020.

[143] Gerhard Thür. Transaction costs in athenian law. *Law and Transaction Costs in the Ancient Economy*, pages 36–50, 2015.

[144] P. Tiwari, G. Tere, and P. Singh. Malware detection in android application by rigorous analysis of decompiled source code. In *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, pages 1–6, Aug 2016.

[145] Christof Ferreira Torres, Mathis Baden, and Radu State. Towards usable protection against honeypots. In *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–2. IEEE, 2020.

[146] Christof Ferreira Torres, Mathis Steichen, et al. The art of the scam: Demystifying honeypots in ethereum smart contracts. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1591–1607, 2019.

[147] Kentaroh Toyoda, P. Takis Mathiopoulos, and Tomoaki Ohtsuki. A novel methodology for hyip operators' bitcoin addresses identification. *IEEE Access*, 7:74835–74848, 2019.

[148] Kentaroh Toyoda, Tomoaki Ohtsuki, and P. Mathiopoulos. Time series analysis for bitcoin transactions: The case of pirate@ 40's hyip scheme. In *IEEE ICDMW'18*, pages 151–155. IEEE, 2018.

[149] Kentaroh Toyoda, Tomoaki Ohtsuki, and P Takis Mathiopoulos. Identification of high yielding investment programs in bitcoin via transactions pattern analysis. In *GLOBECOM 2017*, pages 1–6. IEEE, 2017.

[150] Kentaroh Toyoda, Tomoaki Ohtsuki, and P Takis Mathiopoulos. Multi-class bitcoin-enabled service identification based on transaction history summarization. In *iThings/ GreenCom/ CPSCom/ SmartData/ Blockchain/ CIT/ Cybermatics 2018*, pages 1153–1160. IEEE, 2018.

[151] Huahong Tu, Adam Doupé, Ziming Zhao, and Gail-Joon Ahn. Users really do answer telephone scams. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1327–1340, 2019.

[152] Adam Turner, Stephen McCombie, and Allon Uhlmann. Follow the money: Revealing risky nodes in a ransomware-bitcoin network. In *Proceedings of the 54th Hawaii International Conference on System Sciences*, page 1560, 2021.

[153] Marie Vasek and Tyler Moore. There's no free lunch, even using bitcoin: Tracking the popularity and profits of virtual currency scams. In *International conference on financial cryptography and data security*, pages 44–61. Springer, 2015.

[154] Marie Vasek and Tyler Moore. Analyzing the bitcoin ponzi scheme ecosystem. In *International Conference on Financial Cryptography and Data Security*, pages 101–112. Springer, 2018.

[155] Marie Vasek, Micah Thornton, and Tyler Moore. Empirical analysis of denial-of-service attacks in the bitcoin ecosystem. In *International conference on financial cryptography and data security*, pages 57–71. Springer, 2014.

[156] Friedhelm Victor and Tanja Hagemann. Cryptocurrency pump and dump schemes: Quantification and detection. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 244–251. IEEE, 2019.

[157] Thomas Vissers, Jan Spooren, Pieter Agten, Dirk Jumpertz, Peter Janssen, Marc Van Wesemael, Frank Piessens, Wouter Joosen, and Lieven Desmet. Exploring the ecosystem of malicious domain registrations in the. eu tld. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 472–493. Springer, 2017.

[158] Kristin Weber, Andreas E Schütz, Tobias Fertig, and Nicholas H Müller. Exploiting the human factor: Social engineering attacks on cryptocurrency users. In *International Conference on Human-Computer Interaction*, pages 650–668. Springer, 2020.

[159] Chun Wei, Alan Sprague, Gary Warner, and Anthony Skjellum. Clustering spam domains and destination websites: Digital forensics with data mining. *Journal of Digital Forensics, Security and Law*, 5(1):2, 2010.

[160] Haixian Wen, Junyuan Fang, Jiajing Wu, and Zibin Zheng. Transaction-based hidden strategies against general phishing detection framework on ethereum. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2021.

[161] L Whitney. Covidlock ransomware exploits coronavirus with malicious android app. *online]., Mar*, 17, 2020.

[162] Monica T Whitty. Anatomy of the online dating romance scam. *Security Journal*, 28(4):443–455, 2015.

[163] Monica T Whitty. Do you love me? psychological characteristics of romance scam victims. *Cyberpsychology, behavior, and social networking*, 21(2):105–109, 2018.

[164] Monica T Whitty and Tom Buchanan. The online romance scam: A serious cybercrime. *CyberPsychology, Behavior, and Social Networking*, 15(3):181–183, 2012.

[165] Dimaz Ankaa Wijaya, Joseph K Liu, Ron Steinfeld, and Dongxi Liu. Risk of asynchronous protocol update: Attacks to monero protocols. In *Australasian Conference on Information Security and Privacy*, pages 307–321. Springer, 2019.

[166] Shuangke Wu, Yanjiao Chen, Minghui Li, Xiangyang Luo, Zhe Liu, and Lan Liu. Survive and thrive: A stochastic game for ddos attacks in bitcoin mining pools. *IEEE/ACM Transactions on Networking*, 28(2):874–887, 2020.

[167] Jiahua Xu and Benjamin Livshits. The anatomy of a cryptocurrency pump-and-dump scheme. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1609–1625, 2019.

[168] Haohua Sun Yin and Ravi Vatrapu. A first estimation of the proportion of cybercriminal entities in the bitcoin ecosystem using supervised machine learning. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3690–3699. IEEE, 2017.

[169] Guorui Yu, Guangliang Yang, Tongxin Li, Xinhui Han, Shijie Guan, Jialong Zhang, and Guofei Gu. Minergate: A novel generic and accurate defense solution against web based cryptocurrency mining attacks. In *China Cyber Security Annual Conference*, pages 50–70. Springer, Singapore, 2020.

[170] Qi Yuan, Baoying Huang, Jie Zhang, Jiajing Wu, Haonan Zhang, and Xi Zhang. Detecting phishing scams on ethereum based on transaction records. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2020.

[171] Zihao Yuan, Qi Yuan, and Jiajing Wu. Phishing detection on ethereum via learning representation of transaction subgraphs. In Zibin Zheng, Hong-Ning Dai, Xiaodong Fu, and Benhui Chen, editors, *Blockchain and Trustworthy Systems*, pages 178–191, Singapore, 2020. Springer Singapore.

[172] Rongxin Zheng, Cuiwen Ying, Jun Shao, Guiyi Wei, Hongyang Yan, Jianmin Kong, Yekun Ren, Hang Zhang, and Weiguang Hou. New game-theoretic analysis of ddos attacks against bitcoin mining pools with defence cost. In *International Conference on Network and System Security*, pages 567–580. Springer, 2019.

[173] Zhongyuan Qin, Yuqing Xu, Yuxing Di, Qunfang Zhang, and Jie Huang. Android malware detection based on permission and behavior analysis. In *International Conference on Cyberspace Technology (CCT 2014)*, pages 1–4, Nov 2014.

[174] Aaron Zimba, Mumbi Chishimba, Christabel Ngongola-Reinke, and Tozgani Fainess Mbale. Demystifying cryptocurrency mining attacks: A semi-supervised learning approach based on digital forensics and dynamic network characteristics. In *3rd IEEE INTERNATIONAL CONFERENCE IN ICTs (ICICT 2019)*, 2019.

[175] Aaron Zimba, Zhaoshun Wang, and Mwenge Mulenga. Cryptojacking injection: A paradigm shift to cryptocurrency-based web-centric internet attacks. *Journal of Organizational Computing and Electronic Commerce*, 29(1):40–59, 2019.