

# A Longitudinal Study of Hacker Behaviour

Thomas Walshe

Department of Computer Science  
University of Oxford

firstname.secondname@cs.ox.ac.uk

Andrew Simpson

Department of Computer Science  
University of Oxford

firstname.secondname@cs.ox.ac.uk

## ABSTRACT

Bug bounty programmes employ the skills and curiosity of independent security researchers (hackers) to support pre- and post-deployment security. Driven by the question *How effective are bug bounty platforms at retaining the interest of hackers?*, this paper aims to address two issues concerning hackers' behaviour. First, to resolve the information asymmetry between programme and platform operators, it is necessary to measure the number of active hackers on a platform. Second, to assist programme operators' understanding, we identify the archetypal behaviours of hackers across a platform. We found that 6,813 hackers (with public accounts) have successfully submitted at least one vulnerability report on Bugcrowd. Of these, approximately 45% (with an account age greater than 9 months) can be considered inactive. We conclude that a significant number of inactive and unproductive hackers may contribute, in part, to the difficulties faced by programme operators. In particular, difficulties in retaining the focus of hackers can lead to underwhelming returns from the resources invested.

## CCS CONCEPTS

• **Security and privacy** → **Vulnerability management**; *Software and application security*; • **General and reference** → *Empirical studies*;

## KEYWORDS

Bug bounty programmes, vulnerability disclosure, software security, behavioural modelling

### ACM Reference Format:

Thomas Walshe and Andrew Simpson. 2022. A Longitudinal Study of Hacker Behaviour. In *Proceedings of ACM SAC Conference (SAC'22)*. ACM, New York, NY, USA, Article 4, 10 pages. [https://doi.org/xx.xxx/xxx\\_x](https://doi.org/xx.xxx/xxx_x)

## 1 INTRODUCTION

One approach to pre- and post-deployment security is realised through the operation of a bug bounty programme [37]. White-hat hackers (or hackers) compete to be the first to find, document and report a vulnerability. Vast in number and with a wide breadth of technical backgrounds, these hackers have proved to be successful at finding thousands of vulnerabilities [50].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SAC'22, April 25 – April 29, 2022, Brno, Czech Republic

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-8713-2/22/04...\$15.00

[https://doi.org/xx.xxx/xxx\\_x](https://doi.org/xx.xxx/xxx_x)

Hackers are often motivated by the lucrative rewards that can be achieved for finding critical vulnerabilities [53]. As an illustration, in the 31-day period preceding the 10<sup>th</sup> of February 2021, a total of \$411,554.0 was publicly paid out on the bug bounty platform HackerOne, with the largest single payout being \$30,000.

A majority of the 10 largest technology companies (by market capitalisation) currently operate bug bounty programmes [44]. The maximum bounty payouts offered or previously awarded are summarised in Table 1. The total monthly payouts and the high potential rewards suggest that such programmes can offer hackers a source of income and the potential to earn a great fortune.

The operation of bug bounty programmes is not limited to major technology companies: approximately 50% of public programmes listed on HackerOne are created by small and medium businesses [51]. Although these organisations do not necessarily offer incentives on a par with large organisations, they still see success in the operation of their programmes, thus demonstrating that bounty payouts are not the only motivator for hackers [51].

The operators of bug bounty programmes have previously expressed concerns over motivation and focus [1]. Previous work has also highlighted the role that high-productivity hackers play, with the top 100 hackers on the HackerOne platform discovering a majority of the reported vulnerabilities [51]. A behavioural study of participants has the potential to indicate the platform-wide motivation of hackers when searching for vulnerabilities to organisations looking to operate bug bounty programmes; it also has the potential to provide additional insight to organisations already operating such programmes. It is hoped that the approach detailed in this paper can be extended to allow for inter-platform comparisons, enabling organisations to make more informed decisions when choosing the platform that provides the largest active user-base.

## 2 BACKGROUND AND MOTIVATION

### 2.1 Background

Secure Software Development Lifecycles (SSDLs) are representative of a development paradigm that encourages security to be considered at each stage of development or over the entire lifetime of a product [30]. With an increased focus on security, the employment of an SSDL framework is intended to reduce the presence of vulnerabilities. This is typically achieved by integrating security into all development phases [41].

One example of an SSDL framework is the Building Security in Maturity Model (BSIMM), a collection of 116 curated security activities employed by 120 organisations [4]. One of these activities advocates for the use of bug bounty programmes:

“the organization solicits vulnerability reports from external researchers and pays a bounty for each verified and accepted vulnerability received” [4].

Organisation	Maximum bounty (\$)
Apple	1,000,000 [28]
Microsoft	250,000 [40]
Amazon	15,000 [22]
Alphabet	132,500 [16]
Facebook	80,000 [19]
Tencent	140,000 [24]
TSMC	NA
Samsung	200,000 [48]
NVIDIA	0
Adobe	0

**Table 1: Summary of bounty payouts by the largest technology companies by market capitalisation in 2021. Note that NVIDIA and Adobe operate responsible disclosure programmes, and, as such, do not offer public bounty payouts. TSMC does not offer any form of public vulnerability disclosure programme.**

Both the HackerOne and BugCrowd bug bounty platforms<sup>1</sup> report that approximately 79% of programmes hosted are private [5, 20]. (Christian noted that, as of 2018, the large number of private programmes inhibited research, as these organisations were reluctant to share private data [7].) These platforms are operated by a third party, serving as a centralised directory for programmes.

Previous empirical studies of bug bounty programmes and platforms have investigated the economics of programme operation. Finifter et al. [13] compared the effectiveness of the Google Chrome and Mozilla Firefox rewards programmes, finding, among other observations, that Google’s tiered reward structure was more effective at encouraging hacker participation. Zhao et al. [54] compared the managed platform HackerOne and the now defunct platform, Wooyun. The incentives provided by monetary rewards were investigated, along with the search strategy used by hackers. The present authors investigated the the benefits and costs associated with employing a bug bounty programme instead of hiring additional security researchers [51]. Together, these three studies have quantified the benefit hackers can provide to an organisation’s post-deployment security.

## 2.2 Related work

We now present a brief survey of relevant literature, giving consideration to hacker motivation, behaviour, and productivity.

**2.2.1 Hacker motivation.** There are several theories as to the primary factors that motivate hackers to search for vulnerabilities. The relationship between monetary compensation and intangible rewards is discussed consistently within the literature.

Malladi and Subramanian [37] argue that “financial compensation is the sole motivation for researchers” and that intangible rewards such as reputation are ineffective. Algarni and Malaiya [3] note that, for top hackers (i.e. those with a high number of submissions), monetary rewards are a key incentive to hackers.

Metric	HackerOne	Bugcrowd
Number of programmes	326	133
Number of users	>600,000 [23]	>200,000
Total bugs found	>150,000 [23]	>26,000 (2019)
Est. total payout	>\$80M [23]	>\$6.5M (2018)
Total funding	\$110.4M [10]	\$78.7M [8]

**Table 2: Comparison between the HackerOne and Bugcrowd platforms. Data current as of 10<sup>th</sup> February 2021, or as otherwise specified.**

Results from the HackerOne 2019 *Hacker Report* [21] reveal that 14.3% of those surveyed reported that “to make money” was their primary motivation for participation. Conversely, 56.7% of hackers stated that non-monetary motivating factors, such as the potential to learn, were their primary motivation. Approximately 28% reported altruistic reasons for hacking. However, it could be argued that this figure is over-inflated due to social-desirability biases.

The success of programmes and full-disclosure platforms without any monetary reward suggests that hackers often value public recognition and reputation above immediate financial gain. The defunct platform Wooyun was studied by Zhao et al. [53], who found that hackers found thousands of vulnerabilities in the absence of financial motivation. Prestige as a motivator may emerge from a high reputation score, featuring in a hall-of-fame, or being named in a public disclosure — all of which are used as rewards [54].

Votipka et al. [50] identified three motivators for hackers when selecting programmes and areas on which to work: potential monetary payouts, value of target asset to an organisation (finding vulnerabilities in an important asset), and non-monetary rewards (enjoyment from the task, altruism and pride). With observations of the previous payouts awarded to hackers and with an estimation of the potential monetary payouts, geographic factors and local labour conditions may significantly impact a hacker’s motivation to work on a programme. For example, it has been found that hackers from developing countries with weaker economies make up a significant proportion of those contributing to bug bounty programmes [2, 3]. This may be, in part, due to the increased salary that can be made from bounty payouts when compared to the local median annual wage [21]. In these regions, monetary rewards may become the primary motivator. In contrast, the uncertainty of rewards and relatively high median wage within a richer economy may cause hackers to favour conventional employment.

The use of a tiered payout structure has been found to encourage participation in a particular programme [13]. Organisations will offer a range of payouts for each severity category of reported vulnerability, e.g. \$1,000–\$5,000 for critical bugs. The distribution of payout structures for bug bounty programmes on HackerOne was investigated by Walshe and Simpson [51], who found that the average payout increases significantly with severity. Fryer and Simperl [15] found that BugCrowd has initial recommendations for the payout structure, dependent on the maturity of an organisation’s security process maturity, however the rationale behind the suggestions is not transparent.

One cause of rising operational costs is due to the perceived increase of product security over time [13]. Organisations may

<sup>1</sup>The two most popular bug bounty platforms; basic statistics for each are shown in Table 2.

judge their own security based on the current payout level [43]. It is suggested that increasing payouts over time motivates hackers to search beyond the initial easy bugs [36, 50].

**2.2.2 Behaviour.** Organisations have a pre-adoption fear of untrustworthy hackers participating in the search for vulnerabilities [1]. Instead of turning over the discovered bugs to the programme operator, hackers may choose to sell the bugs on the black or grey markets [18]. A study by Egelman et al. [12] compared arguments for the use zero-day markets (ZDMs) and vulnerability reward programmes (VRPs), raising the question “can prices be regulated, e.g., by VRPs or other mechanisms, in such away as to avoid the need for ZDMs?” [12].

Meakins [39] notes that grey markets can be a big source of vulnerabilities, often attracting the attention of government intelligence agencies. Governments may decide to pass on details of purchased vulnerabilities to the relevant organisation; this is decided through a Vulnerabilities Equities Process (VEP), and is dependent on the strategic value of the vulnerability in question [17]. The economic factors surrounding the decision for a hacker to sell a vulnerability to a black or grey market are among many that influence hacker behaviour.

Kesan and Hayes [32] proposed an independent third party to regulate the sale of vulnerabilities. Another solution puts further emphasis on the use of attractive bug bounty programmes, providing hackers with an easy, legitimate and legal method to monetise their work [2].

It seems somewhat inevitable that the proclivity of certain hackers to ignore white markets will continue to shape the operation of bug bounty programmes, potentially influencing the behaviour of all vulnerability hunters.

**2.2.3 Productivity.** Increasing the number of participants, rather than investing resources to maintain current high-productivity hackers, is often recommended as the most effective way to increase the rate of bug discovery [36, 54]. Maintaining high-productivity hackers is a challenge faced by current programmes [1]. It is recommended that organisations endeavour to have fast and effective communication between security teams and hackers, thus helping to maintain their good standing with participating hackers [37, 50].

Other suggestions to improve productivity and bolster motivation include: having experts liaise with hackers, ‘game-ifying’ the searching and report process, and decomposing search domains into small task areas [11, 34]. So-called ‘gamification’ of platforms currently exists in the form of leaderboards, badges, and reward points; however, it has been recognised that the extent to which such techniques are employed is limited [42].

## 2.3 Motivation for the present study

With limited security budgets, it is difficult to allocate resources in such a way to cover all aspects of security. This is complicated as not all organisations conduct detailed cost–benefit analyses for security investments [47], thus making it increasingly difficult to choose appropriate security investments. These difficulties can be attributed, in part, to organisations possessing incomplete information.

Bug bounty programmes can offer organisations quantifiable returns (e.g. \$ per vulnerability found) [51]. However, when considering a new programme, a resource-allocation problem arises: How should one choose between competing platforms such that the highest rate of vulnerability discovery is achieved for the lowest cost? To explore this question, we consider a manager that is looking either to launch a new programme (hosted on a platform) or to modify an existing programme. With many platforms available for hosting, a manager must consider how to allocate resources to utilise the products and services of one or more platforms.

Raymond suggested that it is beneficial to have as many individuals searching for vulnerabilities as possible [46]. Arising from this is a clear network externality that impacts the utility an organisation can gain from the use of a particular bug bounty platform: platforms should become more attractive as their hacker-base grows. However, although platforms claim to harbour hundreds-of-thousands of hackers (recall Table 2), organisations still worry about the inactivity (and low motivation) exhibited by hackers [1]. As platforms (understandably) do not publish accurate figures representing the number of hackers actively participating at a particular point in time, there exists a significant degree of uncertainty with regards to the expected utility gained when using a specific platform [31].

It follows that further investigation into the true number of active hackers on a platform, as well as their behaviour, is needed to address the information asymmetry that exists between programme managers and bug bounty platforms. Resolving this asymmetry has the potential to aid managers in maximising the benefits an organisation can receive through the operation of a bug bounty programme.

To tackle our research question *How effective are bug bounty platforms at retaining the interest of hackers?* we consider the following set of subsidiary questions: (1) *How many hackers are actively engaged in the search for vulnerabilities?*; (2) *What are the archetypal behaviours of hackers?*; and (3) *Is participation in bug bounty programmes currently viable for most hackers?*

Question 1 aims to quantify the number of active and inactive hackers (taking Bugcrowd as a representative example), to provide an insight into the long-term motivation of hackers as well as a more accurate representation of the size of the network. The ambition of Question 2 is to help identify characteristic behavioural patterns that are common amongst subsets of the user-base. Additionally, we explore the behaviour of the top hackers found on Bugcrowd, who have been shown to contribute far more than other bug hunters [51]. Subsequently, we aim to categorise hackers into productivity groups and provide qualitative descriptions of the predominant working patterns observed in the data. Question 3 seeks to combine the insights and results from the previous questions to comment on the viability of participation in the search for vulnerabilities. Commonly found unproductive and inconsistent behaviour may indicate that the current system is unsustainable for many hackers, who in turn choose not to continue to participate.

## 3 DATA AND METHODOLOGY

### 3.1 Data collection

The bug bounty platform Bugcrowd serves as the source of data for this study, as it allows for the output of hackers to be studied

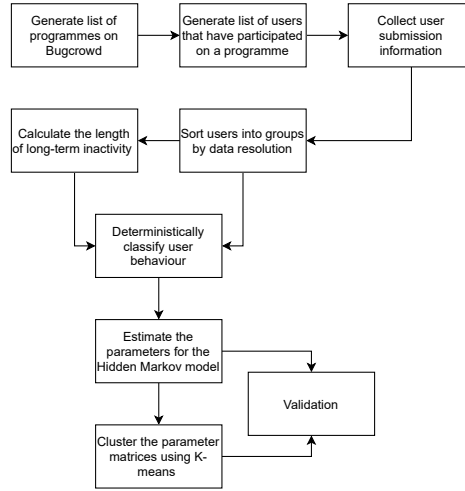


Figure 1: Conceptual diagram of the data pipeline.

over time through time-series data presented on each user profile. Furthermore, as one of the major platforms, it attracts significant interest from large numbers of hackers and organisations.

A conceptual diagram of the utilised data collection, modelling and validation processes is shown in Figure 1. The remainder of this section will detail the data collection stages.

As of the 17<sup>th</sup> of January 2021, Bugcrowd was host to 129 public bug bounty programmes that are open to participation from any individual (private programmes require a hacker to be invited before they are allowed to participate). Upon the submission of a valid vulnerability report, a hacker will be listed within the “Hall of Fame” for the corresponding programme. As such, the “Hall of Fame” for each programme contains a list of all hackers that have previously participated. (Some hackers may choose to remain anonymous, in which case “Private User” is displayed instead of a username.)

Using all available public programmes on Bugcrowd, a list of 6927 unique usernames was collected, with each corresponding to a hacker who has submitted at least one valid vulnerability report in the past.

Following the collection of all relevant user-names, time-series submission data was scraped from each user’s profile page (114 profiles were unavailable or empty).

Bugcrowd uses binning to group occurrences of vulnerability reports into fixed intervals of time (e.g., number of submission made within a month). The granularity of the data (temporal resolution of the time-series) is determined by the age of a particular account. As accounts become older, the temporal resolution decreases. For example, an account created 63 days ago will have the submission data displayed by day, but an account created 77 days ago will use weekly resolution. We refer to the “timebase” as being the temporal resolution of a given time-series, and the “intervals” as the number of data points. An account with weekly data displayed on the time-series is described as being in the “week” timebase category.

Table 3 shows the user data categorised by timebase, and summarises the minimum and maximum age (number of intervals) of

Timebase	Users in group	Age (Intervals)		
		Min	Avg.	Max
Day	74	6	35	63
Week	958	11	26	36
Month	2220	9	15	22
Quarter	2135	8	12	17
Year	1426	5	7	9

Table 3: Hacker profiles grouped by timebase.

user accounts. Data was collected from accounts ranging from 6 days to 9 years old.

### 3.2 Activity classification

To study the behaviour of hackers, it is necessary to classify a user’s activity over time. A conceptual model of a hacker’s behaviour is created from first principles, and contains three primary activities:

- *Submitting*: A hacker has submitted at least one report in a given interval of time.
- *Searching*: A theorised period of time in which a hacker is actively searching for vulnerabilities. A survey of hackers conducted by Hata et al. found that the average hacker takes up to a week in the search for a vulnerability [27]. As such, we define a fixed period of 7 days before the submission of a vulnerability report in which we believe the hacker is actively searching for vulnerabilities.
- *Inactive*: There are extended periods of time in which some hackers seem neither to be searching nor submitting. We further decompose inactive behaviour into three sub-types:
  - (1) Initial account inactivity: The period of time after account creation and before searching for, or submitting, the first report.
  - (2) Inter-report inactivity: The measurable period of time between submissions that is not classified as searching.

- (3) Long-term inactivity: A state in which a hacker is no longer participating in the search for vulnerabilities.

For all user time-series, we deterministically label each interval of time with one of these activities.

### 3.3 Classifying long-term inactivity

As stated in Section 3.2, one of the inactive sub-types relates to the state in which a hacker is no longer participating on the platform. Furthermore, it is assumed that they will not return. To determine the number active hackers on a platform, it is necessary to know the probability of a user submitting a report in the future given that they are currently inactive.

From the time-series of all hackers that have submitted reports on at least two separate occasions, the lengths of inter-report inactivity between adjacent submissions are recorded. From this, distributions of the inter-report inactive durations are created for users within each timebase group (e.g. year, quarter, etc.). The distributions from higher resolution timebases are used as prior distributions to re-sample data in one timebase to a shorter one (e.g. year to quarter). Where a prior distribution can't be generated from existing data, a uniform prior is used. This allows for all inter-report intervals to be re-sampled into the day timebase.

The resulting distribution represents the probability of a report being submitted on or before day  $n$ , given  $n - 1$  days of inactivity. Through Monte Carlo simulation we find that, after submitting a report, there is a 95% probability of the next report being submitted within 289 days [285,293] (95% CI). This allows us to define a length of continuous inactivity greater than 289 days to be classified as long-term inactive, with a Type 1 error of 5%. If a hacker remains inactive for over 289 days following their last submission, the period is reclassified as 'long-term inactive', with the hacker labelled no longer active. In order to make an estimate of the number of hackers that have left the platform, it is necessary to assume that a period of inactivity greater than 289 days is due to a hacker being permanently inactive.

## 4 MODELLING HACKER BEHAVIOUR

### 4.1 Model assumptions

There are several important assumptions made about the behaviour of hackers. First, hackers that enter a state of long-term inactivity will not participate again in the future. As stated in Section 3.3, we take a time limit that is accurate for 95% of hackers. This implies that the long-term inactive state in the model is absorbing (once entered it cannot be left). Second, we do not attempt to model the number of reports submitted in one interval. Instead, we are concerned only if a hacker has submitted any report in a given interval. Third, we assume that all parameters governing the behaviour of hackers are temporally homogenous. The limited resolution of the data results in time-series with few points. It is, therefore, not appropriate to explore temporal heterogeneity within the parameters.

The model adheres to the two Markov assumptions [45]: the probability of moving to a particular state is dependent only on the current state (first-order Markov chain) and the transition probabilities exhibit temporal homogeneity (stationary process).

Timebase	Initial state			Final state		
	0	1	3	2	3	4
Day	7	17	50	69	5	0
Week	281	136	541	898	60	0
Month	709	312	1199	1308	272	640
Quarter	1054	0	1081	638	440	1057
Year	554	0	872	0	513	913

Table 4: Distribution of initial and final states.

### 4.2 Hidden Markov Model

Previous work has highlighted the success of Hidden Markov Models when applied to modelling sequential data [26, 33], especially when used for state-based behavioural models [29].

The activities outlined in Section 3.2 are used as the hidden states in the behavioural model, as shown in the model topology in Figure 2. Note that 'Initial account inactivity' is always a transient state (once left, it is not returned to), and 'Long-term inactivity' is always an absorbing state (once entered, it cannot be left).

If a given hacker has a non-zero transition probability from the 'Submitting' state to the 'Long-term inactive' state, the model becomes an absorbing Markov chain (an absorbing state can always be reached), as all hackers reach the 'Submitting' state at least once. A distribution of the initial and final states is shown in Table 4.

The emissions from the model are limited to binary values, indicating if the hacker has submitted reports in a given time interval. Other than the 'Submitting' state, all hidden states emit a zero, representing the absence of new submitted reports.

For model creation and testing, 70% of the hacker time-series dataset is randomly select, with the remaining 30% used as an unseen validation dataset.

### 4.3 Generating the transition matrix

Within each group, the  $5 \times 5$  state transition matrix is computed for each user and the initial state is recorded. Initially the model is assumed to be fully connected. As both the hidden state sequence and the observation sequence are known (due to the data being fully-labelled), the first-order transition probabilities can be calculated using [49]:

$$p_{ij} = \frac{n_{ij}}{\sum_{j=1}^k n_{ij}} \quad (1)$$

Here,  $n_{ij}$  represents the number of transitions from state  $i$  to  $j$  and  $k$  represents the total number of states.

### 4.4 Clustering hacker behaviour

To group hackers with similar behavioural patterns, unsupervised clustering was performed within each group. First, simple dimensional reduction was performed by removing any unused transition probabilities ( $p_{ij} = 0$ ) within a group of hackers (e.g.  $p_{44}$  is unused in the day timebase). The remaining states were transformed into a feature vector for each hacker.

A K-means algorithm was used to form clusters using the feature vectors. The Elbow and Silhouette methods were both used to select the appropriate number of clusters within each group [38, 52]; the chosen number of clusters for each group are shown in Table 5.

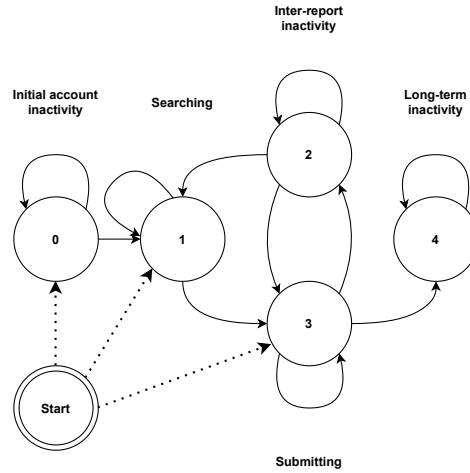


Figure 2: Topology of the Hidden Markov Model, showing commonly used transition pathways.

Timebase	$K$
Day	23
Week	50
Month	100
Quarter	100
Year	52

Table 5: Number of clusters chosen for each group.

Timebase	Individual Error (%)	Cluster Error (%)
Day	-10.3	-11.3
Week	-12.0	-7.5
Month	-17.8	-21.9
Quarter	-7.2	-7.3
Year	-13.7	-14.1

Table 6: Expected rate of submission error, using transition matrices for individual users and those generated from K-means cluster centres.

## 4.5 Model validation and replication

We now consider results from the model validation process.

**4.5.1 Individual.** The model’s ability to replicate individual user data was tested. For each hacker in the aforementioned timebase groups, the rate of submission and standard deviation were calculated using the collected time-series data, as shown in Table 6.

A time-series was generated using the transition matrix calculated from the labelled data for each hacker. Monte Carlo simulation was used to generate 10,000 synthetic time-series for each hacker. The observed values for the rate of submission and standard deviation were compared to the expected values generated from the simulation, and the mean absolute errors (MAE) were recorded. The time spent in each state was recorded, and MAE across all users was calculated (Table 7).

As per Table 6, using individual model parameters to generate synthetic time-series leads to an underestimation of the number of emissions when compared to the hacker data. Hackers in the year group exhibit the largest error, possibly due to the limited sizes of data available.

Table 7 shows the mean absolute error for the time spent in each state. With the exception of state 2 (Inactive) in the day group, all errors are less than 1 time interval.

**4.5.2 Clusters.** The cluster centroids generated by the clustering algorithm were used as transition matrices, and the ability to generalise user behaviour was similarly tested. The final number of

clusters is shown in Table 5. The submission rate error is shown in Table 6; the time-in-state error is shown in Table 8.

Transforming the centroids for use as transition matrices leads to an underestimate in the number of emissions, with the month group having the largest error of -21.9%. All states see a small increase in the time-in-state MAE when compared with the individual time-in-state error.

**4.5.3 Analysis.** The results demonstrate that the approach to modelling hacker behaviour outlined in Sections 3 and 4 is effective.

Clustering the individual transition matrices and using the centroids to represent the behaviour of similar groups of hackers leads to a small decrease in performance. This is, in part, due to the cluster centres being an imperfect representation of each parameter matrix within the cluster as there are variances in intra-group hacker behaviour. This is shown by an increase in the (absolute) expected rate of submission error in most timebases, and an increase in the MAE for the time spent in each state (for all states and timebases). Performance may be improved by further optimising the number of clusters used in each timebase.

Aside from the ability of the centroids to capture behavioural characteristics of hackers, they also serve as a compact representation of the behaviour of hackers on Bugcrowd.

Timebase	Time-in-state MAE (intervals)				
	0	1	2	3	4
Day	0.73	1.95	2.45	0.64	NA
Week	0.91	0.47	2.0	1.18	NA
Month	0.73	0.36	1.34	1.08	0.30
Quarter	0.85	NA	0.87	1.04	0.75
Year	0.26	NA	0.25	0.71	0.60

**Table 7: Mean absolute error calculated for the time in each state, using transition matrices for individual users.**

**4.5.4 Replication.** To allow for replication of the study, the data pipeline and modelling codebase and documentation is available on a public GitHub repository.<sup>2</sup>

## 5 RESULTS

Results pertaining to the questions introduced in Section 2.3 are presented in this section. We consider each question in turn.

### 5.1 How many hackers are active?

We define an active hacker as someone whose behaviour is not classified as long-term inactive. (As stated in Section 3.3, any individual with a trailing period of inactivity longer than 289 days is defined as long-term inactive.)

In the model, long-term inactive hackers are represented as finishing in state 4 (long-term inactivity). Table 4 shows the final states of all hackers within each timebase. It was found that 2610 hackers (37.7%) are estimated to no longer be active; for hackers with accounts older than 9 months old the inactivity rate is 44.3%. A larger proportion of hackers in the Day (93.2%) and Week (93.7%) timebase were found to be in an inactive final state. However, due to the age of the accounts (all less than 289 days), it was not possible to classify any of these accounts as long-term inactive. It is therefore assumed that the true number of inactive hackers is higher than 37.7% due to inactive hackers within these groups with accounts too new to be classified accurately as such.

### 5.2 What are the archetypal behaviours of hackers?

We consider the archetypal behaviours of top hackers in Section 5.2.1 and of all hackers in Section 5.2.2.

**5.2.1 What are the archetypal behaviours of top hackers?** Previous work has highlighted that a small proportion of the overall user-base contributes a significant proportion of the submitted vulnerabilities; the top 100 hackers on HackerOne are responsible for over 50% of all vulnerability discoveries [51]. As such, the work habits of the top 100 Bugcrowd hackers were studied. To be in the top 100, a hacker must discover and report more than 193 vulnerabilities. Hackers meeting this criterion are found in the month (3), quarter (28), and year (69) groups.

Using the inactivity limit of 289 days, the number of long-term inactive top 100 hackers is estimated to be 19 (1 in the quarter group, and 18 in the year group). Within each group, the proportion of

Timebase	Time-in-state MAE (intervals)				
	0	1	2	3	4
Day	0.72	1.97	2.47	0.65	NA
Week	0.98	0.67	2.45	1.44	NA
Month	0.78	0.41	1.47	1.16	0.31
Quarter	0.90	NA	0.94	1.12	0.77
Year	0.27	NA	0.25	0.73	0.61

**Table 8: Mean absolute error calculated for the time in each state, using transition matrices generated from K-means cluster centres.**

hackers classified as long-term inactive is far less than the overall proportion of long-term inactive hackers. As detailed in Section 5.1, the estimated number of long-term inactive hackers within the groups is: 28.8% in months (0.0% in top 100), 49.5% in quarters (3.6% in top 100), and 64.0% for year (26.1% in top 100).

As defined in Section 3.2, initial account inactivity is the period of time before the first interval containing submissions. Table 4 shows the number of hackers that start in an inactive state. For the top 100 hackers, 14 in the month group and 15 in the year group start as inactive. The proportion of top 100 hackers that start as inactive is comparable or lower than the group average.

Analysing the distribution of parameters within the individual transition matrices allows for further investigation into the work habits of the top hackers. The transition parameter  $p_{33}$  represents the probability of submitting vulnerability reports in the next available interval, given that reports were submitted in the current interval. High values of the parameter ( $p_{33} \in [0, 1]$ ) indicate that a hacker is consistently submitting reports in each observable interval of time. Lower values represent more sporadic behaviour. As shown in Table 9, the top 100 have significantly higher average  $p_{33}$  values than other hackers within the same group.

The parameter  $p_{22}$  represents the probability of the next interval being inactive, given that the current interval is inactive. A value close to 0 represents an absence of multiple intervals of inactivity. Table 9 shows that top hackers rarely spend more than one consecutive interval in an inactive state.

To quantify consistent behaviour during the period of time in which a hacker is active, the following harmonic mean is defined:

$$\alpha = \frac{2}{\frac{1}{1-p_{22}} + \frac{1}{p_{33}}} \quad (2)$$

Here,  $\alpha$  is the harmonic mean ( $\alpha \in (0, 1]$ ).

Top hackers across all groups have an  $\bar{\alpha} = 0.923$ , far greater than the figure for all hackers ( $\bar{\alpha} = 0.459$ ). This indicates that top 100 hackers tend to discover and submit vulnerabilities far more consistently than other hackers on Bugcrowd. This also suggests that barriers, other than skill, may limit the majority of hackers in the search for vulnerabilities in a consistent manner, and that only a small proportion of hackers are able to search on a full-time basis.

**5.2.2 What are the archetypal behaviours of hackers?** Clustering hackers allows for the identification of similar behavioural patterns (this process is detailed in Section 4.4).

For each cluster a value of  $\alpha$  was calculated using (2), allowing the clusters to be divided equally between four productivity groups:

<sup>2</sup>[https://github.com/paper-hacker-behaviour/code\\_base](https://github.com/paper-hacker-behaviour/code_base)

Timebase	top 100			All		
	$\bar{p}_{00}$	$\bar{p}_{22}$	$\bar{p}_{33}$	$\bar{p}_{00}$	$\bar{p}_{22}$	$\bar{p}_{33}$
Day	NA	NA	NA	0.23	0.82	0.22
Week	NA	NA	NA	0.34	0.75	0.40
Month	0.00	0.00	0.98	0.29	0.48	0.39
Quarter	0.17	0.05	0.94	0.32	0.36	0.39
Year	0.05	0.02	0.87	0.19	0.15	0.39

**Table 9: Mean parameter values of  $p_{00}$ ,  $p_{22}$ , and  $p_{33}$  for all hackers and top 100 hackers.**

very low, low, medium, and high. Qualitative descriptions of hacker working patterns for each group are given below.

- Very low: Over the lifetime of an account, hackers tend to submit reports in one or two intervals, before going into an inactive state. It is common to see activity only in the first few intervals after account creation.
- Low: Hackers will submit sporadically over time, often with long gaps of inactivity between submissions.
- Medium: Groups of sequential submissions are observable, indicating longer periods of hacker activity.
- High: After a possible period of inactivity following account creation, hackers will submit reports in almost all intervals. All of the active top 100 hackers identified in Section 5.2 are part of this productivity group.

### 5.3 Is participation viable for most hackers?

Of the 6813 hackers identified within the study, it is predicted that 2610 are no longer active. The proportion of hackers entering a state of long-term inactivity (Table 4) increases with account age: 29% of accounts with an age of 9–22 months are inactive, rising to 50% of account between 8–17 quarters, and 64% for older accounts. The behavioural analysis of Section 5.2.2 reveals that large numbers of hackers can be characterised by their unproductive behaviour. Especially notable is the subset of hackers that only submit one or two reports after account creation and thereafter end their participation in the search for vulnerabilities.

## 6 ANALYSIS

### 6.1 How many hackers are active?

From a user-base of over 200,000 hackers, it is only possible to identify 6813 unique individuals that have (publicly) participated in at least one public bug bounty programme. A rough estimate of the number of unique anonymous hackers (listed as “Private user” on a Hall of Fame page) can be made using the average proportion of anonymous hackers (relative to public hackers) across all Hall of Fame lists. This leads to an estimate of 3717 unique hackers that have participated and chosen to remain anonymous (assuming both groups have the same distribution of the number of unique programmes a hacker has participated in). As such, only 10,530 ( $\approx 5.27\%$ ) of accounts created on Bugcrowd are predicted to have participated in at least one bug bounty programme.

Of the 6813 hackers identified within this study, it is estimated that 2610 (37.7%) are no longer active participants. As noted by Al-Banna et al., one issue faced by programme operators is the difficulty

in maintaining participants (within a specific programme) [1] — not least because common or obvious vulnerabilities are quickly found [35]. The high levels of hacker inactivity suggest that bug bounty platforms are also unable to sustain hacker motivation, even as the number of new programmes (and potential vulnerabilities) continues to grow [51].

We would argue that both programme and platform operators should endeavour to explore the impact of financial incentives (by, for example, adjusting bounty values) and ways to improve accessibility, to encourage hackers to continue to search for vulnerabilities. As previously alluded to, gamification of tasks is one such method that has been suggested as a means to increase and sustain motivation in software engineering and crowdsourced activities [34].

The operators of bug bounty programmes may see increasing benefits as the number of active hackers on a platform increases [46]. Calculating the magnitude of this positive network externality requires knowledge of the number of hackers searching for vulnerabilities — a figure not currently published by the operators of bug bounty platforms. It is hoped that the results of the study will help address this existing network externality, and that the method may be extended to platforms other than Bugcrowd.

### 6.2 What are the archetypal behaviours of hackers?

From Bugcrowd, the top 100 hackers (by number of submissions) were identified as those with over 193 discovered vulnerabilities. Within this group far fewer hackers (19%) are classified as long-term inactive (as compared to the all hackers), suggesting that there is continual motivation to search for vulnerabilities.

It has previously been found that use of leaderboards, achievement points, and badges of recognition motivate some students to work harder and undertake increasingly challenging tasks [14]. The ambition to retain a top leaderboard ranking may serve as an additional source of motivation for these top hackers. Highly consistent submission patterns are a characteristic feature within the group of top hackers ( $\bar{\alpha} = 0.923$ ), and all top 100 hackers are found within the high-productivity group (as defined in Section 5.2.2).

A typical (and commonly found) working pattern of hackers in the very low productivity group is submissions being made in one or two intervals following account creation, followed by inactivity. Programme operators have expressed concerns over the low quality of submitted reports and the large amount of time and resources required to process them [1]. The presence of large numbers of these ‘one-off’ hackers, in conjunction with their unfamiliarity with the discovery and reporting process, may result in dissatisfaction from both parties. Further investigation is needed to understand why this type of behaviour is prevalent among hackers.

### 6.3 Is participation viable for most hackers?

Over time, a greater proportion of hackers become inactive and no longer search for vulnerabilities. This suggests that, in their current configuration, long-term participation in bug bounty programmes is not viable for most hackers. Furthermore, a low proportion ( $\approx 5.27\%$ ) of accounts on Bugcrowd have participated, suggesting that more needs to be done to encourage new users to find their first vulnerability. Encouragement for new users could come in the



form of increased gamification of the discovery process, or continued investment in the development and publication of teaching resources. Free online resources such as Bugcrowd University [6] and HackerOne's Hacker101 [25] (acquired from Breaker101 in 2018 [9]) provide individuals with video guides and interactive challenges (e.g. capture the flag) with the aim to improve their understanding of the vulnerability discovery process.

## 6.4 Limitations

It is important that we note some limitations of the study.

First, at the time of writing, only Bugcrowd allows for the contributions of hackers over time to be collected and analysed. Acquiring user data from only one platform (albeit a major platform) may lead to poor generalisation of results when the methodology is applied to other platforms.

Second, data could only be collected from individuals with public accounts and that have previously participated in a public programme. For a given Hall of Fame page, on average approximately 35% of accounts listed are anonymous, preventing the collection of user data. The inability to collection information about all hackers on Bugcrowd may lead to results being unrepresentative of hacker behaviour, particularly within private programmes.

Third, the granularity (temporal resolution) of the data collected limits the accuracy of the calculated model parameters. Access to higher resolution data (e.g. daily data for all hackers) would allow for the temporal heterogeneity of the parameters to be investigated, giving rise to greater in-depth behavioural analysis.

## 7 CONCLUSION

We have described the use of a Hidden Markov Model to model and analyse the behaviour of 6,813 hackers that have participated in the search for vulnerabilities in bug bounty programmes on Bugcrowd. Additional behavioural analysis has enabled a prediction of the proportion of active users to be made: it is estimated that approximately 2610 hackers are no longer active in the search for vulnerabilities and that 19% of the top 100 (at the time of writing) hackers on Bugcrowd are inactive, potentially raising concerns regarding the long-term motivation of hackers when participating in bug bounty programmes.

Providing an otherwise inaccessible figure of the number of active hackers to managers currently operating, or looking to operate, a bug bounty programme has the potential to address the information asymmetry that exists between organisations and platforms. With limited resources to spend on operating a bug bounty programme, it may be advantageous for operators to select the platform with the most active user-base.

Clustering using K-means has proven effective in grouping together hackers with similar working patterns, allowing archetypal behaviours to be identified. Further analysis of the groups has yielded qualitative descriptions of working patterns at different levels of productivity.

The increasing number of inactive hackers over time, together with the small proportion of accounts that have submitted at least one vulnerability report, suggests that participation is not viable for a significant number of individuals. Consequently, the operators of programmes and platforms may wish to further consider how

best to motivate both new and experienced hackers to continue searching for vulnerabilities via consideration of questions such as *Do hackers feel fairly treated by the current incentive mechanisms?* and *What measures can programme operators take to encourage continued participation?* Qualitative surveys and interviews with current and former hackers may allow for a great understanding of the effectiveness of incentive mechanisms and motivating factors. In addition, further research is needed to investigate the network externalities that arise on bug bounty platforms. Specifically, how can the size of the network effect be measured as a function of the number of active hackers and the number of programmes? Research answering *Does increasing the number of programmes on a platform produce positive or negative network externalities?* would produce valuable insights.

## ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their helpful comments.

The research described in this paper was undertaken as part of the “Data and models for secure software engineering” project, which is funded by the UK’s National Cyber Security Centre.

## REFERENCES

- [1] Mortada Al-Banna, Boualem Benatallah, Daniel Schlagwein, Moshe C Barukh, and Elisa Bertino. 2018. Friendly Hackers to the Rescue: How Organizations Perceive Crowdsourced Vulnerability Discovery. In *Pacific Asia Conference on Information Systems (PACIS)*.
- [2] Abdullah Algarni and Yashwant Malaiya. 2014. Software vulnerability markets: Discoverers and buyers. *International Journal of Computer, Information Science and Engineering* 8, 3 (2014), 71–81.
- [3] Abdullah M Algarni and Yashwant K Malaiya. 2013. Most successful vulnerability discoverers: Motivation and methods. In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer ..., 1.
- [4] BSIMM. 2018. BSIMM9: Building Security in Maturity Model version 9. Retrieved December 18, 2019 from <https://www.bsimm.com/download/>
- [5] Bugcrowd. 2018. Bugcrowd: State of bug bounties. Retrieved July 15, 2019 from <https://www.bugcrowd.com/resources/reports/state-of-bug-bounty-2018/>
- [6] Bugcrowd. 2021. Bugcrowd University. Retrieved May 10, 2021 from <https://www.bugcrowd.com/hackers/bugcrowd-university/>
- [7] Joseph L Christian. 2018. *Bug Bounty Programs: Analyzing the Future of Vulnerability Research*. Ph.D. Dissertation. Utica College.
- [8] Crunchbase. 2021. Bugcrowd Financials. Retrieved February 10, 2021 from [https://www.crunchbase.com/organization/bugcrowd/company\\_financials](https://www.crunchbase.com/organization/bugcrowd/company_financials)
- [9] Crunchbase. 2021. HackerOne company financials. Retrieved May 10, 2021 from [https://www.crunchbase.com/organization/hackerone/company\\_financials](https://www.crunchbase.com/organization/hackerone/company_financials)
- [10] Crunchbase. 2021. HackerOne Financials. Retrieved February 10, 2021 from [https://www.crunchbase.com/organization/hackerone/company\\_financials](https://www.crunchbase.com/organization/hackerone/company_financials)
- [11] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*. ACM, 1013–1022.
- [12] Serge Egelman, Cormac Herley, and Paul C Van Oorschot. 2013. Markets for zero-day exploits: Ethics and implications. In *Proceedings of the 2013 New Security Paradigms Workshop*. ACM, 41–46.
- [13] Matthew Finifter, Devdatta Akhawe, and David Wagner. 2013. An Empirical Study of Vulnerability Rewards Programs. In *USENIX Security Symposium*. 273–288.
- [14] Panagiotis Fotaris, Theodoros Mastoras, Richard Leinfellner, and Yasmine Rosunally. 2016. Climbing up the Leaderboard: An Empirical Study of Applying Gamification Techniques to a Computer Programming Class. *Electronic Journal of e-learning* 14, 2 (2016), 94–110.
- [15] Huw Fryer and Elena Simperl. 2017. Web science challenges in researching bug bounties. In *Proceedings of the 2017 ACM on Web Science Conference*. ACM, 273–277.
- [16] Google. 2021. Vulnerability Reward Program: 2020 Year in Review. Retrieved February 10, 2021 from <https://security.googleblog.com/2021/02/vulnerability-reward-program-2020-year.html>
- [17] US Government. 2017. United States Government: Vulnerabilities Equities Policy and Process for the United States Government. Retrieved September 27, 2019

- from <https://www.whitehouse.gov/sites/whitehouse.gov/files/images/External%20-%20Unclassified%20VEP%20Charter%20FINAL.PDF>
- [18] Audrey Guinchard. 2017. The Computer Misuse Act 1990 to Support Vulnerability Research? Proposal for a Defence for Hacking as a Strategy in the Fight against Cybercrime. (2017).
  - [19] Dan Gurfinkel. 2020. Facebook: Marking the 10th Anniversary of Our Bug Bounty Program. Retrieved February 10, 2021 from <https://about.fb.com/news/2020/11/bug-bounty-program-10th-anniversary/>
  - [20] HackerOne. 2019. HackerOne: 188 fascinating facts. Retrieved February 25, 2019 from <https://www.hackerone.com/blog/118-Fascinating-Facts-HackerOnes-Hacker-Powered-Security-Report-2018>
  - [21] HackerOne. 2019. HackerOne: The 2019 hacker report. Retrieved September 23, 2019 from [https://www.hackerone.com/sites/default/files/2019-02/the-2019-hacker-report\\_3.pdf](https://www.hackerone.com/sites/default/files/2019-02/the-2019-hacker-report_3.pdf)
  - [22] HackerOne. 2020. Amazon Vulnerability Research Program. Retrieved February 10, 2021 from <https://hackerone.com/amazonvrp?type=team>
  - [23] HackerOne. 2020. HackerOne: The 2020 hacker report. Retrieved February 10, 2021 from <https://www.hackerone.com/resources/reporting/the-2020-hacker-report>
  - [24] HackerOne. 2020. Tencent. Retrieved February 10, 2021 from <https://hackerone.com/tencent?type=team>
  - [25] HackerOne. 2021. Hacker101. Retrieved May 10, 2021 from <https://www.hackerone.com/for-hackers/hacker-101>
  - [26] Md Rafil Hassan and Baikunth Nath. 2005. Stock market forecasting using hidden Markov model: a new approach. In *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*. IEEE, 192–196.
  - [27] Hideaki Hata, Mingyu Guo, and M Ali Babar. 2017. Understanding the heterogeneity of contributors in bug bounty programs. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. IEEE Press, 223–228.
  - [28] Alex Hern. 2019. The Guardian: Apple to pay hackers more than \$1m to find security flaws. Retrieved September 10, 2019 from <https://www.theguardian.com/technology/2019/aug/12/apple-hackers-black-hat-conference>
  - [29] XA Hoang and Jiankun Hu. 2004. An efficient hidden Markov model training scheme for anomaly intrusion detection of server applications based on system calls. In *Proceedings. 2004 12th IEEE International Conference on Networks (ICON 2004)(IEEE Cat. No. 04EX955)*, Vol. 2. IEEE, 470–474.
  - [30] Russell L Jones and Abhinav Rastogi. 2004. Secure coding: building security into the software development life cycle. *Information Systems Security* 13, 5 (2004), 29–39.
  - [31] Michael L Katz and Carl Shapiro. 1985. Network externalities, competition, and compatibility. *The American economic review* 75, 3 (1985), 424–440.
  - [32] Jay P Kesan and Carol M Hayes. 2016. Bugs in the Market: Creating a Legitimate, Transparent, and Vendor-Focused Market for Software Vulnerabilities. *Ariz. L. Rev.* 58 (2016), 753.
  - [33] Tetsunori Kobayashi and Satoshi Haruyama. 1997. Partly-hidden Markov model and its application to gesture recognition. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4. IEEE, 3081–3084.
  - [34] Markus Krause and René Kizilcec. 2015. To play or not to play: Interactions between response quality and task complexity in games and paid crowdsourcing. In *Third AAAI Conference on Human Computation and Crowdsourcing*.
  - [35] John Leyden. 2016. The Register: Fatigue fears over bug bounty programs. Retrieved August 27, 2020 from [https://www.theregister.com/2016/11/09/bug\\_bounty\\_fatigue\\_fears/](https://www.theregister.com/2016/11/09/bug_bounty_fatigue_fears/)
  - [36] Thomas Maillart, Mingyi Zhao, Jens Grossklags, and John Chuang. 2017. Given enough eyeballs, all bugs are shallow? Revisiting Eric Raymond with bug bounty programs. *Journal of Cybersecurity* 3, 2 (2017), 81–90.
  - [37] Suresh S Malladi and Hemang C Subramanian. 2019. Bug Bounty Programs for Cybersecurity: Practices, Issues, and Recommendations. *IEEE Software* (2019).
  - [38] Dhendra Marutho, Sunarna Hendra Handaka, Ekaprana Wijaya, et al. 2018. The determination of cluster number at k-mean using elbow method and purity evaluation on headline news. In *2018 International Seminar on Application for Technology of Information and Communication*. IEEE, 533–538.
  - [39] Joss Meakins. 2019. A zero-sum game: the zero-day market in 2018. *Journal of Cyber Policy* 4, 1 (2019), 60–71.
  - [40] Microsoft. 2021. Bug Bounty Program. Retrieved February 10, 2021 from <https://www.microsoft.com/en-us/msrc/bounty>
  - [41] Ernest Mougoue. 2016. Synopsys: Secure SDL 101. Retrieved July 21, 2019 from <https://www.synopsys.com/blogs/software-security/secure-sdlc/>
  - [42] Jamie O'Hare and Lynsay A Shepherd. 2020. Proposal of a Novel Bug Bounty Implementation Using Gamification. *arXiv preprint arXiv:2009.10158* (2020).
  - [43] Andy Ozment. 2004. Bug auctions: Vulnerability markets reconsidered. In *Third Workshop on the Economics of Information Security*. 19–26.
  - [44] Melissa Pistilli. 2020. 10 Top Technology Stocks by Market Cap. Retrieved February 10, 2021 from <https://investingnews.com/daily/tech-investing/top-technology-stocks/>
  - [45] Daniel Ramage. 2007. Hidden Markov models fundamentals. *CS229 Section Notes* 1 (2007).
  - [46] Eric Raymond. 1999. The cathedral and the bazaar. *Knowledge, Technology & Policy* 12, 3 (1999), 23–49.
  - [47] Brent R Rowe and Michael P Gallahe. 2006. Private sector cyber security investment strategies: An empirical analysis. In *The fifth workshop on the economics of information security (WEIS06)*.
  - [48] Samsung. 2017. Samsung Group: Get up to \$200,000 by reporting vulnerabilities. Retrieved September 10, 2019 from <https://seap.samsung.com/content/samsung-bug-bounty-get-200000-reporting-vulnerabilities>
  - [49] Anna Sperotto, Ramin Sadre, Pieter-Tjerk de Boer, and Aiko Pras. 2009. Hidden Markov Model modeling of SSH brute-force attacks. In *International Workshop on Distributed Systems: Operations and Management*. Springer, 164–176.
  - [50] Daniel Votipka, Rock Stevens, Elissa Redmiles, Jeremy Hu, and Michelle Mazurek. 2018. Hackers vs. testers: A comparison of software vulnerability discovery processes. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 374–391.
  - [51] Thomas Walshe and Andrew Simpson. 2020. An Empirical Study of Bug Bounty Programs. In *2020 IEEE 2nd International Workshop on Intelligent Bug Fixing (IBF)*. IEEE, 35–44.
  - [52] Chunhui Yuan and Haitao Yang. 2019. Research on K-value selection method of K-means clustering algorithm. *J—Multidisciplinary Scientific Journal* 2, 2 (2019), 226–235.
  - [53] Mingyi Zhao, Jens Grossklags, and Kai Chen. 2014. An exploratory study of white hat behaviors in a web vulnerability disclosure program. In *Proceedings of the 2014 ACM workshop on security information workers*. ACM, 51–58.
  - [54] Mingyi Zhao, Jens Grossklags, and Peng Liu. 2015. An empirical study of web vulnerability discovery ecosystems. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1105–1117.