All your (data)base are belong to us: Characterizing Database Ransom(ware) Attacks

Kevin van Liebergen*†, Gibran Gomez*†, Srdjan Matic* and Juan Caballero*
*IMDEA Software Institute
†Universidad Politécnica de Madrid

***Abstract*—We present the first systematic study of *database ransom(ware) attacks*, a class of attacks where attackers scan for database servers, log in by leveraging the lack of authentication or weak credentials, drop the database contents, and demand a ransom to return the deleted data. We examine 23,736 ransom notes collected from 60,427 compromised database servers over three years, and set up database honeypots to obtain a first-hand view of current attacks. Database ransom(ware) attacks are prevalent with 6K newly infected servers in March 2024, a 60% increase over a year earlier. Our honeypots get infected in 14 hours since they are connected to the Internet. Weak authentication issues are two orders of magnitude more frequent on Elasticsearch servers compared to MySQL servers due to slow adoption of the latest Elasticsearch versions. To analyze who is behind database ransom(ware) attacks we implement a clustering approach that first identifies campaigns using the similarity of the ransom notes text. Then, it determines which campaigns are run by the same group by leveraging indicator reuse and information from the Bitcoin blockchain. For each group, it computes properties such as the number of compromised servers, the lifetime, the revenue, and the indicators used. Our approach identifies that the 60,427 database servers are victims of 91 campaigns run by 32 groups. It uncovers a dominant group responsible for 76% of the infected servers and 90% of the financial impact. We find links between the dominant group, a nation-state, and a previous attack on Git repositories.**

I. Introduction

Databases are a key asset of digital services as they store the data required for the services to operate. Given the large amount of data they may store and the criticality of that data, database servers are a valuable target for attackers. In this work, we perform the first systematic study of *database ransom(ware) attacks*. In this class of attacks, the attackers identify target database servers by scanning the IPv4 address space or by using Internet scanning engines [13], [85], [47]. Once a target is located, the attackers try to login to the database server by leveraging the lack of authentication, using default credentials, or guessing weak credentials. If they manage to log in, they examine the databases the compromised account has access to, optionally exfiltrate the database contents, drop the content of those databases, and leave a ransom note (e.g., by creating a new database table with a catchy name) with instructions on how to get the data back. The note provides ransom payment details (e.g., a Bitcoin address and the ransom amount) or a contact method to request the payment details (e.g., email address, Tor hidden service address).

Database ransom(ware) attacks are popular, being frequently reported by security companies [1], [38], [64], [91], [19], going as far back as 2017 [1]. However, to the best of our knowledge, they have not been studied in the academic literature. Database ransom(ware) attacks are similar to traditional ransomware attacks [9], [67], [5], [44], [87], [52], [20], [41], [68] in the demand of a ransom to recover access to data. In fact, posts often refer to them as ransomware [38], [64], [19]. But, they have differences with traditional ransomware. First, instead of encrypting the victim's data and requesting a ransom to release the decryption key, database ransom attacks delete the database contents and request a ransom to return a copy of the deleted data. Another key difference is the absence of malware. Once logged into the database, the attackers can simply execute a sequence of database commands to delete the data and add the ransom note. The lack of malware makes it difficult to identify which database servers may have been infected by the same campaign or threat group. Due to the absence of malware, some reports refer to these attacks as *malwareless ransomware* [21] or *ransom attacks* [91]. Given the terminology differences, throughout this paper we will refer to these attacks as *database ransom attacks*. Another difference with ransomware targeting desktop computers or mobile devices is that attackers do not need to employ social engineering techniques to convince victims to install malware. Instead, they can scan for target servers on default database ports (e.g., 3306/TCP for MySQL, 9200/TCP for Elasticsearch). They also differ from server ransomware strains such as eChr0aix [2], Qlocker [33], and DeadBolt [3] that exploit software vulnerabilities to break into network-attached storage (NAS) servers and encrypt the stored data. Instead, database ransom attacks leverage misconfigurations that leave a database with no, or weak, authentication allowing the attackers to walk into the database and delete its data.

To study database ransom attacks, we obtain data about 60,427 compromised Elasticsearch, MySQL, and MariaDB servers (identified by their IP address) collected by the LeakIX Internet scanning engine [47] over 3 years from May 2021 until April 2024. We also setup database honeypots to obtain a first-hand view of current attacks. Using these data sources we answer the following 4 research questions.

(RQ1) How many groups are responsible for the database server infections? Given the absence of malware samples, we cannot use family classification to identify groups behind the database infections. Instead, we design a novel clustering approach to identify which victims may have been affected

Network and Distributed System Security (NDSS) Symposium 2025
24-28 February 2025, San Diego, CA, USA
ISBN 979-8-9894372-8-3
https://dx.doi.org/10.14722/ndss.2025.241887
www.ndss-symposium.org

by the same scam campaign and which campaigns may be run by the same threat group. Our approach takes as input 23,736 ransom notes obtained from LeakIX infection events. It first identifies which servers have been infected by the same attack campaign by examining the similarity of the ransom notes' text producing a set of note similarity (NS) clusters. Each NS cluster captures a campaign whose ransom notes are nearly identical, allowing for small changes in the note text such as different payment addresses or minor syntactic modifications. Then, it identifies which campaigns are run by the same threat group. For this, it first extracts indicators of compromise (IOCs) from the ransom notes (i.e., Bitcoin addresses, email addresses, Tor onion addresses) and merges NS clusters that share IOCs into IOC reuse (IR) clusters. Then, it leverages the Bitcoin blockchain data to obtain the multi-input (MI) clusters [63], [6], [78], [58] of the Bitcoin addresses extracted from the ransom notes. If two Bitcoin addresses in the same MI cluster have been extracted from notes in different IR clusters, those IR clusters are merged into a single group cluster that captures the infections attributed to the same threat group. This process outputs a cluster tree where campaigns (i.e., NS clusters) hang from group clusters. Our approach groups the 60,427 infections (i.e., infected servers) into 91 campaigns run by 32 groups. It reveals a dominant group that is responsible for 35 campaigns and 76% of the server infections.

(RQ2) Who is behind the attacks? Can the attacks be attributed? We examine the usefulness of the extracted indicators to start an investigation of a group. For 15 (45%) groups, including the dominant group, we have at least one Bitcoin address. For these groups, we trace the coin flows from the extracted Bitcoin addresses towards services with know-your-customer (KYC) requirements (i.e., requiring users to provide proof of identity), which can be used as attribution points by law enforcement agencies (LEAs) [35]. We find 10 such services (8 exchanges and 2 payment services) used by two groups, one being the dominant group. We also identify links from the dominant group to a previous campaign that compromised Git repositories [34] and to an address attributed by the U.S. Department of Defense to the Democratic People's Republic of Korea (DPRK, or simply North Korea) [95].

(RQ3) How much do the attackers make? We leverage Bitcoin transaction data to produce revenue estimations across all groups, as well as per group. In total, database ransom attacks have received 29.52 BTC, equivalent to $498K using the conversion rate at the time of each payment. The dominant group has generated a revenue of $449K (90% of all measured revenue). The requested ransom amounts are small with 94.4% of all deposits being at most $1,000. Small ransom amounts are likely used to incentivize payments in case the victim has doubts about the data being returned. As recently demonstrated by Gomez et al. [36] revenue estimates are affected by limited coverage from the Internet scanning engines and can be up to 39 times higher in reality. While one could be tempted to think that unauthenticated servers may simply be test servers with little valuable data, we observe that payments keep happening, with 66 payments in 2024 (until April 11). These victims likely have no backups of the deleted data and consider the data more valuable than the requested ransom amount. Attackers infecting our honeypots do not exfiltrate the database contents and we find additional supporting evidence this may often be the case. It is important to send the message to victims that if they are really desperate to get the data back, prior to performing a ransom payment, they should contact the attacker through the provided email and request proof that the attackers have the original data.

(RQ4) Why do these attacks keep happening? Despite database ransom attacks being ongoing for years, they are still rampant. Newly infected servers keep appearing with over 6K infected IPs in March 2024, a 60% increase over one year earlier. Furthermore, our honeypots get infected by database ransom attacks in 14 hours since connected to the Internet. Two-thirds (67%) of infected servers in our dataset run Elasticsearch, compared to 30% running MySQL, and 3% running MariaDB. This is surprising given that Internet scanning engines observe two orders of magnitude more MySQL servers than Elasticsearch servers connected to the Internet. Thus, weak authentication is two orders of magnitude higher in Elasticsearch servers compared to MySQL servers. Elasticsearch only introduced strong default credentials in version 8.0, released in February 2022. After 2.5 years, scanning data shows that only 7.6% Elasticsearch servers run 8.x versions, compared to 92.4% running older versions. Thus, adoption of up-to-date Elasticsearch software with secure default installation is slow, likely leading to servers with weak authentication and vulnerable to database ransom attacks. Still, 11% of Elasticsearch infections and 98% of MySQL infections happen on versions with more secure installation procedures, indicating that the wrong configuration by users is still an important factor in weak authentication issues.

II. Database Honeypots

To gather first-hand information on current database ransom attacks, we deployed MySQL honeypot databases over 12 days in June 2024 and 28 days in September 2024. We configured the honeypots with an empty string as password for the *root* user and populated them with fake data. We use the database logs of executed commands to analyze the actions taken by the attackers. We ran 5 VMs on two cloud hosting providers, each at a separate location: Hong Kong, India, Netherlands, Singapore, and USA.

We call an *infection* to a connection to the honeypot that deletes content and leaves a ransom note. We ignore other connections, e.g., those simply listing the existing databases, as they may not be related to database ransom attacks. All VMs were infected multiple times with infections spread among the VMs, i.e., no VM dominates the infections. The average time to infection was 14 hours since the VM was connected to the Internet, with a minimum of 7.6 hours and a maximum of 26.3 hours, showing that the attackers are constantly looking for new targets. After the first infection, all VMs were frequently reinfected, pointing to the attacks being automated.

We group infections by the sequence of database commands attackers execute. We observe two groups, each using a unique sequence of commands. Infections from both groups are summarized in Table I. Both groups leave their ransom notes in rows of a table in a newly created database. Group A uses the same "RECOVER_YOUR_DATA" name for the database and the table. Group B uses database "README_TO_RECOVER_TNA" and table "README". In addition to the new database, Group B also creates a

	Attacks				Ransom Notes			
ID	**VMs**	**Infect.**	**IP**	**ASN**	**Notes**	**Temp.**	**BA**	**Email**
Group A	5	123	58	23	123	1	6	123
Group B	5	8	4	4	8	2	4	3

Table I: Honeypot infections by attacking group. Groups are identified by the sequence of commands executed. For each group, it captures the number of VMs infected, infections, attacking IPs and their ASNs, the ransom notes and templates used, and the Bitcoin and email addresses in those notes.

"README" table with the ransom note in each database whose content it deleted. While each infection has a unique ransom note (by SHA256 of the text), there are only three templates (one for Group A and two for Group B), where Bitcoin addresses, email addresses, ransom amount values, and unique tokens may change, while the text remains the same. Figure 1 shows Group A's template.

Group A first infects all the VMs and reinfects them at least once a day. It exfiltrates the first 10 rows of each table, removes all databases (including its own, if present), and shuts down the MySQL server, likely to ensure it is not compromised by other attackers. However, our honeypots are automatically re-started in case of a shut down. Their attacks come from 58 IP addresses belonging to 23 autonomous system numbers (ASNs) with 4 preferred cloud hosting providers: Limenet (15 IPs), CDN77 (11), ASN-QUADRANET-GLOBAL (6), and M247 (4). All attacks in September 2024 use IP addresses hosted by Limenet whose webpage is titled "Premium, Secure and Anonymous" [53]. This possibly indicates Group A runs dedicated attack infrastructure, as opposed to using compromised hosts. We extract 6 unique Bitcoin addresses and 123 email addresses from their ransom notes. All the email addresses follow the same pattern "dzen+[0-9a-z]{4,5}@onionmail.org" where the variable part is the <TOKEN> in Figure 1. The ransom amount values range from 0.0122 BTC ($767) up to 0.0143 BTC ($900).

Group B is slower in identifying targets, finding the honeypots after Group A already infected them. It does not attempt to exfiltrate any data. It does not try to prevent other groups from re-infecting the servers. Their attacks come from 4 IP addresses, each in a different residential network. The first Group B infection in four VMs was on the same day and from the same IP address, further confirming the infections were from the same group. Their ransom notes contain 4 unique Bitcoin addresses and 3 emails. The ransom amount is always 0.007 BTC ($440), roughly half the amount Group A requests.

In summary, we observe two groups of attackers. Both seem to automate the attacks and behave similarly regardless of the VM location. However, Group A seems to be a more advanced operation. It identifies targets faster, uses unique tokens and email addresses in each infection, uses more IP addresses possibly in dedicated infrastructures, requests higher ransom amounts, and has more elaborate procedures such as shutting down the VMs post-infection. Importantly, the two groups do not exfiltrate the full data. Furthermore, in case of reinfections, both groups delete the ransom note left by the previous infection, making the infection look like it comes from the second group, even if the second group cannot have the data since it was already wiped. Thus, some database

All your data is backed up. You must pay <AMOUNT> BTC to <BITCOIN> In 48 hours, your data will be publicly disclosed and deleted. (more information: go to <URL>). After payment send mail to us: <EMAIL> and we will provide a link for you to download your data. Your DBCODE is: <TOKEN>)

Figure 1: Ransom note template used by Group A in their honeypot infections.

ransom attacks are scams where the attacker cannot return the data, even if the victim pays, an issue further discussed in Section VIII. In the rest of the paper we analyze a longitudinal dataset with three years of database ransom attacks and propose an approach to identify the number of groups involved, beyond the two groups observed by our honeypots.

III. LeakIX Dataset

We obtained data from the LeakIX Internet scanning engine [47]. LeakIX has plugins that check for specific server vulnerabilities, including unauthenticated databases. Once it locates an unauthenticated database, LeakIX applies undisclosed regular expressions to the names of files, collections, and tables in the database to identify ransom notes. If a note is found, an infection event is produced. This approach may miss infections that do not use the expected keywords to name ransom notes.

On April 5, 2024, we used the search API of LeakIX to query historical infection events produced by the *ElasticSearchOpen* and *MysqlOpen* plugins, which are the two plugins that collect the ransom note text. The *MysqlOpen* plugin provides events for MySQL and MariaDB (an open-source fork of MySQL) databases. The search query returns the last infection event for each IP address, but the same IP address may have multiple infection events, possibly with different ransom notes, e.g., the server was infected multiple times or the IP address was used by different servers over time. For each returned IP address, we query the LeakIX hosts API to obtain the list of all events for the IP address. We filter out events without a ransom note. The regular expressions used by LeakIX for identifying ransom notes can occasionally introduce false positives, e.g., identify as a ransom note a database table with configuration information. To address this issue, as an additional filter we remove all the events whose ransom notes do not contain at least one keyword associated with ransom notes (e.g., hacked, backed, BTC, bitcoin). We built our keyword list iteratively by examining which notes were filtered and not filtered.

Each LeakIX infection event provides the following features: the event timestamp, the IP address and port of the infected service, the country code (CC) and autonomous system number (ASN) for the IP address, the SHA256 hash of the ransom note text, and the service's software, namely the server OS (e.g., Windows, Linux), the database software (i.e., Elasticsearch, MySQL, MariaDB), and the database software version.

Table II summarizes the collected dataset, which contains 302,246 events with valid ransom notes obtained from 60,427 infected IP addresses running Elasticsearch, MySQL, or MariaDB databases. All addresses are IPv4, as LeakIX focuses on enumerating the IPv4 address space. IPv6 scanning is an area

Database	Events	IP	CC	ASN	Notes	Addr.
ElasticSearch	254,629	40,663	123	1,926	17,391	86
MySQL	43,678	18,224	106	1,198	5,790	316
MariaDB	3,939	1,906	75	491	661	191
All	302,246	60,427	139	2,646	23,736	401

Table II: LeakIX dataset summary.

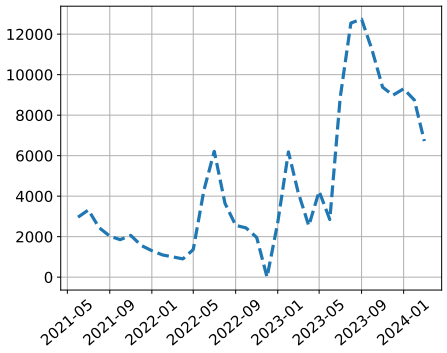


Figure 2: Monthly number of infected IPs.

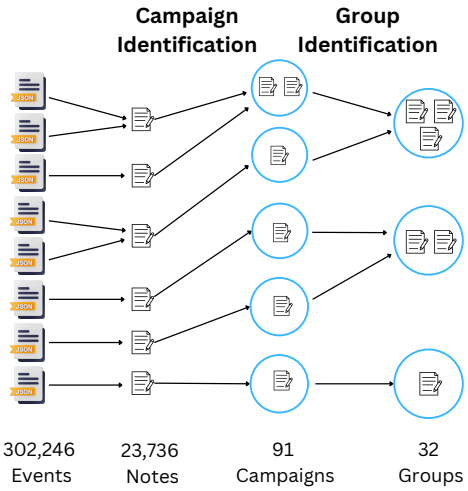


Figure 3: Overview of our clustering approach. It takes as input LeakIX events, extracts ransom notes in the events, groups notes into campaigns, and identifies which campaigns are run by the same threat group.

of active research [113], [49], [39]. The IP addresses are hosted in 139 countries and belong to 2,646 ASNs. The top countries by number of IP addresses are China (47.5%), US (16.0%), and Germany (5.1%). The AS distribution is similarly dominated by Chinese ASNs belonging to Alibaba (21.2%) and Tencent (15.5%), followed by ASNs used for cloud hosting services by Amazon (12.8%) and Google (4.7%). The fact that China has three times more infected servers than the next country may be due to some groups specifically targeting this country (we show one such group in Section VII). It could also be due to servers in China lacking authentication more often.

Roughly two thirds of the infected IPs (67.3%) correspond to Elasticsearch servers, followed by MySQL (30.1%), and much fewer MariaDB servers (3.1%). The servers run primarily on Linux used on 97.1% of Elasticsearch servers, 83.7% of MySQL servers, and 64.9% of MariaDB servers. The next most popular OS is Windows that is used on 35.0% MariaDB, 15.7% MySQL, and 2.1% Elasticsearch servers. Other servers run on MacOS, FreeBSD, OpenBSD, Solaris, and UNIX.

Figure 2 shows the number of monthly infected IP addresses. The infections start on May 25, 2021, likely when LeakIX introduced the plugins to identify unauthenticated databases. Infected servers grow until August-September 2023 and decrease after that, but are larger in February 2024 than prior to August 2023, showing that infections still happen. There are no infections in December 2022, likely because LeakIX did not scan for infections on that month.

The dataset contains 23,736 unique ransom notes (identified by SHA256 hash), with a ransom note appearing on average in 117.8 events from the same or different IP addresses. There are 15,331 (4.9%) events with more than one ransom note. These can happen due to groups (e.g., Group B in Section II) that may leave the ransom note in multiple databases and also due to re-infections of the same server where the later infections failed to remove the ransom notes left by the earlier infections, e.g., due to missing drop permissions. For MySQL and MariaDB databases, the ransom notes are in plain text. For Elasticsearch, they are in JSON format with the attackers placing the ransom note in a field with a name that entices to read it (e.g., *Readme*, *readThis*). When extracting ransom notes from the JSON content, we assume the longest field is the ransom note. We identified the language of the note using langdetect [46]. Ransom notes were written in one of two languages: English, 22,548 notes (95.0%) and Chinese 1,116 (4.7%). The other 0.3% notes are too short for the language to be identified.

IV. Approach Overview

Figure 3 summarizes our clustering approach. It takes as input the LeakIX infection events and outputs a cluster tree where the top clusters correspond to groups and the clusters hanging from a group correspond to the campaigns it runs. The clustering comprises two steps: *campaign identification* and *group identification*.

The goal of campaign identification is to group the ransom notes from infection events into campaigns. We define a campaign as a set of ransom notes created from the same template containing the common text of the notes, as well as macros for elements that may change across notes such as the payment address, ransom amount, contact email, and the list of deleted databases. Such templates are analogous to those used in email [45], [73] and Twitter spam [32]. Campaign identification is detailed in Section V. It comprises two steps. First, it normalizes the ransom note text by replacing elements that change (e.g., payment addresses, ransom amounts, database lists) with macros, thus identifying notes constructed using the same template. Then, it clusters similar templates to capture small modifications producing *note similarity* (NS) clusters, where each NS cluster corresponds to a campaign.

Group identification takes as input the NS clusters and merges clusters determined to be run by the same group into *group clusters*. Group identification is detailed in Section VI.

```
AMOUNT= ([0-9]+([.][0-9]+)?)( )?(比特币|BTC|bitcoin|XMR)
LIST= ([:: ](\s?([0-9a-zA-Z#_\-\+?]+\s?)[,。.])+)\s)
TOKEN= token[:]?[ ]+([0-9a-zA-Z]+)\b
```

Figure 4: Our regular expressions for identifying ransom amounts, lists of database names, and unique identifiers.

It comprises two steps. First, it merges NS clusters that share some selected indicators of compromise (IOCs) that are specific to the threat actors behind the campaigns, namely Bitcoin addresses, email addresses, and onion addresses. We call the resulting merged clusters *IOC reuse* (IR) clusters. Second, it analyzes the Bitcoin blockchain to further merge IR clusters if they contain addresses that belong to the same multi-input (MI) cluster [63], [6], [78], [58]. This step identifies campaigns run by the same group that do not reuse IOCs across campaigns, but aggregate the revenue from different campaigns before cashing out.

Throughout the campaign and group identification steps, our approach updates cluster properties such as the number of servers infected, the cluster lifetime, and the indicators extracted from the ransom notes in the cluster. For each campaign and group cluster, it computes the financial revenue by examining the deposits to the Bitcoin addresses appearing in the cluster's ransom notes. Clusters can be ranked according to the property of choice, enabling comparison and prioritization for investigation. We analyze the generated clusters in Section VII.

V. Campaign Identification

Campaign identification groups infection events into campaigns based on the similarity of their ransom notes. First, it normalizes the ransom note text by replacing elements that change with macros (Section V-A). Normalization identifies notes that have been generated from the same template, i.e., have identical text and only differ in elements that are specific to the victim (e.g., list of deleted databases, unique victim tokens) or that introduce polymorphism for evasion purposes (e.g., payment addresses, contact emails, ransom amounts). For example, in Section II Group A used 123 notes, but all of them normalized to the same template in Figure 1. Then, it clusters similar templates into campaigns (Section V-B). This clustering is designed to be very conservative so that it only groups largely identical templates, i.e., that only differ due to the introduction and removal of spaces, punctuation, and a few words. Such modifications are used by attackers to add polymorphism to the ransom notes. Larger differences (e.g., affecting whole sentences) make the templates be placed in different clusters, which avoids grouping unrelated campaigns.

A. Ransom Note Text Normalization

To identify elements that change we use regular expressions. We use the *iocsearcher* indicator extraction tool [42] to identify blockchain addresses, universal unique identifiers (UUIDs), and networking indicators such as email addresses, Tor onion addresses, URLs, and domain names. *iocsearcher* validates the checksum embedded in blockchain addresses and Tor onion addresses (version 3) avoiding false positives. We build our own regular expressions in Figure 4 to identify lists of deleted databases, ransom amounts, and unique token identifiers that do not follow the UUID format.

	Clusters			
Threshold	**Total**	**Non-singleton**	**Singleton**	**Top-10 Precision**
5	553	106	447	100.00%
6	408	94	314	100.00%
7	321	82	239	99.98%
8	252	77	175	99.98%
9	216	71	145	98.84%
10	184	65	119	81.27%

Table III: Note similarity threshold evaluation. The best threshold is 6 because it provides perfect precision while minimizing the number of clusters.

The most common indicators *iocsearcher* extracts are email addresses (18,361) where victims can contact the attacker, followed by Bitcoin addresses (401) for the victim to pay the ransom. There are also 3 Tor onion addresses, used in notes that ask the victim to visit a Tor hidden service to obtain the payment information. We find one Monero payment address, but no addresses for other popular blockchains (e.g., Ethereum, Cardano), showing that Bitcoin is the preferred payment method. Other extracted indicators are universal unique identifiers (UUID) used as unique victim IDs (36), URLs (35) to provide additional information (e.g., where to buy Bitcoins), and domains embedded in URLs and email addresses (84). For the interested reader, Appendix A evaluates the accuracy of the indicator extraction process.

We replace the identified elements with macros using a custom wrapper for *iocsearcher*. For example, the string "You must pay 0.02 BTC to 1757buFTry kguUzQNQgUSrdoQuyEJEF6CW" is replaced with "You must pay <AMOUNT> BTC to <BITCOIN>". Similarly, the sentence "Databases that we have: legal-advice, pet-fodder-shop, legal-advice2." is replaced with "Databases that we have: <LIST>." This process outputs 14,220 templates a 40.1% decrease from the 23,736 notes due to identifying notes with the same text but different indicators.

B. Note Similarity Clustering

We define a campaign to be a set of templates that are largely identical, but contain minor differences due to the addition or removal of spaces, punctuation characters or a few words, as well as due to normalization failures. To identify templates in the same campaign we examine the similarity of the 14,220 templates using SimHash [15], a fuzzy hash that produces similar digests for similar inputs. SimHash is designed to detect near-duplicate data more efficiently than classical pairwise similarity metrics, like the Jaccard index, which has quadratic complexity. It is Google's preferred algorithm for detecting near-duplicate web pages when crawling the Web [55]. We compute the SimHash of each template and add the digests to an index where digests with similarity above a threshold are placed in the same bucket, i.e., NS cluster.

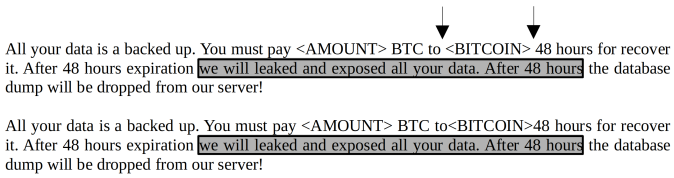
We search for a threshold that conservatively groups templates with minor differences. For this, we cluster the notes using different thresholds and measure the clustering precision on the 10 largest clusters, which contain 70%–85% of the notes depending on the threshold. We examine the entries in the top

Clustering type	All	Non-singl.	Max.	Med.	Mean
Normalization	14,220	139 (40.7%)	3,559	2	69.5
+Note similarity	91	42 (99.8%)	13,701	20	564.0
+IOC reuse	33	19 (99.9%)	13,704	25	1,248.5
+MI clust. (Final)	32	18 (99.9%)	18,792	25	1,317.9

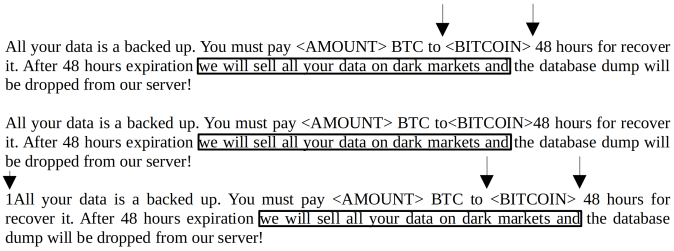
Table IV: Clusters output after each clustering step. The bottom row is the final clustering. The percentage is over the number of notes. Mean and median are over non-singleton clusters.

Step	CID	Notes	Events	IP	NSC
Normalization	NO1	3,559	20,040	3,789	-
	NO2	1,003	2,271	737	-
	NO3	743	903	742	-
	NO4	733	1,011	740	-
	NO5	692	1,068	653	-
Note similarity	NS1	13,701	47,517	16,565	1
	NS2	3,628	42,537	9,366	1
	NS3	1,941	3,826	1,497	1
	NS4	1,481	1,514	1,143	1
	NS5	635	848	655	1
IOC reuse	IR1	13,704	47,521	16,567	3
	IR2	5,088	102,509	30,822	32
	IR3	1,962	3,862	1,506	7
	IR4	1,493	1,525	1,153	6
	IR5	604	1,311	628	2
Bitcoin multi-input	MI1	18,792	149,958	45,778	35
	MI2	1,962	3,862	1,506	7
	MI3	1,493	1,525	1,153	6
	MI4	604	1,311	628	2
	MI5	438	988	449	9

Table V: Top 5 clusters at each clustering step. NSC corresponds to the number of NS clusters merged into that cluster.



(a) The two templates in note similarity cluster NS-6459.



(b) The three templates in note similarity cluster NS-9926.

Figure 5: Example illustrating the conservative design of our NS clustering. The two templates in cluster NS-6459 (subfigure 5a) are nearly identical with the only differences being spaces before and after the Bitcoin address marked with arrows. The three templates in cluster NS-9926 (subfigure 5b) are nearly identical with the only differences being spaces before and after the Bitcoin address and an extra character at the beginning of the bottom template, marked with arrows. The notes in the two NS clusters are also similar, but they are placed in two separate NS clusters due to the differences marked with square boxes.

10 clusters classifying them as true positives (TPs) if they are nearly identical to the rest of the cluster, and as false positives (FPs) otherwise. TPs and FPs are aggregated across the top 10 clusters and we compute $Precision = TP/(TP + FP)$. Table III summarizes the threshold selection evaluation. As the threshold increases, the precision decreases and the number of clusters reduces. We cannot compute the recall as we lack a ground truth to identify the false negatives. But, the fewer clusters with perfect precision the higher the recall will be. We choose 6 because it is the largest threshold with no errors on the top 10 clusters, grouping only nearly identical templates.

NS clustering results. The second row in Table IV captures the results of the NS clustering using the normalization and the selected threshold value. The 14,220 templates are grouped into 91 NS clusters, of which 42 contain multiple notes and 49 are singletons (i.e., clusters of size one). Excluding the singletons, the median cluster size is 20 notes with a mean of 564 notes. All NS clusters contain notes written in a single language. Table V details the 5 largest clusters for each clustering step. The largest campaign contains 13,701 notes and infected 16,565 IPs.

To evaluate the impact of the normalization, we cluster the 23,736 notes without normalization obtaining 832 NS clusters where 666 are singletons and the other 166 clusters have a median of 6 notes. This experiment demonstrates the positive impact of normalization, which reduces the number of NS clusters from 832 to 91, nearly a factor of 9. Without normalization, the text similarity clustering struggles to group similar notes where the indicator values (e.g., database lists) are responsible for most of the text.

NS clustering example. Figure 5 illustrates the conservative design of the NS clustering. Figure 5a shows the two templates in cluster NS-6459. Both templates are nearly identical with the only differences (marked with arrows) being extra spaces before and after the Bitcoin address in the top template. Figure 5b shows the three templates in cluster NS-9926. The three templates are nearly identical with the only differences (marked with arrows) being spaces before and after the Bitcoin address and an extra character at the beginning of the bottom template. The templates in both NS clusters are similar, but differ in the text in the rectangular boxes, which are the reason why both NS clusters are not merged.

VI. Assigning Campaigns to Groups

Group identification determines which campaigns are run by the same group. First, it merges NS clusters that share indicators into *IOC reuse* (IR) clusters, as detailed in Section VI-A. Second, it analyzes the Bitcoin blockchain to further merge IR clusters if they contain Bitcoin addresses that belong to the same multi-input (MI) cluster [63], [6], [78], [58], as detailed in Section VI-B.

A. IOC Reuse

Multiple NS clusters may belong to the same owners due to the attackers running multiple campaigns and also because the selected note similarity threshold is conservative and thus may

fail to cluster some similar notes. To address these issues, we merge NS clusters that share indicators. For example, if notes in different NS clusters request the victims to pay to the same Bitcoin address, or to contact the attackers using the same email or onion address, those campaigns are highly likely to be run by the same group. In Figure 5, the notes in the two NS clusters contain 9 and 33 Bitcoin addresses, respectively. Of those, three Bitcoin addresses appear in both NS clusters, causing the IOC reuse step to merge them into the same IR cluster.

The IR step focuses only on selected indicators of compromise (IOCs): blockchain addresses, email addresses, and onion addresses. Other indicator types do not necessarily indicate that the attacking group is the same, and thus are not used for merging. In particular, URLs are used to direct users to exchanges to buy Bitcoins, domains in emails most often belong to mail services, token identifiers are unique to one note and thus cannot be used for merging, and lists of dropped databases are specific to a victim server.

Merging clusters by IOC reuse. First, we compute the IOC set of each NS cluster by performing the union of the IOC sets (i.e., Bitcoin, email, and onion addresses) extracted from each ransom note in the NS cluster. Then, we merge NS clusters that share IOCs. The merging process starts with an empty set of IR clusters and iterates on the list of NS clusters. For each NS cluster, it checks if its IOC set has a non-empty intersection (i.e., shares at least one IOC) with the IOC set of any of the IR clusters. If the NS cluster shares IOCs only with one IR cluster, the NS cluster is merged into that IR cluster. If the NS cluster shares IOCs with multiple IR clusters, those IR clusters as well as the NS cluster are merged together. If the NS cluster is not similar to any IR cluster, a new IR cluster is created.

IOC reuse clusters. Table IV shows that the 91 NS clusters are merged into 33 IR clusters. Of those, 19 contain more than one note, accounting for 99.9% of the 23,736 notes. The remaining 14 are singleton clusters. Overall, IOC reuse reduces the number of clusters by 63%. The reduction is even more significant for singleton clusters that decrease from 49 to 14 (28%). On average, an IR cluster contains 50% more ransom notes than an NS cluster. The median size of an IR cluster is 25 ransom notes (mean of 1,186), compared to a median of 20 (mean of 564) for NS clusters.

The IOC reuse block in Table V shows the five largest IR clusters. The rightmost (NSC) column in the table captures the number of NS clusters merged into that IR cluster. For example, the IR2 cluster corresponds to 32 merged NS clusters including 2 of the top 5 NS clusters (NS2, NS5). When multiple NS clusters are merged into an IR cluster, the IR cluster may not have a common template for the ransom notes, as each NS cluster may have a different template.

The IOC reuse step reduces the number of clusters by 75%. Still, it is possible that some NS clusters from the same owners did not reuse IOCs and have not been merged. To identify campaigns from the same owners that do not reuse IOCs, we leverage information from the Bitcoin blockchain.

B. Bitcoin Multi-Input Clustering

We leverage the public Bitcoin transaction data to further merge IR clusters. For this, we use the open-source platform WatchYourBack [35] to analyze all Bitcoin transactions from the blockchain inception until April 11, 2014 (block height 838,800), First, we examine which payment addresses extracted from the ransom notes have received funds. Of the 401 Bitcoin addresses, 200 (49.9%) have received at least one deposit. We call these 200 Bitcoin addresses the *seeds*.

Next, we expand the seeds using multi-input (MI) clustering [63], [77], [78], [6], a heuristic to obtain maximum sets of addresses that share ownership according to their transactions. MI clustering considers that addresses that co-spend together in the same transaction are controlled by the same owner, since they should share access to the private keys of the input addresses to sign the transaction. MI clusters (MICs) are created transitively. If a transaction has addresses A and B as inputs, and another transaction has B and C as inputs, then A, B, and C all belong to the same MIC (i.e., owner).

Prior to computing the MICs, we use the tag database and the exchange classifier from WatchYourBack to identify online wallets, which are addresses assigned to users to receive deposits within a service (e.g., exchanges). Online wallets are owned by the services who control their private keys. Thus, online wallets should not be expanded because a service may potentially produce transactions using several of the addresses under its control at the same time (e.g., for efficiency), breaking the assumptions of the multi-input heuristic. If a seed corresponds to an online wallet in a service, the MIC of that seed may incorrectly include online wallets from other unrelated clients of the service [36] into our expanded sets. We identified three online wallets among the seeds, all of them in different MICs. Two were identified as the exchanges Coinbase and Poloniex by the tag database. The third one is detected by the exchange classifier, so the name of the service remains unknown. For these three seeds, we do not use their MICs and instead place each seed by itself in its own MIC.

Overall, the 200 seeds belong to 178 MICs. Of those, 50 MICs contain more than one address and 128 are singletons. The 50 non-singleton clusters contain an additional 64 non-seed addresses that are not part of the LeakIX dataset but were identified by the MI clustering as also belonging to the groups running the corresponding IOC clusters. We keep track of those additional addresses as a MIC feature, and include them in the revenue estimations in Section VII. But, they are not included in the number of addresses in a campaign or group.

Finally, we merge IR clusters that contain addresses in the same MIC. The merging procedure is the same as detailed in Section VI-A but uses a non-empty intersection of MIC identifiers (instead of IOC sharing) as a signal to merge clusters. As shown in the bottom row of Table IV, MI clustering of Bitcoin addresses further reduces the number of clusters from 33 after IOC reuse to the final 32 groups. The reduction is significantly smaller than the one obtained with IOC reuse; a possible explanation is that attackers try to avoid MI clustering on purpose, as recently pointed out [36]. Still, this step critically identifies that the two largest IR clusters are indeed controlled by the same group.

Step	Clust.	Singl.	Max.	Med.	Mean	Time
Normalization	14,220	14,081	3,559	2	69.5	32m
Note similarity	833	668	13,669	6	139.8	106m
IOC reuse (IR)	1,141	992	9,630	8	152.6	22m
MI clustering (MIC)	2,803	2,552	5,818	8	84.3	15m
IR+MIC	1,131	985	9,630	8	155.8	38m
All	32	14	18,792	25	1,317.9	176m

Table VI: Ablation study results for clustering the 23,736 notes using each approach step separately, using IOC reuse (IR) and multi-input clustering (MIC), and using all 4 steps.

C. Ablation Study

We perform an ablation study to quantify how much our whole clustering approach improves compared to using each step (i.e., normalization, note similarity, IOC reuse, MI clustering) in isolation. For each step, we perform an independent clustering of the 23,736 notes and report the results in Table VI. The step that clusters the most by itself is the note similarity clustering, which produces the smallest number of clusters (833), the smallest number of singleton clusters (668), and the largest cluster (13,669). However, the number of clusters is 22 times larger than the 32 groups our whole approach identifies (bottom row in Table VI). It is also the slowest step taking 1.8 hours. IOC reuse is the second step that groups most by itself producing 1,141 clusters and the highest average cluster size (152.6). For MI clustering, we first group notes containing the same Bitcoin address (a lightweight form of IOC reuse) and place notes with no Bitcoin addresses in singleton clusters. Then, we calculate the MI clustering on the Bitcoin addresses and merge clusters containing addresses in the same MI cluster. This step produces 2,803 clusters with a mean cluster size of 84.3 notes. MI clustering is a separate process that only needs to be run when the blockchain height is updated, e.g., once a day to add the transactions of the last day. It takes roughly 15 minutes depending on the number of new transactions. Normalization achieves little grouping by itself producing 14,220 clusters. However, as shown in Section V-A it significantly improves the NS clustering step.

We also evaluate the group identification (IOC reuse and MI clustering) without the campaign identification. This combination can be used as a lower bound on the number of groups, e.g., if there are concerns that campaign identification may over-cluster. This combination identifies 1,131 groups, in contrast to the 32 groups the whole approach identifies.

The ablation study shows that our carefully designed 4-step clustering is significantly better than the individual steps run by themselves. Overall, given the 23,736 notes and the pre-computed MI clusters, our whole approach identifies 32 groups in under 3 hours.

VII. Cluster Analysis

For each group and campaign cluster, our approach automatically produces a set of features that capture distinctive properties such as the number of infected servers, the per-country geographical distribution of those servers, the services targeted (i.e., MySQL, MariaDB, ElasticSearch), and the lifetime. We also compute the financial impact of the cluster using the WYB Bitcoin revenue estimation tool [111]. We compute the two recommended estimations [36]. The

Clusters		Infections		IOCs			Revenue		Life
CID	NS	IP	CC	BA	OA	Email	BTC	USD	Days
12373	35	45,778	135	305	-	17,401	27.15	$449,922	1,813
3	1	16,645	97	2	-	1	0.06	$2,181	187
6	1	14,434	96	2	-	2	0.08	$2,322	231
12926	7	1,506	56	63	-	11	0.75	$28,365	300
13549	6	1,153	51	-	2	-	-	-	1,019
13118	2	628	30	-	-	603	-	-	249
13390	9	449	1	16	-	32	0.29	$7,715	392
8994	1	177	19	-	-	157	-	-	249
35	1	155	25	-	-	131	-	-	504
434	1	58	12	3	-	3	0.6	$612	2,588
123	1	56	11	1	-	2	0.09	$1,759	1,507
135	2	52	16	-	1	-	-	-	563
467	1	45	14	-	-	-	-	-	368
2704	4	34	2	-	-	1	-	-	741
42	1	23	1	2	-	1	0.05	$1,704	644
1760	1	22	8	1	-	1	0.02	$210	1,364
1181	2	18	3	-	-	2	-	-	84
318	1	14	7	2	-	2	0.4	$2,617	1,814
527	1	12	9	-	-	-	-	-	504
89	1	9	2	-	-	2	-	-	248
7452	1	4	2	1	-	1	0.02	$775	83
2064	1	3	3	-	-	1	-	-	159
5431	1	2	1	1	-	1	-	-	792
7192	1	2	2	-	-	1	-	-	229
5034	1	2	1	-	-	1	-	-	6
2583	1	1	1	-	-	-	-	-	10
12028	1	1	1	-	-	1	-	-	7
7745	1	1	1	1	-	-	-	-	1
7797	1	1	1	-	-	1	-	-	1
8372	1	1	1	1	-	-	-	-	1
13087	1	1	1	-	-	1	-	-	1
13639	1	1	1	-	-	1	-	-	1
All	91	60,427	139	401	3	18,361	29.52	$498,187	2,648

Table VII: Group comparison. BA and OA correspond to Bitcoin and Onion addresses, respectively.

DD-DC estimate provides a lower bound on the revenue by aggregating the deposits to the cluster seeds minus double-counting transactions where cluster addresses appear as both inputs and outputs. The DD-OW+MI-DC is a tighter estimate that considers all revenue from addresses in the multi-input clusters also removing the double counting. As recommended, we convert the BTC estimation to US dollars (USD) using the conversion rate on the day of each payment.

A. Group Analysis

Table VII compares the 32 groups. Groups are ranked by number of infected IP addresses, but our approach can rank them according to any group feature. The results show that there is a dominant group (CID-12373) that ranks first across all features, including the number of infected servers (45,778 IP addresses), the lifetime (1,813 days), the revenue (27.15 BTC), and the number of indicators that can be used to bootstrap the attribution (305 Bitcoin addresses and 17,401 email addresses). Next, we compare groups across features.

Infected servers. The dominant group infects over 45K IP addresses throughout our study period, three times more than the next two groups that infect 14K–16K IPs. Two other groups infect over 1K IPs, four infect over 100 IPs, and 23 infect less than 100 IPs. The larger the number of infected IP addresses, the more important the group is.

Geographical targeting. Of the 32 groups, 26 use ransom notes in English, 5 use ransom notes in Chinese, and the dominant group uses ransom notes in both languages. According to the country codes in LeakIX events, the 5 groups exclusively using notes in Chinese have all their infections in China, except for one infection in Myanmar. These 5 groups

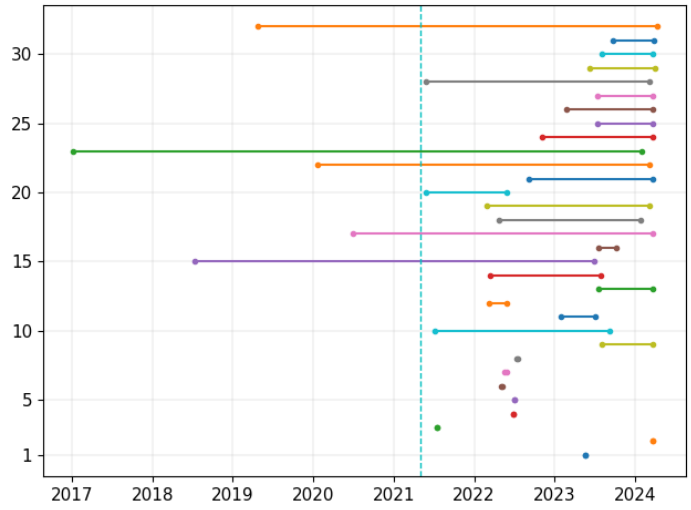


Figure 6: Group lifespan. The groups are sorted by number of infected IPs. The vertical line on May 2021 marks the start of the LeakIX dataset.

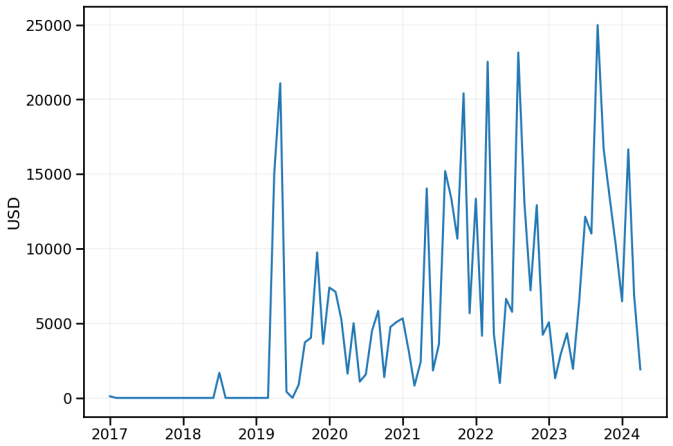


Figure 7: Total monthly revenue across all groups.

seem to specifically target databases in China. This is most evident in CID-13390 which infected 449 IPs, all in China. We do not observe targeting of other countries beyond China. Of the 26 groups using ransom notes in English, those with more than 10 infections target multiple countries, showing no signs of geographical targeting. They likely search for any targets they can find. These groups use ransom notes in English to infect databases in any country, including China, showing no language customization. One group (CID-135) infects 52 IPs but none of them are in China. This is surprising given that 47.5% of all infections are in China, possibly indicating that this group avoids targets in China.

Service targeting. Of the 32 groups, 19 only infect MySQL/-MariaDB servers, 11 only infect Elasticsearch, and two (dominant and CID-6) infect both database types. This points to smaller groups specializing in a target service, while two of the three larger groups target multiple databases.

Lifetime. Figure 6 shows the lifespan of the groups. There are 5 groups including the dominant group that have deposits to their addresses prior to May 2021, showing that they were operating prior to the start of our LeakIX dataset. The dominant group has been active since at least April 2019 and CID-434 has operated since 2017. On the other hand, 5 groups have been observed on a single day and another 3 for at most 10 days. There are 18 groups (including the dominant group) with infections in 2024 showing they are still active.

Revenue. From the 14 groups with Bitcoin addresses, 11 have received payments and 3 did not have any deposit. For the other 18 groups, we cannot produce a revenue estimation due to the lack of seed addresses. Figure 7 shows the total revenue per month for all 11 groups with at least one payment. In total, all groups have received 29.52 BTC, equivalent to $498K using the conversion rate of the day of each payment. As discussed in Section VIII, the revenue we measure is a conservative lower bound on the actual revenue due to LeakIX only seeing a fraction of all database ransom attacks. Database ransom attacks are still a profitable business with 66 victim payments in 2024 (until April 11). Thus, infections of new database servers keep happening, and some server owners consider the deleted data valuable enough to pay the ransom.

The dominant group receives the most revenue with $449,922, 90.3% of the total dollar revenue. The next group by revenue obtains $28,365 (5.7%). Appendix B details the revenue estimation for each group. The measured revenue is modest compared to other cybercrimes [36]. A key reason is that the requested ransom amounts are small with 94.4% of all deposits being at most $1,000. Small ransom amounts are likely used to incentivize payments in case the victim has doubts about the data being returned.

Attackers move the funds quickly with 195 (97.5%) payment addresses having no balance on April 11, 2024. However, five payment addresses from three groups still have balance. For these, it may be possible to trace withdrawals from these addresses in the mempool (i.e., prior to being committed to the blockchain). If the destination address belongs to a known exchange, LEA could request the exchange to block the funds before they are withdrawn.

Reports. We also check if payment addresses have been reported in external datasets. None of the addresses appear in the US Treasury OFAC Sanctions List [97], but 102 have been reported to BitcoinAbuse [11]. Of those, 98 belong to the dominant group. All reports describe database ransom attacks. Five reports state that the victim paid but the data was not returned [103], [104], [105], [106], [107]. Another six reports mention the victim tried to contact the attackers using the provided email, but no response was received [100], [101], [99], [102]. We discuss the implications for data recovery in Section VIII. We also query the most used Bitcoin addresses for each group in search engines to check for external reports. We find payment addresses from two groups (dominant group, CID-6) that have been reported also as payment addresses for database ransom attacks that target MongoDB databases [30], [17]. In both cases, the ransom note in the reported MongoDB infection is nearly identical to ransom notes used by the group in Elasticsearch infections. Another payment address of the

dominant group has been reported as a payment address for database ransom attacks that target PostgreSQL databases [19]. Again, the reported PostgreSQL ransom note is nearly identical to a ransom note from the dominant group. The reuse of payment addresses conclusively indicates that some groups (dominant group, CID-6) target multiple databases. They not only target Elasticsearch and MySQL servers in our dataset, but also MongoDB and PostgreSQL (dominant group). A payment address from CID-318 was used in a database ransom attack on the FitMetrix fitness company [110]. This company left three of its servers without password, exposing 113.5M customer records to the Internet, and to database ransom attacks. Each record contained sensitive user data such as name, gender, email address, phone numbers, profile photos, and their primary workout location. This episode highlights that weak authentication vulnerabilities affect also companies handling sensitive user data.

Contact emails. There are 25 groups with email addresses in their notes. Email addresses often belong to privacy-preserving email services such as onionmail.org, tutanota.com, mail inator.com, protonmail.com, and proton.me. Those email addresses may not provide useful attribution leads unless the email service is intervened. There are 22 groups using only email addresses in privacy-preserving email services. In the rest, we observe email addresses from large email providers such as yahoo.com or from attacker-owned domains (e.g., mysql222.com, mydatabase.to). If attackers do not intend to reply to victims at all, they could also use made-up email addresses, rendering the email addresses useless for attribution.

Onion services. There exist two groups that only provide an onion address in their ransom notes. The user is requested to visit the hidden service and provide a unique identifier in the note. CID-13549 has two onion addresses, the older ransom notes use a deprecated v2 address (hn4wg4o6s5nc7763.onion), later replaced with a v3 address[1]. These two addresses are no longer active, but reports on the v2 address [38] explain that it provided a dashboard where victims could enter their token to obtain the payment address. Furthermore, for victims who did not pay, the stolen databases were offered for sale, each for a price of 0.03 BTC ($520 at the time). The sale was advertised in a Russian forum claiming 85,000 MySQL databases were available [22]. The sale likely indicates this group had been exfiltrating the data prior to deletion. We discuss this in Section VIII. CID-135 has one v3 onion address[2] no longer active and for which we have found no reports. Attribution of these two groups would require previously proposed Tor hidden service deanonymization techniques [66], [10], [57]. While attribution is harder, it is possible, as indicated by Tor hidden services that have been taken down by LEAs [96], [28] including the recent takedown of the Lockbit ransomware hidden service [29].

Lack of indicators. No indicators are extracted for 3 groups. These correspond to campaigns that do not demand a ransom. For example, one group has a campaign using the note: "Your organization's database has crashed in our cyber crawler. All information is deleted from the company's servers to protect users' privacy. Please fix the loophole next time we take more drastic steps Gesh 4S Israel Group" where "Gesh 4S Israel Group" does not seem to exist. A second campaign uses a nearly identical note but the mentioned entity is instead "TEAMTNT && HRM BOTNET", where TeamTNT is a known threat actor [61]. One hypothesis is that these groups that demand no ransom are simply testing their capabilities.

[1] o42xfh5kao7mrtesnok5jgdsfagjsgzxlxdlpkpd2x6lpckhzk225yad.onion

[2] godransm3nnlofdwounmfdfaaivjzlnkeslxmo6siw45gn2gjy7av2qd.onion

B. Linking Honeypot Groups

In this section we examine whether we can link the two groups identified in our honeypots to the groups our clustering of LeakIX events identifies. For this, we create infection events for the 131 honeypot infections. We first cluster only the 131 honeypot infections, resulting in three NS clusters, one for each template identified in Section II. Since there is no IOC reuse and the deposits to the Bitcoin addresses are very recent, the whole clustering also outputs three clusters: one for GroupA and two for Group B.

Next, we cluster the 131 honeypot infections with all LeakIX infections. This clustering produces 33 group clusters. All Group A infections end up in the dominant group cluster due to using a template similar to those used by the dominant group. This seems correct as most email addresses in the dominant group follow patterns such as "rambler+[0-9a-z]{4,5}@onionmail.org" and "recmydb+[0-9a-z]{4,5}@onionmail.org", which seem generated by the same software that infects the honeypots with addresses "dzen+[0-9a-z]{4,5}@onionmail.org". One of the templates of Group B is merged by note similarity into the CID-13549 group. The other Group B template forms a new group that did not exist before. These results point to group A being the dominant group and group B being CID-13549. They also show that our clustering may under-cluster as it failed to merge the two Group B templates into the same group.

C. Bitcoin Attribution

There are 11 groups with at least one Bitcoin address that has received deposits. These groups can be attributed by tracing how their revenue moves towards exchanges with KYC requirements that can be leveraged as attribution points. For this, we leverage WatchYourBack [35] to examine whether the coin flows from the seeds are sent to services with KYC requirements. LEAs can leverage those exchanges as attribution points by requesting the identity of the owner of the exchange address that receives payments from the victims. We ran a back-and-forth exploration of three steps starting from the 200 seeds, and using the provided tag database to identify addresses belonging to services. The exploration identified flows leading from the seeds of 3 groups to 12 services: 9 exchanges, two payment platforms, and a mixer. The dominant group uses 8 exchanges (Binance, ChangeNOW, Coinbase, CoinGate, KuCoin, LocalBitcoins, Paxful, and TradeOgre), two payment platforms (Cash App and WebMoney), and a defunct mixer (ChipMixer). The TradeOgre exchange is also used by CID-12926. The other exchange is HitBTC, used by CID-1760. The exchanges and payment services are likely used for cashing out, and the mixer for obfuscation. All exchanges and payment services have KYC requirements except TradeOgre, which does not list KYC in its terms [92], and is reported to not enforce identity verification [93]. Thus, we find useful attribution points for two groups, including the dominant one.

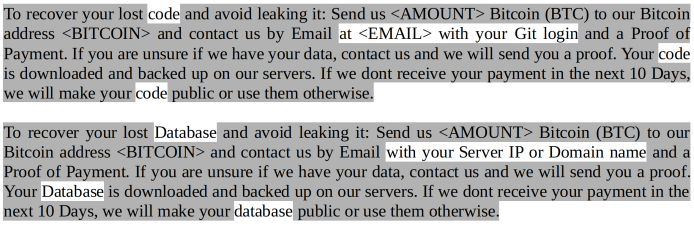


Figure 8: Similarity between the note used in the Git campaign (above) and the normalized text of a campaign from the dominant group (below).

Dominant group links. The dominant group also has flows into two addresses involved in other attack campaigns. One payment address of the dominant group[3] deposits parts of its revenue into an address[4] that belongs to a MIC associated to an attack on Git repositories. In May 2019, attackers logged into repositories hosted on GitLab, GitHub, and BitBucket using leaked credentials, deleted the repository contents, and left notes demanding a ransom to recover the data [34]. However, the campaign was not particularly effective, as the deleted data could be recovered from the repositories. Figure 8 shows that the ransom note used in the Git campaign is nearly identical to one of the database ransom attack campaigns.

The other flow is from a payment address[5] (P1) of the dominant group, that receives a payment of 0.00492860 BTC on April 13, 2021, and within a few hours sends that amount to an address[6] (C1) that has been reported in ransomware campaigns attributed to North Korea in an advisory from the U.S. Department of Defense (DoD) [95]. Those are the only two transactions of P1 and the withdrawal transaction to the DPRK-attributed address uses one input slot and one output slot. Furthermore, the DPRK-attributed C1 address has also been reported to accumulate victim payments from the ech0raix server ransomware that targets QNAP NAS devices [54].

Sending funds to an address belonging to another group does not conclusively mean both groups are the same, e.g., it could signal another kind of business relationship. Still, the similarity between the Git attacks and the database ransom attacks we study and the fact that the ransom notes are nearly identical, makes us think the most plausible explanation is that the dominant group was also responsible for the Git attacks. The discovered link between the dominant group and the DPRK may indicate that North Korea is involved in database ransom attacks. It could even point to the DPRK being responsible for the dominant group. However, there exist alternative explanations such as North Korea having some kind of business relationship with the dominant group or the DoD attribution being incorrect.

VIII. Discussion

We discuss takeaways, limitations, avenues for future work, ethics, and reproducibility.

[3] 17rDr5mbXjLdegWDFuWd61Ymhwm54GjtNK
[4] 12fv8qUbbzrRZrTYSSkELeWPeg3Gy3qsZ7
[5] 1757buFTrykguUzQNQgUSrdoQuyEJEF6CW
[6] 1KmWW6LgdgykBBrSXrFu9kdoHZ95Fe9kQF

Data recovery. Paying a ransom is discouraged as it keeps the attacker's business model alive [98], [94]. However, if victims are really desperate to get their data back, e.g., no backup is available and the data value is larger than the ransom value, it becomes important to understand whether a victim who pays the ransom will recover the deleted data. For the data recovery to be successful two conditions need to be satisfied. First, the attacker should have access to the data, i.e., it should have exfiltrated the data prior to deletion. Furthermore, the attacker needs to be willing to return the data after payment. Infections of our honeypots only exfiltrated the first 10 rows of each table (Group A) or no data at all (Group B). Thus, those groups cannot return the data even after receiving the payment.

We cannot generalize our honeypot results to all groups. In fact, our approach identified two groups using Tor hidden services, with one having provided a dashboard with a list of the exfiltrated databases and their sizes. This site allowed visitors to buy the data, likely indicating that the group had exfiltrated the data. On the other hand, we find additional supporting evidence that victims are unlikely to recover their data in many cases. First, some victims perform post-mortem analysis on their server and network logs concluding (as we do from our honeypot) that data was not exfiltrated [89]. Second, we have searched for reports of database ransom attacks in abuse databases [11], [14] and using search engines. We have not found reports of victims who paid the ransom and got the deleted data back. In contrast, we found multiple reports of victims who paid the ransom and did not get the data back [103], [104], [105], [106], [107]. We also found reports of victims that tried to contact the attackers on the provided email receiving no response [100], [101], [99], [102], further pointing to attackers having no intention to return the data.

It is possible that many database ransom attacks are scams where data is simply deleted with no intention of returning it. It is important to send the message to victims that paying a ransom for deleted data does not make sense unless there is some proof that the deleted data can be recovered. Thus, prior to performing a ransom payment, victims should check their database and network logs to examine if data was leaked prior to deletion. Furthermore, they should contact the attacker through the provided email and request proof that they have the original data. If no contact address is provided or no answer is received, most likely the data cannot be recovered.

Why do these attacks happen? Our work shows that database ransom attacks are prevalent, have been ongoing since 2017, and are still generating infections and payments from victims. The root cause behind these attacks is the weak authentication configured in the databases. Possible reasons for the weak authentication are users willingly remove authentication, the reuse of (leaked) credentials across services, and installation procedures that create (or allow creating) default accounts with weak credentials. The first two are best addressed through user education, e.g., emphasizing to users that attackers proactively search for insecure databases, and that credential reuse should be avoided. In contrast, improving the default installation procedure is a responsibility of the database software publishers.

Two-thirds (67.3%) of infected servers in our dataset run Elasticsearch. This is surprising given that MySQL is more popular than Elasticsearch [23]. We use the Censys and Shodan

Internet scanning engines to check the prevalence of the two services in June 2024. The number of Internet-connected MySQL servers (Censys: 5.7M IPs, Shodan: 3.6M IPs) is two orders of magnitude larger than the number of Elasticsearch servers (Censys: 47K IPs, Shodan: 62K IPs). This indicates that weak authentication is two orders of magnitude higher in Elasticsearch servers compared to MySQL servers.

We investigate whether this difference may be due to the installation procedure. Since December 2010, MySQL ships with a "mysql_secure_installation" script that secures the installation including the creation of strong passwords for default accounts. Only 1.9% of the MySQL infected servers in our dataset run a version that does not include the script. The script is an optional component of the installation process and it is not automatically executed. The documentation clearly states that is up to the user to run the script after finalizing the installation. Users that forget, or fail [108], to execute the script leave their database with weak default credentials. In contrast, Elasticsearch bundles together security features including the use of authentication and TLS to secure connections and the creation of strong passwords. The security bundle was enabled by default from Elasticsearch 8.0, released in February 2022. Prior to this change, default Elasticsearch installations had no authentication. If the user did not explicitly enable the security bundle and the server's IP address was publicly reachable, the databases were publicly accessible on the Internet and vulnerable to database ransom attacks.

LeakIX data shows that 89% of the infected Elasticsearch servers run a version older than 8.0. Thus, infections concentrate on Elasticsearch servers with insecure default installations, which have not been secured by the user. Surprisingly, 11% of the infections affect Elasticsearch versions with strong authentication by default, indicating the responsibility lies on the user configuration, e.g., changing the password to a weaker one or creating a new account with weak credentials. We query 6K randomly sampled IPs running Elasticsearch in Shodan to obtain their version. Of those, 4,909 do not show a version number and 90 show a snapshot ID. Of the servers with a valid version, only 7.6% run 8.x versions and the remaining 92.4% run older versions. Despite Elasticsearch versions with secure default installation having been available for 2.5 years, they are only used by a small fraction of Elasticsearch servers. The slow adoption of up-to-date database software is likely the root cause of the high prevalence of weak authentication in Elasticsearch servers.

Dominant group. Our approach uncovers a dominant group behind database ransom attacks, responsible for 76% of the infected servers and over 90% of the measured revenue. The existence of such group was not previously reported nor it can be guessed from the individual infection events in LeakIX. Previous work has shown other instances of dominant actors in cybercriminal activities, e.g., in illegal Dark Web markets [86]. Beyond the three examined databases, reports show the dominant group also targets other databases such as MongoDB [30], [17] and PostgreSQL [19]. Furthermore, we identify links between the dominant group and an attack on Git repositories, as well as with North Korea. While we cannot conclusively determine that North Korea is behind the dominant group and the Git repositories attacks, it is worth noting that this country has been reported to be involved in a variety of cybercriminal activities including ransomware, cryptocurrency thefts, ATM cash-out schemes, and DDoS [18]. Our link between the dominant group and North Korea is based on a previously published attribution of DPRK's addresses [95], highlighting the importance of sharing attribution results.

Evasion. Attackers actively try to evade detection and analysis. We observe two main evasion techniques. First, some campaigns do not provide the ransom payment details in their notes, instead providing Tor onion addresses or contact emails to obtain them. Such indirection complicates the collection of Bitcoin addresses, hindering revenue estimation and attribution approaches relying on them. However, it may be possible to obtain the Bitcoin addresses through scambaiting techniques [59], [27] such as automating the submission of identifiers to Tor hidden services and the sending of initial contact emails. Second, attackers try to hamper IOC extraction by removing spaces and punctuation before or after the IOCs. Such evasion does not work for blockchain and onion addresses since the IOC extraction tool used verifies the embedded checksum. However, it introduces errors for IOCs such as emails and URLs. As attackers become aware of our approach they may incorporate other evasions. For example, they could minimize the reuse of IOCs across campaigns. However, avoiding IOC reuse does not come for free as it may require automated approaches to handle large numbers of IOCs. Attackers could also copy ransom note templates from other groups to try to make our approach misattribute them. We have designed our campaign identification to be conservative and only group templates with minor modifications, avoiding grouping templates from different groups that may be similar by chance or that may be based on another group's template but with modifications. However, if attackers copy exactly the same template used by another group, replacing only the indicators (e.g., Bitcoin address and contact email) with their own, then our campaign identification would incorrectly merge both groups.

Lack of ground truth. Similar to most works that analyze cybercrime, we do not have ground truth to evaluate the correctness of our results. We rely on manual analysis to validate the accuracy of our approach. An analyst with over 15 years of experience has thoroughly analyzed the produced clusters without finding any false positives where unrelated notes had been merged. Evaluating the recall (i.e., false negatives) is harder because it is possible that campaigns and indicators that look unrelated indeed belong to the same group. What we observe is that in some cases our NS clustering is not able to link notes that we believe are part of the same campaign, e.g., notes in Chinese where lists of deleted databases have not been detected. Thus, we believe the real number of campaigns may be smaller than 91.

LeakIX data. Our longitudinal analysis is constrained by the data LeakIX collects. The main limitation is that LeakIX only observes a subset of all database ransom attacks First, a database server may be shut down by the attackers after infection or may be infected and cleaned between two consecutive LeakIX scans. In addition, LeakIX identifies infections through regular expressions applied to database names, which could miss infections using previously unknown names. Moreover, LeakIX only collects ransom notes from unauthen-

ticated Elasticsearch, MySQL, and MariaDB database servers. Thus, it misses servers using other database software (e.g., MongoDB, Oracle, PostgreSQL), as well as databases with some, but weak, authentication, which may have been compromised through credential guessing. Furthermore, Internet scanning engines may miss 5% of services due to blocking and transient losses [26]. Thus, the observed 60,427 infected database servers is a conservative lower bound on the number of infections during the three years analyzed. Gomez et al. [36] have recently shown for the DeadBolt server ransomware that out of 2,504 payment addresses they identified on the Bitcoin blockchain, only 66 (2.6%) were known to the Shodan and Censys Internet scanning engines, i.e., over 97% of infections were missed by those engines. This caused the revenue estimation using the data of those engines to be 39 times smaller than the real revenue. Another limitation is that in the case of reinfections, LeakIX may only observe the last infection, unless attackers do not wipe the previous ransom note databases.

IPs versus infections. Throughout the paper, we use the number of IP addresses to approximate the number of infected database servers. The caveats are that the same server could appear multiple times in the dataset under different IP addresses and that the same IP address may be assigned to different servers over time. Both cases may happen for servers in cloud hosting services. The first case would introduce overestimation and the second underestimation. We expect the second case to be more common and thus believe the number of IP addresses may underestimate the number of victim servers.

Generalization. Our approach has been designed for database ransom attacks. However, it could be applied to other attacks such as the sextortion emails dataset used by Paquet-Clouston et al. [69] (not publicly available) to automate the manual cluster merging step in that work. It could also be applied for identifying addresses controlled by the same group in cryptocurrency abuse databases [11], [14], where victims submit written abuse reports. Furthermore, it could be used to cluster scam websites by analyzing the website text [72].

Ethics. We do not collect data from users or infected servers ourselves. Instead, we leverage data independently collected by the LeakIX scan engine. In particular, LeakIX collects data from database servers that have not set up authentication. From those servers, they collect the names of database tables and the table content for tables whose name matches their regular expressions for ransom notes names. Although LeakIX does not modify any data in the server, accessing unauthenticated servers, even in read-only mode, raises ethical concerns. We note that the authors of this work are not associated with LeakIX. The data we use is publicly offered by LeakIX to any user who registers a free account. Since the regular expressions used by LeakIX could have false positives, we apply a filtering step to eliminate any data that is not a ransom note. Thus, our dataset contains no sensitive user data. We believe that casting light on database ransom attacks justifies using the LeakIX data, and we hope that the publication of this manuscript can inspire discussion on the ethics of such data collection by Internet scanning services.

Reproducibility. We have released our clustering code and clustering results [24].

IX. Related Work

Ransomware. A wealth of research has investigated traditional ransomware that encrypts data in personal devices such as desktops and mobile phones [9], [67], [5], [44], [87], [52], [20], [41], [68]. Those works cover aspects such as detection [5], [44], analysis of victim payments [87], [52], [20], [41], [68], and ransomware's command-and-control (C&C) infrastructure [74], [88]. Some works have instead analyzed server ransomware strains that encrypt data stored in network-attached (NAS) servers [36]. Database ransom(ware) attacks can be considered an evolution of ransomware attacks. They are similar in requesting a ransom to recover data, but they delete data rather than encrypt it. Other differences include leveraging weak authentication, scanning for targets, and the absence of malware binaries.

Cybercrime Bitcoin abuse. Due to its privacy properties and poorly regulated legal status in some countries, the Bitcoin ecosystem has attracted cybercriminal operations such as ransomware [87], [52], [20], [74], [41], [68], [88], thefts [58], scams [69], [8], [58], [112], [7], [50], human trafficking [75], cryptojacking [90] hidden marketplaces [16], [48], [79], and money laundering [62]. In contrast, we investigate database ransom attacks.

Clustering. Many works automatically group similar malicious instances such as scam websites [25], [72], [50], scam emails [69], and malware samples [70], [71], [43], [40], [76], [51], [83], [31], [84], [65]. Most related is the work by Paquet-Clouston et al. [69] that clusters emails where the last 50 words are similar, identifies sextortion emails through keyword searches, and manually merges related email clusters. In contrast, our clustering automatically merges similar ransom notes and identifies campaigns run by the same threat group.

Attribution. Other research attributes malware and advanced persistent threats (APTs) [37]. Some approaches leverage stylometric code features to identify the code's author among a set of pre-defined ones [81], [12]. However, it is difficult to select features that uniquely capture the author's programming styles [4]. Other approaches start with a set of attributed binaries and attribute new samples by finding similarities to those. Such process can be performed manually by skillful analysts [56], using machine learning classifiers [80], [109], or through clustering [60], [82]. These approaches can handle malware developed by multiple authors (e.g., by threat groups), but can only attribute the malware to previously observed threat groups. The above techniques do not apply to database ransom attacks due to the absence of malware binaries. Instead, we propose a novel clustering to identify attacks from the same group and perform attribution by tracing the ransom payments until attribution points such as exchanges with KYC requirements [35].

X. Conclusions

We present the first systematic analysis of database ransom(ware) attacks. To address the question of who is behind these attacks we develop a novel clustering approach that groups the infections into campaigns, identifies campaigns run by the same threat group, and ranks threat groups according to a wealth of properties. Our clustering groups 60,427 database

server infections into 91 campaigns run by 32 groups, identifying a dominant group that is responsible for 76% of the infections and has generated 90% of the $498K revenue we measure across all groups. We also identify links from the dominant group to a previous campaign that compromised Git repositories, as well as to North Korea.

ACKNOWLEDGMENTS

We thank the reviewers for their time and valuable comments. We are especially grateful to LeakIX for making their scanning results publicly available. This work was partially funded by the Spanish Government MCIN/AEI/10.13039/501100011033/ through grants TED2021-132464B-I00 (PRODIGY), PID2022-142290OB-I00 (ESPADA), PRE2019-088472, and PREP2022-000165. Partial support was also provided by the Regional Government of Madrid through grant 2020-T2/TIC-20184. The above grants are co-funded by European Union ESF, EIE, FEDER, and NextGeneration funds.

REFERENCES

[1] L. Abrams, "MongoDB Apocalypse: Professional Ransomware Group Gets Involved, Infections Reach 28K Servers," January 2017, https://www.bleepingcomputer.com/news/security/mongodb-apocalypse-professional-ransomware-group-gets-involved-infections-reach-28k-servers/.

[2] ——, "Ongoing eCh0raix ransomware campaign targets QNAP NAS devices," June 2020, https://www.bleepingcomputer.com/news/security/ongoing-ech0raix-ransomware-campaign-targets-qnap-nas-devices/.

[3] ——, "New DeadBolt ransomware targets QNAP devices, asks 50 BTC for master key," January 2022, https://www.bleepingcomputer.com/news/security/new-deadbolt-ransomware-targets-qnap-devices-asks-50-btc-for-master-key/.

[4] S. Alrabaee, N. Saleem, S. Preda, L. Wang, and M. Debbabi, "OBA2: An Onion approach to Binary code Authorship Attribution," *Digital Investigation*, vol. 11, pp. S94–S103, 2014.

[5] N. Andronio, S. Zanero, and F. Maggi, "Heldroid: Dissecting and Detecting Mobile Ransomware," in *Recent Advances in Intrusion Detection*, 2015.

[6] E. Androulaki, G. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating User Privacy in Bitcoin," in *Financial Cryptography and Data Security*, 2013.

[7] M. Bartoletti, S. Lande, A. Loddo, L. Pompianu, and S. Serusi, "Cryptocurrency Scams: Analysis and Perspectives," *IEEE Access*, vol. 9, pp. 148 353–148 373, 2021.

[8] M. Bartoletti, B. Pes, and S. Serusi, "Data Mining for Detecting Bitcoin Ponzi Schemes," in *Crypto Valley Conference on Blockchain Technology*, June 2018.

[9] E. Berrueta, D. Morato, E. Magaña, and M. Izal, "A Survey on Detection Techniques for Cryptographic Ransomware," *IEEE Access*, vol. 7, pp. 144 925–144 944, 2019.

[10] A. Biryukov, I. Pustogarov, and R. Weinmann, "Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization," in *IEEE Symposium on Security and Privacy*, 2013.

[11] "Bitcoin abuse," 2023, https://www.bitcoinabuse.com.

[12] A. Caliskan-Islam, R. Harang, A. Liu, A. Narayanan, C. Voss, F. Yamaguchi, and R. Greenstadt, "De-anonymizing Programmers via Code Stylometry," in *USENIX Security Symposium*, 2015.

[13] "Censys," 2022, https://censys.io/.

[14] "Chainabuse," 2023, https://www.chainabuse.com/.

[15] M. Charikar, "Similarity Estimation Techniques from Rounding Algorithms," in *Annual ACM Symposium on Theory of Computing*, 2002.

[16] N. Christin, "Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace," in *The Web Conference*, 2013.

[17] "Broken water - a story about database being hacked," August 2023, https://xlog.zwh.moe/database-hacked?locale=en.

[18] CISA, "North Korea Cyber Threat Overview and Advisories," 2024, https://www.cisa.gov/topics/cyber-threats-and-advisories/advanced-persistent-threats/north-korea.

[19] O. Cohen, "Analysis: A Ransomware Attack on a PostgreSQL Database," October 2023, https://www.imperva.com/blog/postgresql-database-ransomware-analysis/.

[20] M. Conti, A. Gangwal, and S. Ruj, "On the economic significance of ransomware campaigns: A bitcoin transactions perspective," *Computers & Security*, vol. 79, pp. 162–189, 2018.

[21] G. Corfield, "'Malwareless' ransomware campaing operators pwned 83k victims' MYSQL servers, 250k databases up for sale," December 2020, https://www.theregister.com/2020/12/10/mysql_malwareless_ransomware/.

[22] Cyble, "85,000 MySQL Databases On Sale in Darkweb," December 2020, https://cyble.com/blog/85000-mysql-databases-on-sale-in-darkweb/.

[23] "DB-Engines Ranking," 2024, https://db-engines.com/en/ranking.

[24] "DB Ransom," 2024, https://github.com/KevinLiebergen/dbransom.

[25] J. Drew and T. Moore, "Automatic Identification of Replicated Criminal Websites Using Combined Clustering," in *IEEE Security and Privacy Workshops*, 2014.

[26] Z. Durumeric, D. Adrian, P. Stephens, E. Wustrow, and J. A. Halderman, "Ten Years of ZMap," *arXiv preprint arXiv:2406.15585*, 2024.

[27] M. Edwards, C. Peersman, and A. Rashid, "Scamming the Scammers: Towards Automatic Detection of Persuasion in Advance Fee Frauds," in *International Conference on World Wide Web Companion*, 2017.

[28] Europol, "Multi-million euro cryptocurrency laundering service bestmixer.io taken down," May 2019, https://www.europol.europa.eu/newsroom/news/multi-million-euro-cryptocurrency-laundering-service-bestmixerio-taken-down.

[29] ——, "Law enforcement disrupt world's biggest ransomware operation," February 2024, https://www.europol.europa.eu/media-press/newsroom/news/law-enforcement-disrupt-worlds-biggest-ransomware-operation.

[30] FamilyRon, November 2020, https://www.reddit.com/r/mongodb/comments/jqxv2o/my_mongodb_server_hacked/?rdt=35623.

[31] H. Faridi, S. Srinivasagopalan, and R. Verma, "Performance Evaluation of Features and Clustering Algorithms for Malware," in *IEEE International Conference on Data Mining Workshops*, 2018.

[32] H. Gao, Y. Yang, K. Bu, Y. Chen, D. Downey, K. Lee, and A. N. Choudhary, "Spam Ain't As Diverse As It Seems: Throttling OSN Spam With Templates Underneath," in *Annual Computer Security Applications Conference*, 2014.

[33] S. Gatlan, "Qlocker ransomware returns to target QNAP NAS devices worldwide," January 2022, https://www.bleepingcomputer.com/news/security/qlocker-ransomware-returns-to-target-qnap-nas-devices-worldwide/.

[34] GitLab, GitHub, BitBucket, "Git ransom campaign incident report," May 2019, https://about.gitlab.com/blog/2019/05/14/git-ransom-campaign-incident-report-atlassian-bitbucket-github-gitlab/.

[35] G. Gomez, P. Moreno-Sanchez, and J. Caballero, "Watch Your Back: Identifying Cybercrime Financial Relationships in Bitcoin through Back-and-Forth Exploration," in *ACM SIGSAC Conference on Computer and Communications Security*, 2022.

[36] G. Gomez, K. van Liebergen, and J. Caballero, "Cybercrime Bitcoin Revenue Estimations: Quantifying the Impact of Methodology and Coverage," in *ACM SIGSAC Conference on Computer and Communications Security*, 2023.

[37] J. Gray, D. Sgandurra, L. Cavallaro, and J. Blasco Alis, "Identifying Authorship in Malicious Binaries: Features, Challenges & Datasets," *ACM Computing Surveys*, vol. 56, no. 8, pp. 1–36, 2024.

[38] O. Harpaz, "Ransomware Devastating MySQL Servers," 2020, https://www.akamai.com/blog/security/please-read-me-opportunistic-ransomware-devastating-mysql-servers.

[39] B. Hou, Z. Cai, K. Wu, T. Yang, and T. Zhou, "6Scan: A High-Efficiency Dynamic Internet-Wide IPv6 Scanner With Regional En-

coding," *IEEE/ACM Transactions on Networking*, vol. 31, no. 4, pp. 1870–1885, August 2023.

[40] X. Hu, K. G. Shin, S. Bhatkar, and K. Griffin, "MutantX-S: Scalable Malware Clustering Based on Static Features," in *USENIX Annual Technical Conference*, 2013.

[41] D. Y. Huang, M. M. Aliapoulios, V. G. Li, L. Invernizzi, E. Bursztein, K. McRoberts, J. Levin, K. Levchenko, A. C. Snoeren, and D. McCoy, "Tracking Ransomware End-to-end," in *IEEE Symposium on Security and Privacy*, 2018.

[42] "iocsearcher," 2023, https://github.com/malicialab/iocsearcher.

[43] J. Jang, D. Brumley, and S. Venkataraman, "BitShred: Feature Hashing Malware for Scalable Triage and Semantic Analysis," in *ACM SIGSAC Conference on Computer and Communications Security*, 2011.

[44] A. Kharraz, S. Arshad, C. Mulliner, W. K. Robertson, and E. Kirda, "UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware," in *USENIX Security Symposium*, 2016.

[45] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage, "On the Spam Campaign Trail," in *USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2008.

[46] "langdetect," 2023, https://pypi.org/project/langdetect/.

[47] "LeakIX," 2022, https://leakix.net/.

[48] S. Lee, C. Yoon, H. Kang, Y. Kim, Y. Kim, D. Han, S. Son, and S. Shin, "Cybercriminal Minds: An investigative study of cryptocurrency abuses in the Dark Web," in *Network and Distributed System Security Symposium*, 2019.

[49] X. Li, B. Liu, X. Zheng, H. Duan, Q. Li, and Y. Huang, "Fast IPv6 Network Periphery Discovery and Security Implications," in *Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2021.

[50] X. Li, A. Yepuri, and N. Nikiforakis, "Double and Nothing: Understanding and Detecting Cryptocurrency Giveaway Scams," in *Network and Distributed System Security Symposium*, 2023.

[51] Y. Li, S. C. Sundaramurthy, A. G. Bardas, X. Ou, D. Caragea, X. Hu, and J. Jang, "Experimental Study of Fuzzy Hashing in Malware Clustering Analysis," in *Workshop on Cyber Security Experimentation and Test*, 2015.

[52] K. Liao, Z. Zhao, A. Doupé, and G. Ahn, "Behind Closed Doors: Measurement and Analysis of CryptoLocker Ransoms in Bitcoin," in *APWG Symposium on Electronic Crime Research*, 2016.

[53] "Limenet," 2024, https://limenet.io.

[54] Malware.News, "Story of the week: Ransomware on the darkweb," July 2021, https://malware.news/t/w2-july-en-story-of-the-week-ransomware-on-the-darkweb/50791.

[55] G. S. Manku, A. Jain, and A. D. Sarma, "Detecting near-duplicates for web crawling," in *The Web Conference*, 2007.

[56] M. Marquis-Boire, M. Marschalek, and C. Guarnieri, "Big game hunting: The peculiarities in nation-state malware research," 2015.

[57] S. Matic, P. Kotzias, and J. Caballero, "CARONTE: Detecting Location Leaks for Deanonymizing Tor Hidden Services," in *ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[58] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A Fistful of Bitcoins: Characterizing Payments Among Men with No Names," in *Internet Measurement Conference*, 2013.

[59] N. Miramirkhani, O. Starov, and N. Nikiforakis, "Dial One for Scam: A Large-Scale Analysis of Technical Support Scams," in *Network and Distributed System Security Symposium*, 2016.

[60] O. Mirzaei, R. Vasilenko, E. Kirda, L. Lu, and A. Kharraz, "SCRUTINIZER: Detecting Code Reuse in Malware via Decompilation and Machine Learning," in *Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2021.

[61] MITRE, "TeamTNT," 2024, https://attack.mitre.org/groups/G0139/.

[62] M. Möser, R. Böhme, and D. Breuker, "An Inquiry into Money Laundering Tools in the Bitcoin Ecosystem," in *APWG eCrime Researchers Summit*, 2013.

[63] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008, https://bitcoin.org/bitcoin.pdf.

[64] L. O'Donnell, "Please read me Ransomware Attacks 85K MySQL Servers," December 2020, https://threatpost.com/please_read_me-ransomware-mysql-servers/162136.

[65] J. Oliver, M. Ali, and J. Hagen, "HAC-T and Fast Search for Similarity in Security," in *IEEE International Conference on Omni-layer Intelligent Systems*, 2020.

[66] L. Øverlier and P. F. Syverson, "Locating Hidden Servers," in *IEEE Symposium on Security and Privacy*, 2006.

[67] H. Oz, A. Aris, A. Levi, and A. S. Uluagac, "A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions," *ACM Computing Surveys*, vol. 54, no. 11s, pp. 1–37, 2022.

[68] M. Paquet-Clouston, B. Haslhofer, and B. Dupont, "Ransomware Payments in the Bitcoin Ecosystem," *Journal of Cybersecurity*, vol. 5, no. 1, 2019.

[69] M. Paquet-Clouston, M. Romiti, B. Haslhofer, and T. Charvat, "Spams meet Cryptocurrencies: Sextortion in the Bitcoin Ecosystem," in *ACM Conference on Advances in Financial Technologies*, 2019.

[70] R. Perdisci, D. Ariu, and G. Giacinto, "Scalable Fine-Grained Behavioral Clustering of HTTP-Based Malware," in *Network and Distributed System Security Symposium*, 2009.

[71] R. Perdisci, W. Lee, and N. Feamster, "Behavioral Clustering of HTTP-Based Malware and Signature Generation Using Malicious Network Traces," in *USENIX Symposium on Networked Systems Design and Implementation*, 2010.

[72] R. Phillips and H. Wilder, "Tracing Cryptocurrency Scams: Clustering Replicated Advance-Fee and Phishing Websites," in *IEEE International Conference on Blockchain and Cryptocurrency*, 2020.

[73] A. Pitsillidis, K. Levchenko, C. Kreibich, C. Kanich, G. M. Voelker, V. Paxson, N. Weaver, and S. Savage, "Botnet Judo: Fighting Spam with Itself," in *Network and Distributed System Security Symposium*, 2010.

[74] S. Pletinckx, C. Trap, and C. Doerr, "Malware Coordination using the Blockchain: An Analysis of the Cerber Ransomware," in *IEEE Conference on Communications and Network Security*, 2018.

[75] R. S. Portnoff, D. Y. Huang, P. Doerfler, S. Afroz, and D. McCoy, "Backpage and Bitcoin: Uncovering Human Traffickers," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[76] M. Z. Rafique and J. Caballero, "FIRMA: Malware Clustering and Network Signature Generation with Mixed Network Behaviors," in *Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2013.

[77] F. Reid and M. Harrigan, "An Analysis of Anonymity in the Bitcoin System," in *IEEE PASSAT/SocialCom*, 2011.

[78] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Financial Cryptography and Data Security*, 2013.

[79] ——, "How Did Dread Pirate Roberts Acquire and Protect his Bitcoin Wealth?" in *Financial Cryptography and Data Security*, 2014.

[80] I. Rosenberg, G. Sicard, and E. David, "DeepAPT: Nation-State APT Attribution Using End-to-End Deep Neural Networks," in *International Conference on Artificial Neural Networks*, 2017.

[81] N. Rosenblum, X. Zhu, and B. P. Miller, "Who Wrote This Code? Identifying the Authors of Program Binaries," in *European Symposium on Research in Computer Security*, 2011.

[82] A. Saha, J. Blasco, L. Cavallaro, and M. Lindorfer, "ADAPT it! Automating APT Campaign and Group Attribution by Leveraging and Linking Heterogeneous Files," in *Recent Advances in Intrusion Detection*, 2024.

[83] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero, "AVClass: A Tool for Massive Malware Labeling," in *Recent Advances in Intrusion Detection*, 2016.

[84] S. Sebastián and J. Caballero, "AVClass2: Massive Malware Tag Extraction from AV Labels," in *Annual Computer Security Applications Conference*, 2020.

[85] "Shodan," 2022, https://www.shodan.io/.

[86] K. Soska and N. Christin, "Measuring the longitudinal evolution of the online anonymous marketplace ecosystem," in *USENIX Security Symposium*, 2015.

[87] M. Spagnuolo, F. Maggi, and S. Zanero, "BitIodine: Extracting Intelligence from the Bitcoin Network," in *Financial Cryptography and Data Security*, 2014.

[88] T. Taniguchi, H. Griffioen, and C. Doerr, "Analysis and Takeover of the Bitcoin-Coordinated Pony Malware," in *ACM Asia Conference on Computer and Communications Security*, 2021.

[89] B. Team, "Help! My database was compromised!" January 2024, https://www.border0.com/blogs/help-my-postgres-database-was-compromised.

[90] E. Tekiner, A. Acar, A. S. Uluagac, E. Kirda, and A. A. Selçuk, "SoK: Cryptojacking Malware," in *IEEE European Symposium on Security and Privacy*, 2021.

[91] B. Toulas, "Hundreds of Elasticsearch databases targeted in ransom attacks," June 2022, https://www.bleepingcomputer.com/news/security/hundreds-of-elasticsearch-databases-targeted-in-ransom-attacks/.

[92] TradeOgre, "Terms," 2024, https://tradeogre.com/terms.

[93] "Tradeogre kyc level," 2024, https://kycnot.me/service/tradeogre.

[94] UK Government, "UK and Singapore secure agreement against ransomware payments," November 2023, https://www.gov.uk/government/news/uk-and-singapore-secure-agreement-against-ransomware-payments.

[95] U.S. Department of Defense, "StopRansomware: Ransomware Attacks on Critical Infrastructure Fund DPRK Malicious Cyber Activities," 2023, https://media.defense.gov/2023/Feb/09/2003159161/-1/-1/0/CSA_RANSOMWARE_ATTACKS_ON_CI_FUND_DPRK_ACTIVITIES.PDF.

[96] U.S. Department of Justice, "Russian national and bitcoin exchange charged in 21-count indictment for operating alleged international money laundering scheme and allegedly laundering funds from hack of mt. gox," July 2017, https://www.justice.gov/usao-ndca/pr/russian-national-and-bitcoin-exchange-charged-21-count-indictment-operating-alleged.

[97] U.S. Department of the Treasury, "OFAC Sanctions List," 2024, https://sanctionssearch.ofac.treas.gov/.

[98] U.S. Federal Bureau of Investigation, "How to protect your networks from ransomware," 2024, https://www.fbi.gov/file-repository/ransomware-prevention-and-response-for-cisos.pdf.

[99] "Victim not response 3," April 2019, https://www.chainabuse.com/report/1c201c87-adb6-4e5f-8f46-0fd2b13d7244.

[100] "Victim not response 1," April 2019, https://www.chainabuse.com/report/a7e2b175-be6d-4b2e-86d6-0efe59f93627.

[101] "Victim not response 2," May 2019, https://www.chainabuse.com/report/d1015ac9-6ab8-483c-910e-782f223f897c.

[102] "Victim not response 4," May 2019, https://www.chainabuse.com/report/5c0f3a61-5e84-4460-a7a5-957e9e81a99c.

[103] "Victim Pay 1," March 2022, https://www.chainabuse.com/report/f685b874-59e9-4838-9491-3a28a3a30e4e.

[104] "Victim Pay 2," August 2022, https://www.chainabuse.com/report/822d1adf-6141-40a6-926a-b4b6b49c6bed.

[105] "Victim Pay 3," March 2022, https://www.chainabuse.com/report/95ff2474-4b84-4906-835c-564e19eec08f.

[106] "Victim Pay 4," June 2019, https://www.chainabuse.com/report/73918841-3933-4b55-b4d0-293436299f4c.

[107] "Victim Pay 5," June 2019, https://www.chainabuse.com/report/7ee05fc8-1ae1-47a1-8308-7693abc2d157.

[108] H. Virdó and M. Drake, "How to install mysql on ubuntu 20.04," July 2022, https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04.

[109] Q. Wang, H. Yan, and Z. Han, "Explainable APT Attribution for Malware using NLP Techniques," in *IEEE International Conference on Software Quality, Reliability and Security*, 2021.

[110] Z. Whittaker, "Mindbody-owned fitmetrix exposed millions of user records — thanks to servers without passwords," October 2018, https://techcrunch.com/2018/10/11/fitmetrix-mindbody-data-exposed-password/?guccounter=1.

[111] "WatchYourBack," 2022, https://github.com/cybersec-code/watchyourback.

[112] P. Xia, H. Wang, B. Zhang, R. Ji, B. Gao, L. Wu, X. Luo, and G. Xu, "Characterizing Cryptocurrency Exchange Scams," *Computers & Security*, vol. 98, 2020.

[113] T. Yang, Z. Cai, B. Hou, and T. Zhou, "6forest: An ensemble learning-based approach to target generation for internet-wide ipv6 scanning," in *Internet Measurement Conference*, 2017.

APPENDIX

Element	Initial Prec.	Initial Recall	Initial F1	Final Prec.	Final Recall	Final F1
amount	1.000	1.000	1.000	1.000	1.000	1.000
bitcoin	1.000	0.999	0.999	1.000	0.999	0.999
email	0.888	0.887	0.887	**0.999**	**0.999**	**0.999**
list	1.000	0.993	0.996	1.000	0.993	0.996
monero	1.000	1.000	1.000	1.000	1.000	1.000
onionAddress	1.000	1.000	1.000	1.000	1.000	1.000
url	0.894	0.808	0.849	**1.000**	**1.000**	**1.000**
uuid	1.000	1.000	1.000	1.000	1.000	1.000
token	1.000	0.999	0.999	**1.000**	**0.999**	**0.999**
ALL	0.942	0.915	0.928	1.000	0.999	0.999

Table VIII: Initial and final extraction accuracy.

A. IOC Extraction Evaluation

To evaluate the accuracy of the indicator extraction, we manually created a ground truth (GT) that specifies, for a subset of 5,792 ransom notes, the start offset and value of the indicators in the note. If a note contains the same indicator multiple times, each indicator instance is added with its own start offset. In total, the GT contains 34,529 indicator instances. Table VIII shows on the left the initial indicator extraction accuracy. The overall F1 score is 0.928, with the lowest F1 score being for URLs (0.849) and email addresses (0.887). Their lower F1 score is due to an obfuscation, where attackers remove the spaces before and after an indicator to hamper indicator extraction tools. This can introduce false negatives and also false positives if two indicators appear next to each other. More specifically, two prevalent cases introduce most errors in URLs and emails. The first is that an email address follows a Bitcoin address without space, so *iocsearcher* only extracts an email address with the Bitcoin address as part of the username. To handle this case, we re-apply *iocsearcher* on the extracted email addresses and if a Bitcoin address is found inside, we remove it from the email. The second case is two consecutive URLs being joined into a single URL due to the lack of space. To handle this case, if an extracted URL contains the string "http", we split it into two URLs. While these two fixes could potentially introduce errors, they do not because these cases are confined to notes from two prevalent campaigns. Table VIII on the right shows the final indicator extraction accuracy after applying these two fixes. The overall F1 score raises from 0.928 to 0.999 with the F1 score for emails being 0.999 and for URLs 1.0.

B. Revenue Estimation

Table IX shows the revenue estimations for the 9 groups that have received deposits. For each group, the table shows its cluster ID, its number of payment addresses, the period of activity of the group according to the transactions, and two revenue estimations. The *DD-DC* lower bound estimation shows the number of seeds receiving deposits, the BTC received, and its value in dollars. The tighter *DD-OW+MI-DC* estimation

		Period		DD-DC			DD-OW+MI-DC			
CID	**Addr.**	**Start**	**End**	**Seeds**	**BTC**	**USD**	**MI Addr.**	**MICs**	**BTC**	**USD**
12373	305	2019-04-11	2024-04-09	154	22.65468725	410,459.23	209	137	27.15077189	449,922.25
12926	63	2023-06-08	2024-04-02	26	0.74087599	28,029.52	27	21	0.75387599	28,365.30
13390	16	2023-02-24	2023-07-16	10	0.29313614	7,715.98	10	10	0.29313614	7,715.98
318	2	2017-09-24	2018-09-09	2	0.26189800	1,682.93	7	2	0.40099263	2,617.54
6	2	2023-08-05	2023-08-21	1	0.07994339	2,322.12	1	1	0.07994339	2,322.12
3	2	2023-11-07	2023-12-03	1	0.05991624	2,181.39	1	1	0.05991624	2,181.39
123	1	2020-01-23	2023-10-27	1	0.08771142	1,759.54	1	1	0.08771142	1,759.54
42	2	2022-04-22	2023-03-14	2	0.05250000	1,704.25	2	2	0.05250000	1,704.25
7452	1	2022-03-12	2022-03-12	1	0.02000000	775.50	1	1	0.02000000	775.50
434	3	2016-12-31	2017-01-03	1	0.10459291	108.31	4	1	0.59794241	612.36
1760	1	2020-06-27	2020-07-01	1	0.02300600	210.50	1	1	0.02300600	210.50
Total	398	2016-12-31	2024-04-09	200	24.37826734	456,949.28	264	178	29.51979611	498,186.74

Table IX: Revenue of the clusters with seeds.

NSID	**CID**	**Lang.**	**IPs**	**Addr.**	**Start**	**End**	**Seeds**	**BTC**	**USD**
8133	12373	EN	5,991	33	2021-08-06	2023-09-28	28	4.65902095	141,373.87
11955	12373	EN	561	91	2019-04-11	2022-09-18	31	12.29866669	100,370.16
14220	12373	EN	16,565	16	2023-05-22	2024-04-09	16	2.17541702	72,184.59
10241	12373	CN	655	49	2021-03-31	2023-01-05	41	1.66348944	57,570.25
11389	12373	EN	9,366	41	2020-11-01	2023-08-14	16	1.25699545	34,963.76
13537	12926	EN	1,497	63	2023-06-08	2024-04-02	26	0.74087599	28,029.52
416	12373	CN	16	7	2021-05-12	2022-01-26	7	0.46157184	20,121.66
1781	12373	EN	2	2	2023-10-17	2023-12-10	2	0.38435216	14,269.83
763	12373	CN	38	9	2020-04-30	2020-11-14	6	0.88233597	10,220.58
9209	12373	CN	64	11	2020-12-02	2022-05-07	7	0.3227606	9,544.34
11145	12926	EN	17	6	2023-10-15	2023-12-14	6	0.25953293	9,440.21
14197	13390	CN	441	16	2023-02-24	2023-07-16	10	0.29313614	7,715.98
6797	12373	EN	770	9	2021-05-17	2023-03-29	5	0.18757327	6,586.88
530	12373	EN	89	18	2020-09-11	2021-01-14	5	0.37506335	5,489.74
460	12373	EN	71	2	2023-03-14	2023-06-26	2	0.11013006	2,886.07
14115	12373	EN	40	27	2019-09-05	2019-12-19	3	0.29955008	2,650.34
10552	12926	EN	1	1	2023-11-21	2023-12-14	1	0.0615834	2,363.17
6	6	EN	14,409	2	2023-08-05	2023-08-21	2	0.07994339	2,322.12
3	3	EN	16,645	2	2023-11-07	2023-12-03	1	0.05991624	2,181.39
123	123	EN	56	1	2020-01-23	2023-10-27	1	0.08771142	1,759.54

Table X: Top 20 campaigns by revenue using the DD-DC estimation. NSID is the campaign ID and CID is the group ID. Revenue from a Bitcoin address is counted in all campaigns where the address appears.

shows the number of addresses receiving deposits after MI expansion, the number of MICs found. the total BTC received, and its value in dollars.

The dominant group receives 90.3% of the total revenue in dollars with the second largest group receiving 5.7%. As recently shown by Gomez et al. [36], the revenue estimated is a lower bound of the actual revenue, which may be 39 times higher than reported, as discussed in the LeakIX data paragraph in Section VIII. The tighter estimation is 17% larger in the number of BTCs received by the operations, but just 9.4% larger when converted to dollars. The groups obtain an average of $43,449.6 and a median of $2,322.12 in revenues.

Table X details the top 20 campaigns by revenue using the tighther DD-DC estimation, which does not include the revenue of other addresses in the same MIC. We count the revenue of a Bitcoin address in all campaigns where it appears. Of the top 20 campaigns, 13 belong to the dominant group including all top 5 campaigns.