2018 IEEE Symposium on Security and Privacy

Tracking Ransomware End-to-end

Danny Yuxing Huang[1], Maxwell Matthaios Aliapoulios[2], Vector Guo Li[3]
Luca Invernizzi[4], Kylie McRoberts[4], Elie Bursztein[4], Jonathan Levin[5]
Kirill Levchenko[3], Alex C. Snoeren[3], Damon McCoy[2]

[1] Princeton University [2] New York University [3] University of California, San Diego [4] Google Inc [5] Chainalysis

***Abstract*—Ransomware is a type of malware that encrypts the files of infected hosts and demands payment, often in a cryptocurrency such as Bitcoin. In this paper, we create a measurement framework that we use to perform a large-scale, two-year, end-to-end measurement of ransomware payments, victims, and operators. By combining an array of data sources, including ransomware binaries, seed ransom payments, victim telemetry from infections, and a large database of Bitcoin addresses annotated with their owners, we sketch the outlines of this burgeoning ecosystem and associated third-party infrastructure. In particular, we trace the financial transactions, from the moment victims acquire bitcoins, to when ransomware operators cash them out. We find that many ransomware operators cashed out using BTC-e, a now-defunct Bitcoin exchange. In total we are able to track over $16 million in likely ransom payments made by 19,750 potential victims during a two-year period. While our study focuses on ransomware, our methods are potentially applicable to other cybercriminal operations that have similarly adopted Bitcoin as their payment channel.**

I. Introduction

Ransomware is a type of malware that encrypts a victim's documents and media, and then urges payment for their decryption. In its beginnings, ransoms were demanded via a collection of online cash-equivalent payment instruments, such as MoneyPak, Paysafecard, or UKash [1]. From the ransomware operators' perspective, these instruments have undesirable properties: their limited geographic availability shrinks the paying-victim pool, and they are run by companies subject to the local law, which might compel them into reversing transactions or tracking the ransom recipients.

To overcome these drawbacks, the operators of many major ransomware families have adopted Bitcoin. This cryptocurrency poses challenges to law enforcement, as it is decentralized, largely unregulated, and all parties in a transaction are hidden behind pseudo-anonymous identities. Moreover, all transactions are irreversible, and it is widely available for victims to purchase. Due to these properties, Bitcoin has also gained adoption as a payment method for other illicit activities, such as drug markets [2], online sex ads [3], and DDoS-for-hire services [4].

However, Bitcoin has a property that is undesirable to cybercriminals: all transactions are public by design. This enables researchers, through transaction clustering and tracing [5], [6], [7], to glean at the financial inner workings of entire cybercriminal operations. Before Bitcoin, these insights had to be only partial and infrequent, as they hinged on sporadic data leaks [8], [9], [10].

In this paper, we perform a large-scale, two-year measurement study of ransomware payments, victims, and operators. While prior studies have estimated the revenue for a single ransomware operation [6] or reverse engineered the technical inner works of particular ransomware binaries [11], [12], our study is the first to perform an end-to-end analysis of a large portion of the ransomware ecosystem, including its revenue, affiliate schemes, and infrastructure.

To do so, we combine multiple data sources, including labeled ransomware binaries, victims' ransom payments, victim telemetry (collected through an IP sinkhole we deploy), and a large database of Bitcoin addresses annotated with their owners (provided by Chainalysis[1]). This wealth of data allows us to follow the money trail from the moment a victim acquires bitcoins, to when the ransomware operators cash them out. In total, we establish a lower-bound estimate on ransom payments' volume of $16 million USD, made by 19,750 potential victims over two years.

The bitcoin-trail allows us to determine the likely geographic locations of paying victims, which we corroborated with the collected telemetry of a large ransomware campaign. We find that South Koreans likely paid over $2.5 million USD in ransoms to the Cerber ransomware family, which is 34% of the total Cerber's revenue we tracked. Our measurements indicate that South Koreans were also likely disproportionately impacted by other ransomware campaigns. This calls for further studies on why this region is disproportionately impacted, and what can be done to better protect it.

We also find that ransomware operators strongly preferred to cash out their bitcoins at BTC-e, a Russian Bitcoin exchange that converted bitcoins to fiat currencies. This exchange has now been seized.

Finally, we describe some unique ethical issues that we faced during our study and limit possible interventions against ransomware campaigns. For example, any disruption of the payment infrastructure can result in both the victim's inability to access their data and an increased financial burden, as ransom amounts increase with time in many families.

In summary, our main contributions are as follows. (1) We develop a set of methodologies that enable an end-to-end analysis of the ransomware ecosystem. (2) We conduct a two-year measurement study of the ecosystem, conservatively

[1]Chainalysis is a proprietary online tool that facilitates the tracking of Bitcoin transactions by annotating Bitcoin addresses with potential owners. See https://www.chainalysis.com/.

DOI 10.1109/SP.2018.00047

estimating that ransomware operators have collected over $16 million USD in ransoms from 19,750 potential victims. We also identify that BTC-e, a Russian-operated Bitcoin exchange (now seized by US law enforcement) appeared to be a key cash out point for ransomware operators. (3) We discuss possible intervention points, open challenges in ransomware measurement, and unique ethical issues specific to ransomware research.

II. Background

In this section, we describe the timeline of the archetypal ransomware infection, from malware delivery to ransom cash out.

Delivery: Ransomware is distributed through a variety of vectors, much like generic malware. For instance, Cerber and Locky were also spread via malicious email attachments [13], [14], while Karma partly relied on pay-per-install networks [15]. More recent families, such as WannaCry and NotPetya, exploited known vulnerabilities in network services to propagate within a LAN [16].

Execution: Once a ransomware binary executes on a host, it silently encrypts a set of files files deemed valuable to the user, such as documents and images. When the encryption completes, the ransomware displays a *ransom note* on the host's screen, informing the user that those files are held for ransom, payable in bitcoins.

Payment: A ransom note usually includes a guide on how to purchase bitcoins from *exchanges*, online services that facilitate the conversion between Bitcoin and fiat currencies. Exchanges come in different flavors: they can operate globally (e.g., Paxful), or regionally (e.g., Coinbase, which only caters to a US clientele). Most exchanges are centralized, except from a handful that facilitate direct transactions between buyers and sellers (e.g., LocalBitcoin).

Furthermore, ransom notes include either *ransom addresses*, Bitcoin wallets victims are expected to pay into, or a link to a payment website which displays this address. Many ransomware families (e.g., Locky and Cerber) generate a unique ransom address for each victim to automate the identification of paying victims, whereas others reuse addresses for multiple victims (e.g., WannaCry and CryptoDefense). When addresses are reused, ransomware operators cannot discriminate paying victims, so they either require the victim to send them the payment transaction hash (a verification mechanism susceptible to abuse, as all transactions are public), or they simply do not decrypt the victim's files (e.g., WannaCry [17]). Ransom amounts are typically fixed, denominated in US dollars (e.g, $1,000 for some Cerber strains [11]) or Bitcoins (e.g., 0.5 BTC for some Locky strains [18]). A notable exception is Spora [19], where the estimated value of each victim's files is factored into the ransom.

Decryption: Once the payment has been confirmed, ransomware either automatically decrypts the files held for ransom, or it instructs the victim to download and execute a decryption binary.

Liquidation: To cash out their proceeds, ransomware operators often deposit their bitcoins into a wallet controlled by an exchange to trade them for fiat currencies. As law enforcement agencies might compel exchanges into disclosing the identity of their clients, some operators first deposit their bitcoins into *mixers*, services that obfuscate bitcoin trails by intermixing bitcoin flows from multiple sources.

III. Discovering Ransom Deposit Addresses

The Bitcoin blockchain is a public sequence of timestamped transactions that involve wallet addresses, which are basically pseudo-anonymous identities. To discern transactions attributable to ransom campaigns, we design a methodology to trace known-victim payments (this section), cluster them with previously-unknown victims (Sections IV-A, and IV-B), estimate potentially missing payments (Section IV-E), filter transactions to discard the ones that are likely not attributable to ransom payments (Section IV-F). We show results of these methods in Section V, where we estimate the revenue of different ransomware families and characterize their financial activities.

Our payment tracking pipeline begins with methods to discover *seed* addresses — ransom addresses associated with a small number of known ransomware victims. Two sources provide us with a list of seed ransom addresses: real victims who reported ransomware infection [7], [6]; and our method of generating *synthetic* victims, where we extract ransom addresses by executing the ransomware binaries in a controlled environment — effectively becoming the infection "victims" ourselves.

A. Real Victims

To find real victims, we automatically scrape reports of ransomware infection in public forums, such as Bleeping Computer. These reports typically contain screenshots or excerpts of ransom notes, from which we extract the seed ransom addresses via text or image analysis. In addition, we obtain a list of seed ransom addresses from proprietary sources such as ID Ransomware, which maintains a record of ransomware victims and the associated ransom addresses [20].

B. Synthetic Victims

However, infection reports from real, paying victims are hard to come by. For example, we were initially unable to find any real victim infection reports which contained the ransom addresses for several families, such as Cerber and Locky.[2] To extend our coverage, we complement real victims with synthetic victims.

Using a technique that we will discuss in Section IV-E, we first obtain the binaries of Cerber and Locky from VirusTo-

[2]We selected Cerber and Locky based on media reports indicating that they were actively infecting large numbers of victims [13], [14].

tal.[3] We execute a subset of the ransomware binaries from each family on four independent platforms: VmRay [21], a hypervisor-based commercial sandbox; a VMware-based commercial sandbox; Cuckoo, an open-source sandbox that runs on VirtualBox virtual machines (VMs); and Windows XP on a bare-metal machine. We opt for these diverse platforms to mitigate potential anti-VM and anti-sandbox techniques in some variants of Cerber and Locky.

We execute each malware sample for up to twenty minutes, and then we collect the memory dump (in the case of VM executions), created files, and screenshots, from which to extract Bitcoin wallet addresses. We do not have any false positives, since Bitcoin wallet addresses have 32 bits of error-checking code. We also extract the visible text from screenshots through a commercial OCR provider [22], and process the output through the same extraction pipeline in order to obtain the ransom addresses.

C. Summary of Results

Using the method in Section III-A, we have collected 25 seed ransom addresses from actual victims across 8 ransomware families: CoinVault, CryptXXX, CryptoDefense, CryptoLocker, CryptoWall, Dharma, Spora, and WannaCry.

In addition, we apply the method in Section III-B and obtain 32 seed ransom addresses (8 of which from bare-metal infections) for synthetic Cerber infections, and 28 seed ransom addresses (3 of which from bare-metal infections) for synthetic Locky infections. As we will discuss in Section IV-B, anyone of Cerber's addresses was sufficient to discover the additional addresses that Cerber uses; likewise, anyone of Locky's addresses was sufficient.

IV. Discovering Additional Ransom Addresses

The seed ransom addresses obtained in Section III are associated with a small number of known victims, both real and synthetic. For families such as WannaCry, existing reports suggest that victims from multiple infections are shown the same ransom addresses [17]. Thus, the seed addresses themselves are sufficient for us to identify both known and potential victims,[4] and we can proceed with estimating the ransomware revenue.

In contrast, families such as Locky, Cerber, and Spora generate unique ransom addresses for every infection, a fact that is corroborated by prior research [19], vetted proprietary sources, and by our binary executions (Section III-B). On their own, these seed ransom addresses do not reveal any information about other potential victims. Any bitcoins received by each seed ransom address is likely associated with only a single victim that was provided with that seed ransom address.

[3]We also obtained binaries of the Sage ransomware, but as we will discussion in Section IV-B, micropayments to Sage's ransom addresses did not result in subsequent bitcoin transfers, so we excluded Sage from our analysis.

[4]We refer to any victims whom we do not know *a priori* as *potential* or *likely* victims (as opposed to *real* or *known* victims that we know from ground truth). Absent ground truth, we are uncertain that they are actual victims of ransomware infections.

Such unique ransom addresses motivate the need to expand our analysis beyond the seed addresses and identify additional addresses that are likely to be associated with ransomware activities. In this section, we describe a new method that is based on clustering and *micropayments* to discover additional ransom addresses.

A. Clustering by Co-spending

Even though we have a relatively small number of seed ransom addresses, we can infer payment activities of other potential victims by discovering wallet addresses that *co-spent* with the seed addresses. Two wallet addresses are known to be co-spent if they are used as the input to the same transaction. In non-CoinJoin transactions [23], we assume that an entity that creates a transaction has access to the private keys of all the input wallet addresses in the transaction [24], and that the entity is in control of all the input addresses. We call this assumption the *co-spending heuristic*, which we use to recursively look for addresses that co-spent with the seed ransom addresses, and also addresses that co-spent with the seed ransom addresses' co-spending addresses [7], [6], [5].

In this way, we construct a cluster of wallet addresses. Every address in the cluster, which we shall refer to as *cluster address*, is presumably under the control of the same ransomware family. These cluster addresses include the seed ransom addresses; ransom addresses to which *likely* victims made ransom payments (which we cannot validate as coming from actual victims absent ground-truth); and wallet addresses that a ransomware family uses for internal book-keeping (e.g., aggregating ransom payments). For a given family, if we have multiple seed ransom addresses, it is possible that each of them may be in a disjoint cluster (likely because no two addresses from different clusters were ever co-spent). Since we know that the seed ransom addresses all belong to the same family, we manually merge the disjoint clusters into a single cluster, which we subsequently refer to as the ransomware's cluster.

We stress that the clustering technique does not apply to CoinJoin transactions [23], which violate the co-spending heuristic. The sender of a CoinJoin transaction does not have access to the private keys of the input wallet addresses. Effectively, two addresses that are co-spent in the same CoinJoin transaction cannot be clustered together. To detect CoinJoin transactions in our clustering, we apply a set of heuristics [25] using BlockSci [26]. We find no CoinJoin transactions in our clusters, although there is still a possibility that the heuristics might have failed to detect some CoinJoin transactions. We will mitigate this problem in Section IV-F by proposing and evaluating filtering techniques.

B. Augmenting Clustering with Micropayments

The construction of clusters uses the co-spending heuristic, which requires that bitcoins are *spent* from the seed addresses. However, for synthetic victims whose ransom addresses are unique to individual victims (e.g., Cerber and Locky), the addresses are not associated with any Bitcoin payments. As such, there is no co-spending.

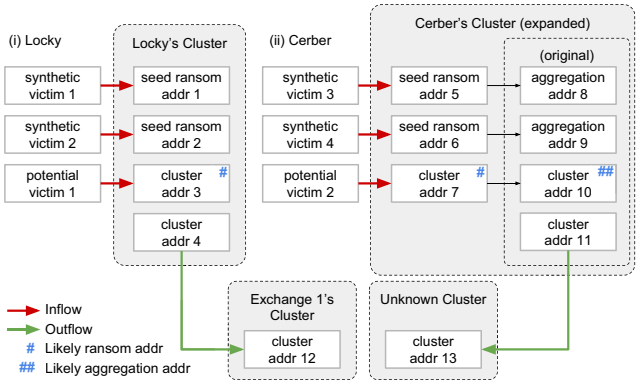


Fig. 1: A schematic illustration for transactions involving the Locky and Cerber clusters.

In order to observe co-spending, we develop a novel method — making 0.001 bitcoins worth of *micropayments* to the ransom addresses of synthetic victims and observe the subsequent flow of the bitcoins. We make micropayments only to seed addresses of Locky and Cerber, because at the time we started the analysis, we were unaware of any real victims of either ransomware families.

Locky: We generated 28 synthetic victims. We made a micropayment to each of the 28 seed ransom addresses.[5] What appears to be the ransomware operator later co-spent the ransom addresses with other wallet addresses, presumably in an attempt to aggregate ransom payments. All these 28 ransom addresses lead to the construction of a single Locky cluster with 7,093 addresses. In fact, if we had only made one of the 28 micropayments, we would have discovered the same cluster. In addition to the synthetic victims, we also discover a ransom address that belongs to a real Locky victim who paid 0.5 bitcoins of ransom, according to data we obtain from proprietary sources. This ransom address belongs to the same Locky cluster above, which validates our micropayment approach. We show an illustration of Locky's transaction graph in **Figure 1**(i), where we discover Addresses 3 and 4 in Locky's cluster by making micropayments to Addresses 1 and 2.

Cerber: We generated 32 synthetic victims. Again, we made a micropayment to each of the 32 ransom addresses. In each case, after the ransom address received our micropayment, what appears to be the ransomware operator moved our micropayment from the ransom address into a unique *aggregation* address. The ransom address was subsequently not used in any transactions. Also, it is never co-spent with any other wallet addresses. The aggregation address is used in exactly two transactions: (i) first receiving the micropayment from the ransom address, and (ii) sending the micropayment somewhere else by co-spending with other addresses, presumably to aggregate the ransom payments collected. We cluster the aggregation address to form Cerber's cluster. Similar to Locky's case, if we had only made any one of the 32 micropayments to Cerber, we would also have arrived at the same Cerber cluster.

[5]The timings of our micropayments, both for Locky and Cerber, do not follow any fixed schedule.

TABLE I: Ransomware clusters.

Family	*Cluster ID*	T_{min}	T_{max}	N_{addr}	N_{seed}
Cerber	1	2016-02-25	2017-08-31	8,526	32S
CoinVault	1	2012-11-03	2017-07-10	1,404	1R
CryptXXX	1	2016-05-11	2016-10-06	1,742	4R
CryptoDefense	1	2014-03-18	2014-08-15	1	1R*
	2	2014-02-28	2014-08-15	1	1R*
CryptoLocker	1	2013-09-07	2017-07-10	968	1R
	2	2016-03-07	2016-10-06	3,489	1R
CryptoWall	1	2015-11-24	2016-03-08	216	1R
	2	2014-05-06	2014-09-13	10	2R
	3	2015-01-14	2015-04-30	101	1R
	4	2015-11-03	2015-12-04	42	1R
	5	2014-04-29	2014-10-09	7	1R
	6	2014-05-25	2014-08-05	1	1R
Dharma	1	2016-05-13	2017-06-29	274	1R
	2	2017-01-23	2017-02-20	4	1R
Locky	1	2016-01-14	2017-06-02	7,093	1R 28S
Spora	1	2017-01-05	2017-08-26	2,126	1R
	2	2017-01-18	2017-03-01	24	1R
WannaCry	1	2017-05-12	2017-07-24	1	1R*
	2	2017-03-31	2017-07-03	3	2R*
	3	2017-05-12	2017-06-20	1	1R*
	4	2017-05-12	2017-08-09	1	1R*

As the ransom address and the corresponding aggregation address are never co-spent for each of the 32 micropayments, we cannot use the co-spending heuristic to cluster both types addresses. However, for each of our synthetic victims, the ransom address and the aggregation address were used exclusively to receive and send ransom payments. It is reasonable to assume that the ransomware operator is in control of both addresses. As such, we decide to expand the original cluster to also include the ransom addresses. We show an illustration of Cerber's transaction graph in **Figure 1**(ii), where aggregation Addresses 8 and 9 are clustered with Addresses 10 and 11 (i.e., original cluster). Since Addresses 5 and 6 appear to be exclusively used as an intermediary to receive and send ransom payments to the aggregation addresses, they are potentially under Cerber's control. As a result, we expand the cluster to include both addresses. In the rest of the paper, we always use Cerber's expanded cluster for analysis.

C. Clusters Identified

Using the seed ransom addresses that we obtained earlier, we construct clusters of wallet addresses for the 8 families in Section III-A and the 2 families in Section III-B. We present the result in **Table I**, where each row corresponds to a single cluster that we construct from one or multiple seed ransom addresses. We show the first date a cluster first received bitcoins as T_{min} and the last date as T_{max}, according to our observation on August 31, 2017. The number of seed ransom addresses is shown in the N_{seed} column; "R" denotes addresses from real victims, and "S" denotes synthetic victims. For example, Cluster 1 of Locky is constructed from one ransom address from a real victim and 28 ransom addresses from synthetic victims. By contrast, the two seed ransom addresses from CryptoLocker victims each belong to Clusters

TABLE II: Ransomware binaries in VirusTotal.

Family	*Seed*	*Expansion*	*Total*
Cerber	77,131	119,508	196,639
CryptoLocker	46,645	4,581	51,226
CryptoDefense	6,682	17,247	23,929
WannaCry	9,230	0	9,230
Locky	6,115	9	6,124
Cryptowall	51	4,344	4,395
CryptXXX	802	81	883
Spora	1,154	587	1,741
CoinVault	22	0	22
Dharma	5	0	5
Total	147,837	146,357	294,194

1 and 2 respectively. We manually merge both clusters and construct a single CryptoLocker cluster for later analysis.

The size of each cluster, in terms of the number of constituent addresses is shown in the N_{addr} column. For some families, such as Locky and Cerber (where we know each victim is assigned a unique ransom address), a small number of seed ransom addresses leads to the construction of clusters of thousands of addresses. For other families, each seed ransom address belongs to a cluster of size one. Examples include WannaCry and CryptoDefense, in which the same ransom addresses are known to be reused across victims and never co-spent with any other addresses. These reused ransom addresses are annotated with asterisks (*) in the table. As expected, families that generate individual addresses for each victim, such as Locky and Cerber, result in a larger number of additional addresses being discovered by our clustering than families that reused ransom addresses.

D. Limitations of Micropayments and Co-spending Heuristics

Not all micropayments resulted in subsequent bitcoin movements. For the Sage ransomware (which is not included in this study), we discovered two ransom addresses that real victims reported, but to which they did not make ransom payments. Similar to the case with synthetic victim, unpaid ransom addresses, even from real victims, cannot help us find the ransomware's cluster, because they have not been co-spent yet. To this end, we made micropayments to both of the Sage ransom addresses. However, the bitcoins have since remained in the addresses without being transferred to other wallet addresses. As a result, we are unable to find Sage's cluster. In general, ransomware operators may ignore micropayments, especially when the payment amount is below some minimum threshold. In addition, micropayments may even cause suspicion to the operators, although in our case Cerber and Locky continued to process our micropayments (as well as potential victim payments) throughout our analysis.

Furthermore, regardless of whether real victims made payments or we made micropayments, our clustering technique only discovers addresses that co-spent with the seed ransom addresses. It is possible that a ransomware operator may decide to switch to a completely different wallet cluster after our known victims, whether real or synthetic, have paid. Our technique will thus miss the new cluster. Additionally, it is also possible that a ransomware family may operate in the affiliate model [11], where each affiliate may choose to receive ransom payments in its own cluster that is disjoint from other affiliates' clusters. If the known payments are made to only a subset of the affiliates, our technique will again miss clusters of the other affiliates.

E. Coverage of Clusters

It is possible that we may not have discovered all clusters of addresses used by a particular ransomware family. For instance, a ransomware operator may have switched to a different cluster of wallet addresses, or there exists multiple operators for the same family, yet each operator uses a disjoint cluster of addresses. To determine if we have potentially missed clusters, we corroborate the timing of payment events on the blockchain against the timing of two external indicators of ransomware activity: relative number of searches according to Google Trends [27], and discovery of new binaries on VirusTotal [28]. Our assumption is that if, during some period, we discover new binaries for a ransomware family and/or observe relevant Google searches while we do not observing any incoming bitcoins into the ransomware's cluster, then we may have missed payment clusters. To this end, we measure the timing of the following three events:

Bitcoin inflow: We compute the total bitcoin amount in *inflows* to a ransomware cluster per day. An inflow to a ransomware cluster is a transaction that sends bitcoins from addresses outside the cluster to addresses inside the cluster. An inflow can come from multiple sources: e.g., real victims and synthetic victims (both from us and potentially from other researchers); or affiliates that pay for ransomware-as-a-service [11]. Presumably, the presence of any inflows to the cluster implies that the ransomware is actively in operation.

Google searches: Google Trends can produce an estimate for the relative number of searches per week for any user-specified search terms. For each ransomware family, we construct a search query by concatenating the name of the family along with the term "ransomware" and extract the relative number of searches from Google Trends. We assume that if a ransomware family is actively causing harm during some period, there would be some number of related searches — for instance, when victims look for help online.

Number of binaries on VirusTotal: The discovery of new binaries for a given ransomware family suggests a likely active ransomware operation, as new versions of the binary may be released, or the binary may undergo repacking. However, it is difficult to obtain binaries for a given ransomware family using VirusTotal. Even though VirusTotal provides a tag for each family, some of the tags are generic (e.g., `W32/Ransom`) and have little indication if a binary is related to a particular ransomware family.

To this end, we label a large dataset of ransomware binaries with its family (e.g., Cerber), its variant (e.g., Cerber's v2), and the date of its first upload to VirusTotal. We can use the dataset as a proxy to establish the timeline in which each ransomware family is active, thereby corroborating with

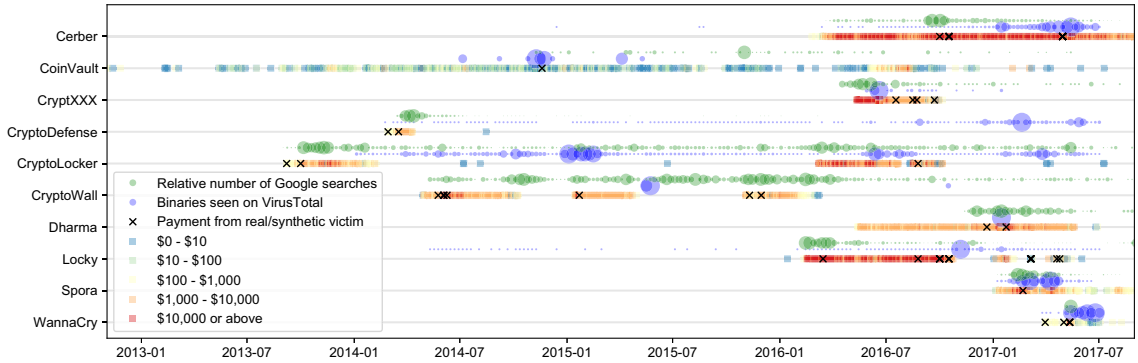


Fig. 2: A comparison of bitcoin inflows, binary discovery on VirusTotal, and Google searches for each ransomware family.

the timing of the ransom payments. (This labelled dataset also allows us to download and execute binaries of specific ransomware families, as we did in Section III.)

To build our dataset, we first sieve through all malware samples collected by VirusTotal [28] with 38 high-confidence YARA rules, which are Perl-based regular expressions often used to identify malware samples belonging to a specific family of malware. These rules have been crafted both by us and third-party researchers [29]. We call this our *seed* dataset: a high-confidence low-recall collection of binaries belonging to ransomware families. Using these YARA rules we identified 147,837 unique ransomware samples for ten ransomware families as shown in **Table II**.

To improve our recall and capture additional ransomware variants, we also collect a low-confidence dataset of malware that has been tagged by anti-virus engines as ransomware, either by generic tags (e.g., `W32/Ransom`), or family-specific tags. We then use the VxClass [30] to compute code-structure similarity scores based on BinDiff [31] across the binaries of the seed and low-confidence datasets. Finally, we leverage these similarity scores to cluster binaries into families. Specifically, we represent the binaries from both datasets as nodes in a graph, where two nodes are connected through an edge if the binary similarity of the two is high (over a manually-selected similarity of 0.9). We then identify the connected components in the graph, and we discard all clusters that do not contain seed binaries. After this expansion, the dataset was expanded to over 294k binaries, as shown in **Table II**. The expansion step was most effective for ransomware families with multiple variants, such as Cerber [32].

We validate the cluster purity by manually looking at execution screenshots (see Section III), checking that all screenshots belong to the same family. We also note that no generated cluster includes seeds from more than one ransomware family.

By using this labelled dataset, we count the number of new ransomware binaries as discovered on VirusTotal every day, based on both the seed and the expanded set of binaries.

Result of comparison: Using the three events above, we present this comparison in **Figure 2**, where the x-axis shows the period of our study from November 3, 2012 to August 31, 2017.[6] For each ransomware family on the y-axis, we show three types of events. On the gray horizontal line that corresponds to a ransomware family, we denote the daily bitcoin inflow with colored bands. As the color temperature increases from "cold" to "warm," the daily bitcoin values of inflows also increase. We have converted the bitcoin amounts into US Dollars based on the USD-Bitcoin exchange rate on the day the ransomware cluster received the bitcoins. Overlaid on top of the colored bands are X's that denote the time when a seed ransom address received payments from real or synthetic victims. Some of the X's may appear to overlap, as the payments may have occurred within a short period of time. As the clusters are constructed based on seed ransom addresses, by definition, the X's must appear within the colored bands. While the colored bands show inflows, the blue circles immediately above the bands indicate the discovery of new binaries on a given day for a ransomware family. The size of each circle denotes the relative number of binaries within the family. Finally, above the blue circles are the green circle, the sizes of which denote the relative number of Google searches.

This comparison offers a qualitative sanity check for both our binary classification and address clustering. To facilitate a quantitative comparison, we measure the *overlap* among three types of events: a ransomware cluster receiving bitcoins (B), VirusTotal detecting new binaries (V), and the ransomware family appearing in Google Trends (G). Two events overlap if both events occur in some 7-day period. We compute the conditional probability for these three types of events. Given a ransomware family, for instance, $\Pr[V \| B]$ is the probability of VirusTotal detecting at least a new binary (possibly due to polymorphism) in a random week, given that the cluster received some bitcoins during the same week. We show these conditional probabilities in **Table III**. Cerber, in particular, has the highest probability of overlap among all three events, whereas CryptoWall has one of the lowest.

A low $\Pr[V \| B]$ value, as is the case for CoinVault, CryptoWall, and Dharma, suggests that we are likely missing

[6] We do not apply Filter 1 (Section IV-F) to bitcoin inflows in the figure. If a given period has new binaries and/or Google searches but no bitcoin payments, it is difficult to visually distinguish whether we are missing payment clusters, or Filter 1 has removed the Bitcoin transactions.

TABLE III: Conditional probabilities of a ransomware cluster receiving bitcoins (B), VirusTotal detecting new binaries (V), and the ransomware family appearing in Google Trends (G) in any 7-day periods. All units are in percentages.

Family	$\Pr[V\|B]$	$\Pr[G\|B]$	$\Pr[B\|V]$	$\Pr[B\|G]$
Cerber	77.2	100.0	95.3	89.8
CoinVault	4.8	20.0	88.9	60.0
CryptXXX	56.5	91.3	81.3	44.7
CryptoDefense	87.5	87.5	4.8	4.8
CryptoLocker	63.0	100.0	19.0	25.5
CryptoWall	0.0	95.1	0.0	34.5
Dharma	1.8	51.8	100.0	67.4
Locky	94.3	98.1	50.5	57.8
Spora	76.7	100.0	95.8	71.4
WannaCry	75.0	68.8	92.3	42.3

binaries. This result is consistent with our binary classification results in Table II where we only discovered 22 CoinVault and 5 Dharma binaries (likely due to the lack of YARA rules). However, the correlation between Google Trends and payments ($\Pr[G\|B]$) is higher for all these families. A low $\Pr[B\|V]$ or $\Pr[B\|G]$ implies that we might be missing clusters; we expect that for a ransomware family, if we discover new binaries and/or observe relevant Google searches, the ransomware is likely to be receiving ransom payments from victims. It is likely that we are missing most ransom payment clusters for CryptoDefense, CryptoLocker, and CryptoWall. We also appear to be missing some payment clusters for most of the other families. This is likely caused by our limited number of seed addresses and the amount of co-spending each operator performs. However, it would be difficult to validate whether we are actually missing binaries or payment clusters absent ground-truth.

F. Filtering Transactions

As we alluded to in Section IV-A, an inflow of bitcoins to a ransomware cluster does not necessarily mean a real victim ransom payment. It could be, for instance, another researcher paying a synthetic victim or a completely non-ransomware-related CoinJoin transaction which our CoinJoin heuristics have failed to detect. Our goal is to examine inflows that come from potentially real victims and estimate the revenue that a ransomware family generates from ransom payments. To this end, we develop a number of inflow *filters*, which remove transactions from the inflows that are potentially unrelated to victims making ransom payments.

Filter 1: First, we create Filter 1, which identifies inflows that are consistent with known ransom payment patterns. There are two types of known patterns: (i) historically what ransom amounts are paid by real victims [7], [6]; and (ii) our novel method of identifying properties of the Bitcoin transaction graph for such historical payments.

As an example for Pattern (i), we observe, from screenshots online and executing the ransomware itself, that Locky demanded each victim pay a total ransom of $0.5n$ bitcoins (for some integer n) [18], and that CryptXXX charged a ransom of 1.2 bitcoins, 2.4 bitcoin, $500, or $1,000 per victim [33].

TABLE IV: An analysis of bitcoin inflows under different filters. V_1 offers a way to estimate the total revenue from ransom payments for a given family.

Family	V_0 *(k$)*	V_1 *(k$)*	$\frac{V_1}{V_0}$ *(%)*	V_2 *(k$)*	$\frac{V_2}{V_0}$ *(%)*	$\frac{V_3}{V_0}$ *(%)*
Cerber	7,702	7,678	99.7	3,990	51.8	51.7
CoinVault	198	20	10.3	18	9.3	1.3
CryptXXX	1,871	1,841	98.4	858	45.9	45.5
CryptoDefense	70	69	99.8	28	41.3	41.3
CryptoLocker	2,048	667	32.6	691	33.8	11.1
CryptoWall	1,214	244	20.2	529	43.6	9.2
Dharma	1,266	231	18.3	631	49.9	8.1
Locky	7,825	6,632	84.8	3,032	38.7	33.2
Spora	827	3	0.5	131	15.9	0.1
WannaCry	100	100	99.4	36	36.5	36.3

Using this pattern, if, for an inflow to a ransomware cluster, the total amount of bitcoins or US Dollars sent is consistent with the known ransom amounts above, we assume that the inflow is likely to be a ransom payment. The output address in the inflow is likely to be a ransom address; we call the output address a *likely ransom address*. As an illustration, if Address 3 in **Figure 1**(i) receives some increments of 0.5 bitcoins, then it is a likely ransom address for Potential Victim 1.

However, certain ransomware families do not have fixed ransom amounts, so we look for Pattern (ii). This pattern describes properties in the Bitcoin transaction graph that we observe in real/synthetic victim payments. For instance, any payments received by seed ransom addresses are potentially victim payments, especially for ransomware families where a single ransom address is used for multiple victims; inflows to seed ransom addresses therefore satisfy Filter 1. On the other hand, some families generate unique ransom addresses for each victim. For these families, we analyze how bitcoins move after known victims have paid. Using Cerber as an example, we observe that, after a synthetic victim transfers bitcoins into his ransom address, all the bitcoins are emptied into a unique aggregation address, which subsequently transfers all the bitcoins by co-spending with other wallet addresses.

We use **Figure 1**(ii) to illustrate how we use Pattern (ii) to identify a potential Cerber ransom address. Suppose Address 10, an address in Cerber's cluster, is used in two transactions: once receiving all bitcoins from some Address 7 (not in Cerber's cluster yet), before subsequently sending the bitcoins away by co-spending with other addresses. Suppose, further, that Address 7 is never co-spent, and that it sends all the received bitcoins into Address 10. Both of these observations are consistent with how bitcoins flow for all our synthetic victim payments. As such, we say that Address 10 is a likely aggregation address, and Address 7 is a likely ransom address to which some Potential Victim 2 made a ransom payment. We include Address 7 in the expanded cluster of Cerber. In general, our method of filtering by Pattern (ii) is not restricted to Cerber and can be applied to other ransomware families that have special properties in their Bitcoin transaction graphs.

We show the results of filtering in **Table IV**. For each family's cluster, we first apply Filter 0 as a baseline, which does not filter any inflows. Bitcoin amounts sent by any Filter 0 inflows are denoted as V_0, converted into US Dollars.

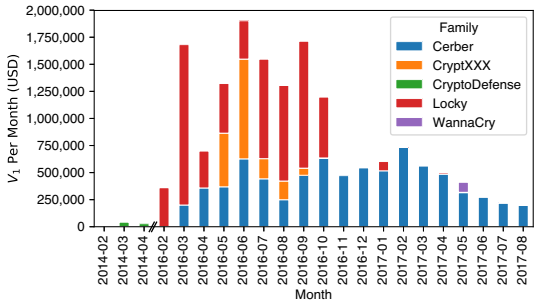


Fig. 3: Monthly V_1 values for each ransomware family.

We then apply Filter 1, and the resultant amounts of bitcoin inflows are shown in the V_1 column. Since we are summing up inflow amounts that conform to known payment patterns, V_1 offers an estimate on each ransomware's revenue from ransom payments. We stress that V_1 is an underestimate; per our discussion in Section IV-E, an unknown number of wallet addresses are likely to be missing from our clusters.

To show how much of the original inflows satisfies Filter 1, we calculate the ratio, $\frac{V_1}{V_0}$. A larger ratio suggests a higher coverage and that we can account for a higher portion of values in the inflows as potential ransom revenue; for instance, 99.7% of Cerber's inflow values potentially consist of victim payments. In contrast, this ratio is low for certain families, likely because we are unable to identify all known payment amounts or Bitcoin graph properties. Spora victims, for example, can choose to pay for one or multiple ransom "packages". The ransom amounts are not fixed for each victim. As expected, V_1 for Spora only includes payments by the real victims we found in public sources, rather than by potential victims that we do not know about.

Filter 2: Another signal for an inflow to be a likely ransom payment is that the inflow sends bitcoins from an exchange's cluster to a ransomware cluster. As described in Section II, a ransom note typically suggests that the victim purchases bitcoins from exchanges, presumably because a random victim is unlikely to already possess bitcoins himself. Thus, we assume that a victim is likely to first acquire bitcoins from an exchange, before sending the bitcoins to his ransom address. To this end, we develop Filter 2, which includes an inflow transaction only if it sends bitcoins from a wallet address(es) from a known exchange's cluster. We check if a wallet address belongs to an exchange, and what exchange, using Chainalysis' API.

Chainalysis is a proprietary online service that links clusters of wallet addresses to the likely real-world identities. It regularly transacts with known Bitcoin-related services, such as exchanges, to discover and cluster wallet addresses used by these services [5], while excluding CoinJoin transactions using a proprietary heuristics-based algorithm.

We show the result as V_2 in the table. The ratio of $\frac{V_2}{V_0}$ suggests how much of a ransomware family's inflows is sent from exchange clusters. For 6 of the 10 families, this ratio is lower than $\frac{V_1}{V_0}$; for the remaining 4 families, $\frac{V_2}{V_0} > \frac{V_1}{V_0}$, but

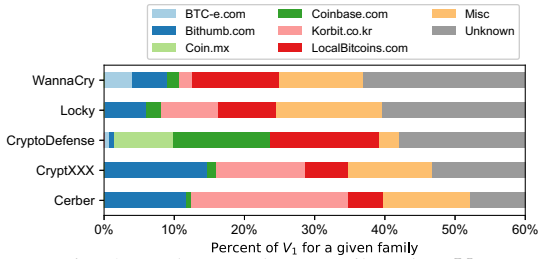


Fig. 4: Exchanges that contributed to V_1.

$\frac{V_2}{V_0} < 50\%$. The relatively low V_2 values are likely because Chainalysis does not have perfect coverage and is unable to identify many of the exchange clusters.

To show the overlap between Filters 1 and 2, we apply both filters to the inflows and compute V_3. However, the recall of V_3 is likely too low to meaningfully estimate ransomware revenue.

For the rest of the paper, we apply only Filter 1 due to the likelihood that Filter 2 has poor recall because of incomplete coverage of exchange addresses by Chainalysis. Moreover, the paper will only consider Cerber, CryptXXX, CryptoDefense, Locky, and WannaCry, as we can account for most of their inflows as potential ransom payment, i.e. $\frac{V_1}{V_0} > 50\%$. We leave as future work creating additional tracing and filtering algorithms that can improve our accounting for the other ransomware families.

V. Payment Analysis

Based on the methods we created in the prior sections, we are able to estimate each ransomware family's revenue in Section V-A and characterize potential victim payments in Sections V-B through V-D. Finally, we measure the potential cash-out behaviors in Section V-E.

A. Estimating Revenue

The V_1 column in **Table IV** shows the total revenue potentially generated from ransom payments. To visualize the likely revenue over time, we plot a stacked bar graph, **Figure 3**. In total, we are able to trace $16,322,006 US Dollars in 19,750 likely victim ransom payments for 5 ransomware families over 22 months. This is probably a conservative estimate of total victim ransom payments due to our incomplete coverage.

B. Payment Mechanisms

Some of the victims likely purchased bitcoins from exchanges before paying the ransom. We would like to determine what these exchanges are and how much of the ransom came from each exchange. For every inflow that satisfies Filter 1, we identify the input wallet addresses and construct a cluster using the co-spending heuristic; we call this cluster the *source cluster*. We use Chainalysis' API to determine the likely real-world identity of the source cluster; it could be an exchange, a non-exchange, or "Unknown," in which case Chainalysis has no information regarding the cluster's identity. For each ransomware family, we identify the top three exchanges that sent the highest amount of inflows in US Dollars. Across

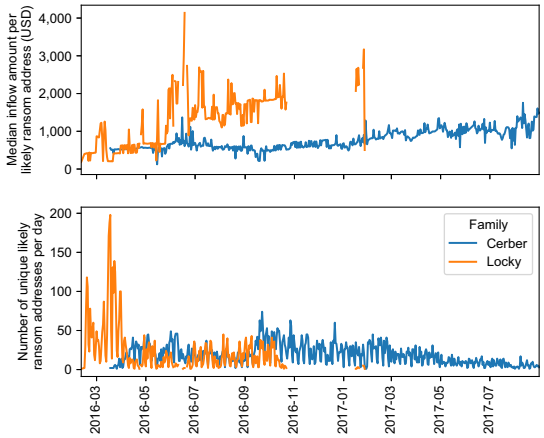


Fig. 5: Inflows to Cerber and Locky that satisfy Filter 1.

the five ransomware families we study, the top exchanges include BTC-e, Bithumb, Coin.mx, Coinbase, Korbit, and LocalBitcoins. We label all other exchanges or real-world entities as "Misc". **Figure 4** presents the breakdown of V_1 values by exchanges, relative to the total V_1 of each family. We truncate the x-axis at 60% to highlight the distribution of known exchanges.

Of the top exchanges, both Bithumb and Korbit require users to purchase bitcoins in Korean Won. Also, both require users to have Korean phone numbers upon account creation. These requirements present hurdles to non-Korean users attempting to use these exchanges. Both of these exchanges account for $2,619,709, or 34.1% of Cerber's ransom payments, likely paid for by victims in South Korea. The remaining four exchanges, in contrast, do not have such geographic restriction; international users can deposit money to these four exchanges and purchase bitcoins.

C. Payment Dynamics

Once a victim acquires bitcoins, she typically sends the bitcoins to a ransom address. For ransomware families that generate a unique ransom address per victim, such as Cerber and Locky, we can use the likely ransom addresses to characterize the individual behaviors of potential victims. In particular, we can estimate the number of paying victims by counting the number of likely ransom addresses, as shown in the bottom chart of **Figure 5**. The Locky cluster sees the highest number of likely ransom addresses on March 18, 2016, when 198 victims are likely to have paid a ransom. In the top chart, we plot the median inflow amount per likely ransom address. Of particular note, Locky received a median inflow amount of $4,137 on June 18, 2016 across two likely ransom addresses: one received 4 bitcoins ($3,008) in a single transaction, and the other received 7 bitcoins also in a single transaction ($5,265). Unfortunately, we do not have an explanation for these larger payments, but they make up a tiny fraction of the payments.

In addition to using the median, we present the distribution of inflow amounts per likely ransom address in **Figure 6**,

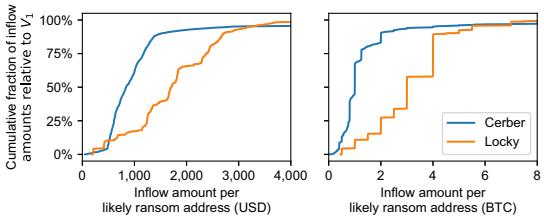


Fig. 6: Distribution of likely ransom payments.

denominated in either US Dollars (left) or Bitcoin (right). In particular, 50% of all Cerber's ransom payments comes from likely ransom addresses that receive at least $838 worth of bitcoins. By contrast, 50% of all Locky's ransom payments are from likely ransom addresses that receive at least $1,715 worth of bitcoins. Overall, both charts suggest that a potential Locky victim probably pays a higher ransom than a potential Cerber victim. Note that the distribution of Locky's bitcoin inflow amounts is step-like, as Locky's ransom amounts are known to be increments of 0.5 bitcoins.

D. Payment Timing

For each likely ransom address, the inflow's timestamp may reveal when victims likely paid. Most of the likely ransom addresses are each associated with only one inflow, which suggests that a potential victim paid the ransom in a single transaction. In fact, 95.4% of Cerber's ransom payments and 98.3% of Locky's ransom payments were paid for in this way. We extract the timestamp from every single inflow transaction. The remaining likely ransom addresses are each associated with two or more inflows, likely because a victim did not fully pay the ransom as he had not accounted for the transaction fees; in this case, we extract the timestamp from the earliest inflow per likely ransom address.

These timestamps show when victims potentially paid the ransom, in terms of days of the week and hours of the day. We show the distribution in **Figure 7**. For instance, 22.2% of Locky's ransome payments comes from inflows on Thursdays. Also, 9.0% of Cerber's ransom payments are made around 08:00 hours UTC. Of particular note is that a single peak hour contributed most to Cerber's ransom payments, while there are two such peak hours for Locky. One possible explanation is that Cerber's paying victims were more concentrated in a certain geographic region (hence the same timezone) than Locky's paying victims, although we cannot validate this absent ground-truth. Furthermore, the least amounts of inflows for both Cerber and Locky are observed around 23:00 hours UTC. It is likely that most of the paying victims were located in Asia based on the diurnal pattern [34].

E. Characterizing potential Cash Out

In addition to inflows, we examine the *outflows* from ransomware clusters. An outflow is a transaction that transfers bitcoins from a wallet address of a ransomware cluster to an address of a non-ransomware cluster. Using **Figure 1** as an illustration, Address 4 sends an outflow transaction to an exchange wallet address, while Address 11 sends an outflow to

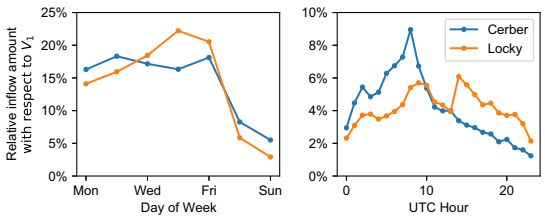


Fig. 7: Distribution of inflow amounts over a week (left) and over a day (right) of ransom payments.

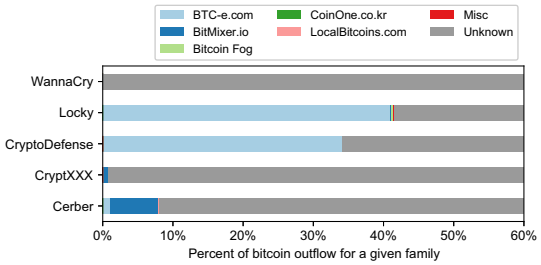


Fig. 8: Real-world entities that received bitcoins from ransomware's outflows.

an unknown cluster. We observe an outflow when, for instance, a ransomware operator moves bitcoins from her own cluster to a Bitcoin exchange, presumably to convert her bitcoins to fiat currency (i.e., US Dollars). Also, an outflow occurs when an operator moves bitcoins into a mixer (recall from Section II). After mixing, the operator will presumably transfer her bitcoins from the mixer to an exchange for conversion to fiat currency, although we are unable to track bitcoins that have entered a mixer's cluster. In general, whether the operator sends the bitcoins from the ransomware's cluster to an exchange, a mixer, or some unknown clusters, we cannot guarantee if and when the operator cashes out. Nonetheless, outflows mark the beginning of a process that could potentially lead to exchange for fiat currency.

To study outflows, we first compare the timings of outflows against inflows. This comparison allows us to estimate the duration in which ransomware operators are holding bitcoins before potentially cashing out. To this end, we trace how bitcoins flow from likely ransom addresses (i.e. inflows that satisfy Filter 1) to outflow transactions. The bitcoins could flow directly from a likely ransom address to an outflow transaction. Alternatively, bitcoins may go through intermediate transactions, flowing from one wallet address of the ransomware's cluster to another wallet address of the same cluster, before the bitcoins reach an outflow transaction. In either case, we extract the timestamp when the likely ransom address first receives bitcoins, and also the earliest timestamp among the outflows (as the bitcoins may be split into multiple outflows). In the median case, the bitcoins remained in WannaCry's cluster for 79.8 days, while for Cerber and Locky, the median holding durations for Cerber and Locky are 5.3 and 1.6 days respectively.

Another insight we can gain from outflows is how ransomware operators potentially cash out the ransom bitcoins. For each outflow transaction of a given ransomware family, we look at the output wallet address(es), which, by definition, should be in non-ransomware clusters. Using Chainalysis' API, we obtain the real-world identities of these clusters, such as exchanges, mixers, or "Unknown." For each ransomware family, we identify top three real-world entities that receive the most US Dollars from the ransomware's cluster. Across the five ransomware families, the top entities overlap and include BTC-e, CoinOne, and LocalBitcoins (all exchanges), along with BitMixer and Bitcoin Fog (both mixers). We show their distribution in **Figure 8**. Real-world entities that are not a part of the top ones are labelled as "Misc."

Compared with **Figure 4**, **Figure 8** shows a different set of known exchanges. In particular, BTC-e (whose operator was arrestes and which is now defunct) is the biggest known exchange responsible for the outflows of Locky and CryptoDefense; $3,223,015 of Locky's outflows entered BTC-e's cluster. If law enforcement agencies were able to obtain BTC-e's internal transaction records (which presumably map Bitcoin wallet addresses to banking information), they could potentially trace 41.0% of Locky's outflow values to real-world entities. A further difference is the use of Bitcoin mixers, which we did not detect in source clusters for the inflows. For example, $541,670 (6.8% of Cerber's total outflows) was sent from Cerber's cluster to BitMixer.

VI. Impact on Infected Victims

In the previous two sections, we use the blockchain to characterize the behaviors of victims who likely paid the ransom. However, it is difficult to infer the behaviors of victims who did not pay, as the blockchain only records activities of payments. We propose and implement a sinkholing-based method to intercept the communication between an infected machine and the ransomware's command-and-control (C2) server. Using Cerber as a case study, this section discusses how we use our method to gather statistics on victims infected with Cerber, along with the insights we draw from this data. We choose to focus on one ransomware family, Cerber, because of the manual effort required to reverse engineer how the malware communicates with the ransomware operators.

We start by reverse engineering Cerber's telemetry protocol. Using this knowledge, we then intercept Cerber victims' telemetry traffic, along with what this traffic reveals about the impact on victims — for instance, how many victims are infected and how long it takes for the encryption process to complete.

A. Reverse Engineering Network Traffic

We start our reverse engineering process by executing two binaries which our algorithm classified as Cerber on a bare metal machine with Windows XP installed. We choose not to use a VM to reduce the likelihood that the ransomware samples might behave differently from those in the wild. The host is not connected to the Internet. We capture all packets that the machine sends out with TCPDump. Inside the host's

file system, we place documents that Cerber is known to encrypt [11]. We also instrument the file system to log when files are changed, so that we can track Cerber's process of encryption.

In each of the three executions of the same binary, it consistently broadcasts a UDP packets to four different /24 subnets at port 6892. For each of the subnets, the binary sends three types of packets. Before the binary encrypts any files, it sends out the first type of UDP packets, which we shall call Packets A. Each of these packets include a five-byte identifier in the payload, which is referred to as the Partner ID by an analysis report [11]. This ID is the constant when re-executing the same binary, but it changes with a different binary. Within five seconds of broadcasting Packets A to four subnets, the binary starts encrypting our files, followed by the broadcast of the second type of packets, which we call Packets B. The payload of these packets includes the Partner ID and a 12-byte identifier that changes across execution of the same binary, which the report refers to as the Machine ID [11]. Within five seconds of the termination of encryption, the binary broadcasts the third type of packets, which we call Packets C, whose payload includes the Machine ID. Afterwards, the ransom note is displayed, from which we extract the ransom wallet address. Finally, we re-image the host's hard disk to prepare for the next execution.

B. Analyzing Cerber's Packets in the Wild

Broadcasting telemetry packets across different subnets effectively hides the IP address(es) of Cerber's infrastructure, but it has also created an opportunity for us to observe these packets. By buying an IP address within the subnets, we can capture the telemetry packets and analyze the behaviors of Cerber binaries in the wild.

First, we need to determine what IP address to purchase. In the last two weeks of January 2017, we executed 1,256 binaries that we classify as Cerber in sandboxes. These binaries contacted 158 /24 subnets in total. We compute the number of packets that each subnet was sent. The top two subnets, based in Norway and Greece, were sent 111,361 and 110,595 packets respectively. Neither responded to our request to purchase an IP address. The third subnet was sent 46,336 packets. It belongs to a Russian hosting provider, from which we purchased a server at an IP address in the subnet and ran TCPDump on the server between February 2 and 20, 2017.

Overall distribution of infection: In total, we received 88,240 UDP packets across 1,512 IP addresses. Of these packets, 92.0% are Packets A, 4.7% are Packets B, and 3.2% are Packets C. The relative under-representation of Packets B and C could be due to the loss of UDP packets, as the prior broadcast of Packets A could have filled up the queues along the route, or it could be due to the duplication of Packets A. Even though the Cerber binary in our own environment sent out the same number of Packets A, B, and C, the exact behavior of the ransomware in the wild could be different — possibly due to a different version.

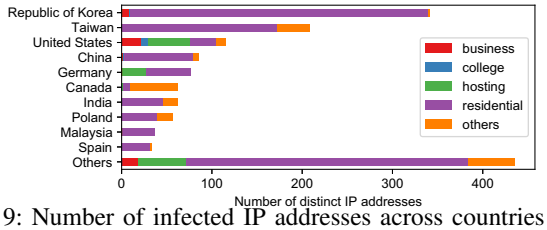


Fig. 9: Number of infected IP addresses across countries and ISP types.

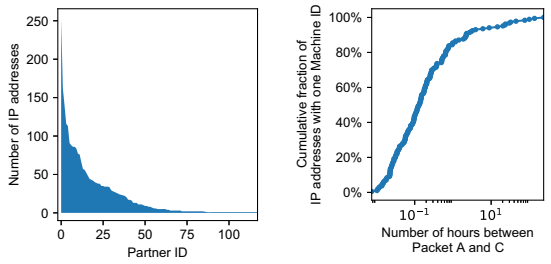
The 1,512 IP addresses offer a lower-bound on the number of infected hosts. There are two reasons why this number is a lower-bound. First, the packets we captured came from Cerber binaries that sent telemetry data to our choice of the subnet. We cannot guarantee that all binaries in the wild would broadcast to the subnet. Thus, an unknown number of infected hosts are likely to be missing from our data. Second, multiple infected hosts could be behind a NAT or may be assigned the same IP address at different times.[7] As such, we could be underestimating the scale of infection.

We show the geographic distribution of infected IP addresses, along with the type of ISP for each IP address, in **Figure 9**. We determine the country and ISP type of each IP address using Maxmind [35]. As shown in the chart, South Korea has the most number of infected IP addresses that sent us the telemetry packets; 22.6% of all infected IP addresses are from the country. IP addresses from residential ISPs contribute to 74.5% of the infected IP addresses. ISPs that are labelled as "hosting," which are hosting service providers, account for 8.5% of the IP addresses.

Distribution of infected hosts: To distinguish between different infected hosts with the same IP addresses, we use the Machine IDs extracted from Packets B and C. Only 583 IP addresses (38.6%) reported Machine IDs.[8] Among these IP addresses, 412 of them (70.7%) were associated with exactly one Machine ID, and 132 of them (22.6%) are each associated with 2 to 10 Machine IDs. For the 412 IP addresses, residential IP addresses account for 80.3% while hosting services account for 9.0%. For the 132 IP addresses, 59.1% are residential and 23.5% are hosting services. At the tail end, one IP address reported 1,162 different Machine IDs. This IP address belongs to a major hosting service in the US (labelled as "hosting" in **Figure 9**). On average, we received Packet B from this hosting IP address with a new Machine ID every 22.6 minutes. We are not sure if telemetry packets from this IP address are from real victims or synthetic ones. In general, we do not remove IP addresses from our analysis even if they appear to be coming from synthetic victims. Absent ground truth, we cannot distinguish IP addresses from sandboxes or from, for instance, a commercial VPN provider that uses hosting

[7]It is unlikely that the same host is infected multiple times; we tried executing the same Cerber binary the second time in our VM (Section VI-A), but the repeated execution did not result in more telemetry packets.

[8]There are 25 Machine IDs, such that each of them was reported from two IP addresses. Together, there are 18 such IP addresses. We suspect the infected hosts were either mobile clients, or they experienced DHCP reassignment.



(a) (b)

Fig. 10: (a) A histogram that shows that number of infected IP addresses per Cerber partners. (b) The time from observing Packet A at a given IP address to observing Packet C at the same IP address.

services and whose subscribers happen to be widely infected with Cerber. This uncertainty highlights some of the challenges of distinguishing real infections from sandbox traffic.

Distribution of partners: Based on our own execution of Cerber's two binaries in Section VI-A, it appears that the Partner ID might be persistent across binaries. As Cerber is known to operate in the affiliate model [11], it is likely that a partner is an affiliate that distributes a particular binary in exchange of a percentage of ransom revenue; the Partner ID thus offers a possible insight on the number of infected IP addresses for each affiliate.

We extract the Partner ID from Packets A or B for each infected IP address and present the number of unique IP addresses for each partner in **Figure 10a**. In particular, Partner 0 infected 253 distinct IP addresses while Partner 1 infected 162. We observe a total of 118 partners, but Partners 0 through 7 account for 51.5% of all infected IP addresses. Each of these 8 partners are associated with between 13 and 33 infected countries across the world. Identifying major affiliates could be useful for prioritizing which affiliates to investigate and target for technical and law enforcement based interventions.

Duration of encryption: Recall, from Section VI-A, that the Cerber binary sends Packets A before encrypting user files, and when the encryption is complete, the binary sends Packets C. The duration between Packets A and C allows us to measure how long it takes for encryption of files to complete; the shorter the duration, the smaller the window is between infection and when the malware needs to be detected before all files are encrypted.

In order to compute this duration, we first identify IP addresses that are associated with exactly one Machine ID. If an IP address is associated with multiple Machine IDs, then there could be multiple Packets A and multiple Packets C; it would be difficult to determine which Packets A-C pair belongs in the same infection. Using IP addresses each with a single Machine ID reduces this error, although what appears to be a pair of Packets A and C could come from different infections due to packet losses. Absent ground-truth, we assume that Packets A and C, in this case, likely mark the beginning and end of encryption for a single infection.

In total, 412 IP addresses are associated with a single Machine ID, but we have received both Packets A and C from only 182 of the IP addresses. For these 182 IP addresses, we plot the distribution of the encryption duration in **Figure 10b**. The median duration is 7.8 minutes. In other words, if a victim finds that one of his file is encrypted, she has less than 7.8 minutes to back up other documents before they are all encrypted. Alternatively, if the victim uses software that automatically detects ransomware encryption — for instance, using techniques proposed by Kharaz *et al* [36] — then the detection algorithm has less than 7.8 minutes to react.

VII. Discussion

It is widely known that ransomware causes harm to many people either through monetary losses or destruction of files. This threat is challenging to measure, but an improved understanding of the ransomware ecosystem is a key first step to identifying new and potentially more effective intervention strategies. Based on our measurements, we propose a multi-pronged strategy to improve our ability to measure and reduce the harm caused by ransomware. In this section, we outline our ideas and discuss looming hurdles, including ethical issues that are unique to ransomware. A full investigation of our suggestions will require significant future work.

Estimating conversion: One open question that remains unanswered in this paper is conversion. Given an infection, what is the probability that a victim might pay the ransom? The telemetry data that we collected in Section VI could have been used to estimate the conversion rate of Cerber, but it involves ethical problems that caused us to decide against performing this analysis.

Specifically, Cerber's telemetry gives us indirect access to individual victims' payment record (or the lack thereof). After the ransomware finishes the encryption, the ransom note automatically appears on the victim's desktop and asks the victim to visit a set of ransom payment websites. The URLs are in the form of `http://`$id1$`.hostname/`$id2$, where $id1$ is the hidden service ID shared across multiple infections (as victims can make payments via Tor at `http://`$id1/id2$ as well), and $id2$ concatenates the Partner ID and Machine ID, along with an MD5-based checksum (which we discovered in our own reverse engineering of the binary). To pay, the victim visits one of the URLs and sees a webpage customized for the victim. The webpage contains $id2$, a Bitcoin ransom address unique to the victim, and the ransom amount. A five day countdown is started when a victim visits the page for the first time; afterwards, the ransom doubles (based on our experience with synthetic victims). Our telemetry data's Packets B contain both the Partner IDs and Machine IDs, enabling us to compute $id2$ and, in theorey, visit the victim's payment URL to check if and when the victim paid.

However, we did not conduct this analysis, since visiting the URL might cause harm to victims. If we visit the URL before the victim visits, the countdown would start immediately,

which might cause the victim to have to pay double the ransom amount. One strategy is to wait for several months after our data collection in February 2017 before we visit the victims' URLs. Regardless of how long we wait, we cannot guarantee that all victims would have either visited the payment URLs or decided to re-install their systems during this period. As such, the risks of the analysis outweigh the benefit of estimating the conversion rate.

Coverage limitations: Our measurement techniques provided improved coverage over prior techniques, but it is still not complete. One of the main limitations is that our transaction filtering methods were not effective for some ransomware families, such as those that had a dynamic pricing structure or for which we did not know the ransom amount (e.g., Spora). Another limitation was that, for ransomware campaigns where we did not have a binary or that were no longer operating, we could not generate synthetic victims and make micropayments. We plan on exploring improved filtering techniques to cover more families. We are also exploring OCR and NLP methods for finding more reported payments from victims to improve coverage of ransomware measurement.

We will also continue to investigate methods of tracing additional cash-outs that traverse mixers or use other methods of obfuscating flows of bitcoins. This is key to increasing the risk of liquidating ransomware profits.

Intervention: Our measurement study indentified potential intervention beyond improved ransomware detection and backing up of files, such as increasing the difficulty and risk when ransomware operators cash out their bitcoins. Our methods of tracing ransomware payment assist in this objective; indeed BTC-e's operator was arrested and the exchange was closed [37].

Another potential intervention point is to disrupt a victim's ability to pay the ransom. During our analysis of Cerber, we found that the `hostname` part of Cerber's payment URL is generated using a domain-generation algorithm (DGA). Based on our analysis, the `hostname` is the prefix of the most recent wallet address that receives and sends bitcoins from and to w, another wallet address likely controlled by Cerber. This DGA creates an opportunity for us to disrupt the payment infrastructure. We could prevent victims from being able to pay the ransom by using our own wallet addresses to send bitcoins to w, which would have created new `hostnames` to divert victim visits from Cerber's payment infrastructure. However, we chose not to conduct this intervention, since it would have also prevented victims from recovering their files.

This introduces a unique ethical issue. We must consider the impact on victims before taking down ransomware infrastructure. Whereas disrupting conventional malware reduces the damage on victims, the effect could be the opposite for ransomware. Any attempts to prevent ransom payment could be risky. Herein lies a common ransomware dilemma: If every victim did not pay or was prevented from paying, the scale of the problem would likely decrease; however, this would mean that some individuals would incur additional harm by not being able to recover their files.

VIII. Related Work

The initial Bitcoin tracing method that links together flows with multiple input transactions was proposed in prior studies [38], [39], [5] and in the original paper proposing Bitcoin [24]. However, this method is now prone to incorrectly linking flows that use anonymization techniques, such as CoinJoin [23], [40] and CoinSwap [41]. Moser and Bohme [42] developed methods of detecting likely anonymized transactions. We use Chainalysis's platform, which uses all these methods and additional proprietary techniques to detect and remove anonymized transactions, to trace flows of bitcoins.

BitIodine is a Bitcoin forensic analysis tool which used reported victim ransom payments to perform a payment analysis for a single ransomware family, CryptoLocker [7]. A followup study by Liao Et. Al. [6] performed an expanded analysis of CryptoLocker by discovering additional reported victim ransom payments and further analyzing whether an inflow is a likely victim payment (similar to our Filter 1). Finally, researchers from FireEye performed an analysis of the actual conversion rate of infections to paying victims for TeslaCrypt [43]. The data source for the FireEye study was not explicitly mentioned, but was likely the backend database for TeslaCrypt that was either leaked or seized.

Intervention strategies include technical solutions, such as detecting changes on the file system as a result of ransomware infection [36] and analyzing network traffic [44]; or economic solutions, such as disrupting the payment processors [45].

We use BinDiff [31] to identify similar malware binaries. Other related methods include SigMal [46].

IX. Conclusion

Our study of the ransomware ecosystem illustrates that the phenomenon of cybercriminals increasingly using Bitcoin for payments produces an opportunity to gain key insights into the financial inner workings of these operations. We have created a set of measurement methodologies and used them to conduct a detailed two year end-to-end examination of the ransomware ecosystem. Our methods allowed us to track ransom payments from the acquisition of bitcoins by victims, to the cash-out of bitcoins by the ransomware operators. We were able to conservatively estimate that the overall ecosystem revenue for the past two years was over 16 million USD extorted from on the order of 20,000 victims. Our ensuing analysis of ransomware operators' cash-out strategies indicated that BTC-e was a key piece of support infrastructure that was used to exchange millions of USD worth of ill-gotten bitcoins into fiat currency. Our study also illuminates many open technical and ethical issues to measuring and intervening on the ransomware ecosystem.

Acknowledgments

This work was funded in part by the National Science Foundation through CNS-1619620, CNS-1717062, and CNS-1629973, and by gifts from Comcast and Google. We thank Melissa McCoy, Kurt Thomas, and Geoffrey M. Voelker for their feedback, and Cindy Moore and Brian Kantor for their technical assistance.

REFERENCES

[1] Wismer, David. Hand-To-Hand Combat With The Insidious "FBI MoneyPak Ransomware Virus". *Forbes, https://www.forbes.com/sites/davidwismer/2013/02/06/hand-to-hand-combat-with-the-insidious-fbi-moneypak-ransomware-virus/#56afdeb1504a*, 2013.

[2] Nicolas Christin. Traveling the Silk Road: A Measurement Analysis of a Large Anonymous Online Marketplace. *CoRR*, abs/1207.7139, 2012.

[3] Rebecca S Portnoff, Danny Yuxing Huang, Periwinkle Doerfler, Sadia Afroz, and Damon McCoy. Backpage and bitcoin: Uncovering human traffickers. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1595–1604. ACM, 2017.

[4] Ryan Brunt, Prakhar Pandey, and Damon McCoy. Booted: An Analysis of a Payment Intervention on a DDoS-for-Hire Service. In *Workshop on the Economics of Information Security*, 2017.

[5] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoff Voelker, and Stefan Savage. A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. In *Proceedings of the ACM Internet Measurement Conference*, 2013.

[6] K. Liao, Z. Zhao, A. Doupe, and G. J. Ahn. Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin. In *2016 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–13, June 2016.

[7] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. BitIodine: Extracting Intelligence from the Bitcoin Network. In *Financial Cryptography and Data Security: 18th International Conference, FC*, pages 457–468, 2014.

[8] Damon McCoy, Andreas Pitsillidis, Grant Jordan, Nicholas Weaver, Christian Kreibich, Brian Krebs, Geoffrey M Voelker, Stefan Savage, and Kirill Levchenko. Pharmaleaks: Understanding the business of online pharmaceutical affiliate programs. In *Proceedings of the 21st USENIX conference on Security symposium*, pages 1–1. USENIX Association, 2012.

[9] Shuang Hao, Kevin Borgolte, Nick Nikiforakis, Gianluca Stringhini, Manuel Egele, Michael Eubanks, Brian Krebs, and Giovanni Vigna. Drops for stuff: An analysis of reshipping mule scams. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1081–1092. ACM, 2015.

[10] Brett Stone-Gross, Ryan Abman, Richard A Kemmerer, Christopher Kruegel, Douglas G Steigerwald, and Giovanni Vigna. The underground economy of fake antivirus software. In *Economics of information security and privacy III*, pages 55–78. Springer, 2013.

[11] CerberRing: An In-Depth Expose on Cerber Ransomware-as-a-Service. *Check Point Blogs, https://blog.checkpoint.com/2016/08/16/cerberring/*, 2016.

[12] Ransomware in Action. *https://www.tmsi.hu/antidotum-prezentaciok/antidotum-2017-r1/Ransomware_presentation_DP.pdf*, 2017.

[13] Danny Palmer. Now Cerber ransomware wants to steal your Bitcoin wallets and passwords too. *http://www.zdnet.com/article/now-cerber-ransomware-wants-to-steal-your-bitcoin-wallets-and-passwords-too/*.

[14] John Kevin Adriano. Fake invoice email with HTML attachment spreads Locky ransomware. *https://www.trendmicro.com/vinfo/us/threat-encyclopedia/spam/3621/fake-invoice-email-with-html-attachment-spreads-locky-ransomware*.

[15] Lawrence Abrams. Researcher finds the Karma Ransomware being distributed via Pay-per-Install Network. *https://www.bleepingcomputer.com/news/security/researcher-finds-the-karma-ransomware-being-distributed-via-pay-per-install-network/*, 2016.

[16] Lily Hay Newman. Petya Ransomware Hackers Didn't Make WannaCry's Mistakes. *https://www.wired.com/story/petya-ransomware-wannacry-mistakes/*.

[17] Kan, Michael. Paying the WannaCry ransom will probably get you nothing. Here's why. *https://www.pcworld.com/article/3196880/security/paying-the-wannacry-ransom-will-probably-get-you-nothing-heres-why.html*.

[18] Ducklin, Paul. "Locky" ransomware what you need to know. *https://nakedsecurity.sophos.com/2016/02/17/locky-ransomware-what-you-need-to-know/*.

[19] Malwarebytes Labs. Explained: Spora ransomware. *https://blog.malwarebytes.com/threat-analysis/2017/03/spora-ransomware/*.

[20] ID Ransomware. *https://id-ransomware.malwarehunterteam.com/*.

[21] VmRay. *http://vmray.com*.

[22] Cloud Vision API. *https://cloud.google.com/vision/*.

[23] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. Coinshuffle: Practical decentralized coin mixing for bitcoin. In *19th European Symposium on Research in Computer Security - Volume 8713*, ESORICS 2014, pages 345–364, New York, NY, USA, 2014. Springer-Verlag New York, Inc.

[24] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.

[25] S. Goldfeder, H. Kalodner, D. Reisman, and A. Narayanan. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies. *ArXiv e-prints*, August 2017.

[26] H. Kalodner, S. Goldfeder, A. Chator, M. Möser, and A. Narayanan. BlockSci: Design and applications of a blockchain analysis platform. *ArXiv e-prints*, September 2017.

[27] Google Trends. *https://trends.google.com/trends/*.

[28] VirusTotal. *http://virustotal.com*.

[29] Yara rule dataset. *https://github.com/Yara-Rules/rules*.

[30] VXClass. *https://www.zynamics.com/vxclass.html*.

[31] BinDiff manual. *https://www.zynamics.com/bindiff/manual/#chapUnderstanding*.

[32] Cerber version 6 shows how far the ransomware has come. *http://blog.trendmicro.com/trendlabs-security-intelligence/cerber-ransomware-evolution/*.

[33] Tomas Meskauskas. CryptXXX Ransomware. *https://www.pcrisk.com/removal-guides/9963-cryptxxx-ransomware*.

[34] David Dagon, Cliff Changchun Zou, and Wenke Lee. Modeling Botnet Propagation Using Time Zones. In *Proceedings of the 13th Annual Symposium on Network and Distributed System Security*, 2006.

[35] MaxMind. GeoIP2 Precision Insights Service. *https://www.maxmind.com/en/geoip2-precision-insights*.

[36] Amin Kharaz, Sajjad Arshad, Collin Mulliner, William Robertson, and Engin Kirda. UNVEIL: A large-scale, automated approach to detecting ransomware. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 757–772, Austin, TX, 2016. USENIX Association.

[37] 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. *https://www.justice.gov/usao-ndca/pr/russian-national-and-bitcoin-exchange-charged-21-count-indictment-operating-alleged*, 2017.

[38] Elli Androulaki, Ghassan O. Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. *Evaluating User Privacy in Bitcoin*, pages 34–51. Springer Berlin Heidelberg, 2013.

[39] Dorit Ron and Adi Shamir. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Proceedings of Financial Cryptography 2013*, 2013.

[40] Gregory Maxwell. CoinJoin: Bitcoin Privacy for the Real World. *https://bitcointalk.org/index.php?topic=279249.0*, 2013.

[41] Gregory Maxwell. CoinSwap: Transaction graph disjoint trustless trading. *https://bitcointalk.org/index.php?topic=321228.0*, 2013.

[42] M. Möser and R. Böhme. Anonymous Alone? Measuring Bitcoin's Second-Generation Anonymization Techniques. In *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, pages 32–41, April 2017.

[43] Nart Villeneuve. TeslaCrypt: Following the Money Trail and Learning the Human Costs of Ransomware. *https://www.fireeye.com/blog/threat-research/2015/05/teslacrypt_followin.html*, 2015.

[44] K. Cabaj and W. Mazurczyk. Using Software-Defined Networking for Ransomware Mitigation: The Case of CryptoWall. *IEEE Network*, 30(6):14–20, November 2016.

[45] Kirill Levchenko, Andreas Pitsillidis, Neha Chachra, Brandon Enright, Márk Félegyházi, Chris Grier, Tristan Halvorson, Chris Kanich, Christian Kreibich, He Liu, et al. Click trajectories: End-to-end analysis of the spam value chain. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 431–446. IEEE, 2011.

[46] Dhilung Kirat, Lakshmanan Nataraj, Giovanni Vigna, and B. S. Manjunath. Sigmal: A static signal processing based malware triage. In *Proceedings of the 29th Annual Computer Security Applications Conference*, ACSAC '13, pages 89–98, New York, NY, USA, 2013. ACM.