Malware Coordination using the Blockchain: An Analysis of the Cerber Ransomware

Stijn Pletinckx*, Cyril Trap* and Christian Doerr
TU Delft
Cyber Security Group
2628CD Delft, The Netherlands
{S.R.G.Pletinckx@student., C.H.Trap@student., c.doerr@}tudelft.nl

***Abstract*— In order for malicious software to receive configuration information or commands, malware needs to be able to locate and connect to its owner. As hard-coded addresses are easy to block and thus render the malware installation inoperable, malware writers have turned to dynamically generated addresses. Domain generation algorithms (DGA) generate a list of candidate domain names, each valid for only a short time, at which the malware installation searches for its command & control (C&C) server. As DGAs generate a large list of potential domains – out of which one or a few is actually in use –, they leave a characteristic trace of many failed DNS lookups (NXDomain) in the network, and in result most DGAs can be efficiently detected.**

In this paper we describe an entirely new principle of domain generation, actively deployed in the Cerber ransomware, which finds and coordinates with its owner based on transaction information in the bitcoin blockchain. This allows the malware author to dynamically update the location of the server in real-time, and as the malware directly goes to the right location no longer generates a sequence of NXDomain responses. We describe the concept of coordination via the blockchain, and report results on a year-long observation of the assets used in the Cerber campaign.

***Index Terms*— threat intelligence, blockchain, ransomware, C&C, domain-generation algorithm, campaign analysis**

*Both contributed equally to this work.

I. Introduction

When malware such as a botnet or ransomware has infected a victim's PC, the malicious software requires a communication channel back to its owner. Over this line, the botnet installation will receive commands which activities to perform next. It will also use this to upload the digital loot back to the cyber criminal, in case of a ransomware malware the encryption keys used to encrypt the victim's hard drive. Although necessary to create a dynamic functioning malware deployment, this command and control (C&C) channel is also one of the most vulnerable parts in the entire deployment. If the C&C traffic stands out statistically from the background traffic it may be filtered out, hence malware authors have for example moved from IRC used in the early versions of their malware to HTTP traffic which seamlessly blends in. Still, the clients need to actually find the C&C control server.

As a hard-coded IP address is trivial to block, malware authors address their C&C servers by domain names which can be updated with little delay to a new location as soon as some part of their C&C infrastructure is taken offline or its network connectivity blocked. Also domain names may be blacklisted or seized by law enforcement, although some more effort is required for this. Malware authors have hence evolved to the dynamic generation of domain names, which are each active for only a short amount of time, a principle referred to as "domain fluxing". For each time interval, a DGA produces a long list of candidate domain names at which the C&C server may be found, of which in practice only one or a few is actually registered. Each client will independently run the DGA and in random order attempt to connect to the candidates until the C&C server has been found. In order to completely break the control channel, the defender would thus have to register all domain names valid for this time interval, which is prohibitively expensive. In order to generate a predictable list, DGAs use the current time as random number seed, however some recent DGAs have evolved to include some public information such as the current trending topics from Twitter [1] to make a prediction impossible.

This approach of coordinating malware however leaves a very characteristic trace in network traffic. As the candidate lists are large and randomly probed, an infected client will generate dozens or hundreds of lookups to non-existing domains, so-called NXDomains until the correct one is found, only to restart the same behavior in the next time interval. NXDomain responses – especially long sequences of NXDomains – are however fairly uncommon among DNS lookups, in the study of a research network backbone in [2] only 0.51% of all responses in normal DNS traffic were such failed lookups. As these lookup patterns of malware distinctively stand out, recent work has been quite successful in detecting infections based on network traffic patterns [3], [4]. If a malware would be able to evolve in a way to immediately know and thus precisely connect to the current location of a dynamic C&C without generating a (large) number of NXDomain responses, this would significantly reduce the success of current countermeasures.

In this paper, we report on an entirely new class of domain-generation algorithms that locates its server based on information stored in the bitcoin blockchain and in consequence no longer creates any NXDomain responses. This new control paradigm is actively used in the wild for Cerber ransomware installations, and we have followed and traced the activities of this malware ecosystem for a period of 15 months to investigate the principles of coordination, resource churn and ultimately ownership of the malware. This paper makes three main contributions:

- It describes an entirely new paradigm of malware coordination via the bitcoin blockchain, which is actively deployed in the Cerber ransomware family.
- It shows the development of the coordination mechanism over time, and analyzes how some 3700 parts used in the infrastructure over time are rotated and replaced in response to detection.
- It visualizes how individual infrastructure components are reused across malware versions and campaigns in the Cerber family, and demonstrates how they can be linked back by resource reuse to the same actors. It is to the best of our knowledge the first campaign analysis performed on a ransomware family to date.

The remainder of this paper is structured as follows: Section II provides an overview of related work in malware coordination principles and the evolution thereof. Section III describes the principle of the blockchain-based coordination mechanism. Section IV presents the results from our longitudinal analysis of the Cerber ecosystem. Section V summarizes our findings.

II. Related Work

The methods used by malware authors have always co-evolved with the state-of-the-art in detection. While the first type of malware such as Agobot or Spybot implemented a control channel to the C&C server based on Internet Relay Chat (IRC), the fact that IRC traffic is virtually absent in most environments and thus readily stands out let malware developers turn to protocols that better blend in with normal traffic, for example HTTP-based control channels.

Also in the way how the C&C server is identified, surges of development have taken place. The first malware to pick up dynamic domain generation was the Sality botnet in 2006, which combined a dynamically generated subdomain with a hard-coded second level domain [5]. The scheme matured in 2007 with the dynamic generation of the entire fully qualified domain name in the Torpiq botnet and the Conficker C worm [6] which generated some 50,000 new domains per day. Stone-Gross et al. [7] were the first to discover the concept of domain fluxing in 2009, since then dynamic domain generation has become standard practice in malware coordination. Only 7 years after, Plohmann et al. [8] already report and compare the characteristics of 43 DGA families.

A second evasion technique that evolved in response to avoid the easy detection and disruption of the channel to the main server is the diversification of control into a peer2peer overlay. In 2007, the SpamThru botnet was discovered by Grizzard et al. [9] to have added a p2p channel as a backup channel. This mechanism however initially poses the same issue as for a centralized C&C, namely how to find the other instances to talk to. While the first malware used fixed addresses in a p2p mechanism, the Nugache botnet [10] innovated to a fixed list which would be dynamically extended. Today, most botnets incorporating p2p technology will do so in a hybrid approach, due to the long latency in relaying commands in a distributed channel, some of also added niche concepts such as the use of Twitter as a C&C channel [11].

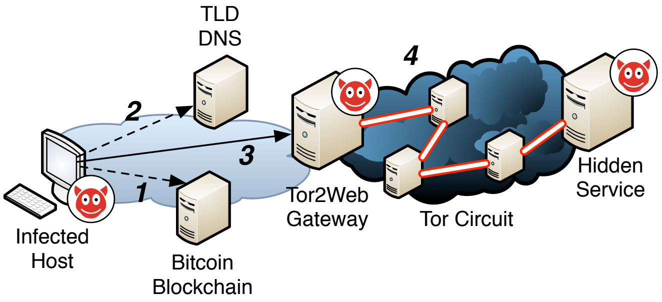


Fig. 1. The Cerber control is hosted as a Tor hidden service and directs bots to the Onion domain via a Tor2Web gateway. The entry point on the open Internet is the weakest link, and the location is dynamically updated through transactions on the bitcoin blockchain.

While we saw a rapid pace of innovation in the way malware is controlled, over the past 15 years only these two paradigms have been observed in practice. In the following, we will describe the emergence of a third paradigm in the wild, which relays location information using the bitcoin blockchain. As we have found this mechanism to be highly robust, we expect this new strategy to experience fast innovation and soon be widely adopted by malware authors.

III. Malware Coordination via the Bitcoin Blockchain

This section will describe the working principle of the blockchain-based malware coordination we have identified within the Cerber ransomware.

When we analyze the existing malware installations, we can identify two Achilles' heels that may be easily targeted by a defender: First, if the connection attempts from the client to the C&C server stand out enough, infected clients can be identified and cleaned up, but more fundamentally the logical or virtual address – i.e., IP addresses or domain names – of the C&C server can be blacklisted to break the entire network. Second, as the C&C infrastructure needs to be physically located somewhere, law enforcement and abuse reports can trigger the network provider to take down the machine.

The Cerber malware coordination introduces mechanisms to address both potential weaknesses. Figure 1 shows the whole setup schematically, which we will explain in the following:

Step 1: A host which executes the malware for the first time will connect to a website listing transactions on the bitcoin blockchain, requesting the activities of a specific wallet. This particular wallet is hardcoded in each malware binary and is static across campaigns, as we will elaborate on in section IV. The malware adds some level of fault tolerance by trying up to 4 bitcoin monitoring websites until the wallet transactions could be retrieved.

Figure 2 displays all the transactions sent and received from one of those bitcoin addresses hardcoded in the Cerber

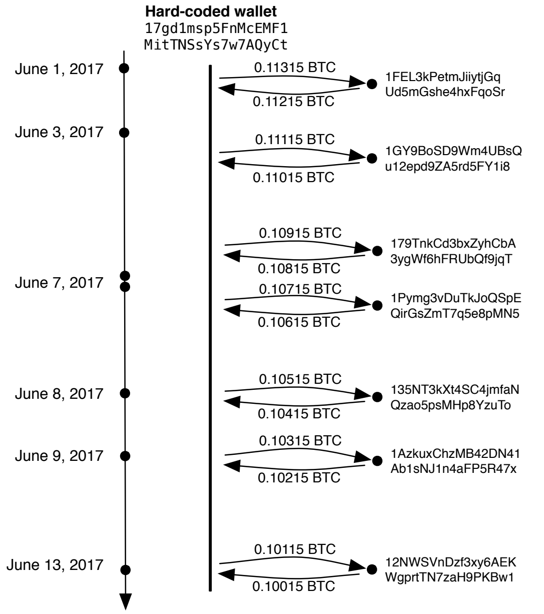
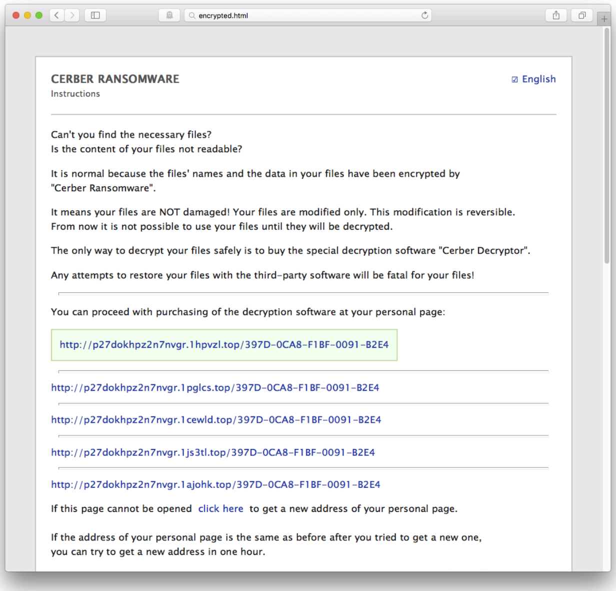


Fig. 2. Transactions in the hard-coded bitcoin wallet within the first two weeks of June 2017.



CERBER RANSOMWARE

Instructions

Can't you find the necessary files?
Is the content of your files not readable?

It is normal because the files' names and the data in your files have been encrypted by "Cerber Ransomware".

It means your files are NOT damaged! Your files are modified only. This modification is reversible. From now it is not possible to use your files until they will be decrypted.

The only way to decrypt your files safely is to buy the special decryption software "Cerber Decryptor".

Any attempts to restore your files with the third-party software will be fatal for your files!

You can proceed with purchasing of the decryption software at your personal page:

http://p27dokhpz2n7nvgr.1hpvzl.top/397D-0CA8-F1BF-0091-B2E4

http://p27dokhpz2n7nvgr.1pglcs.top/397D-0CA8-F1BF-0091-B2E4

http://p27dokhpz2n7nvgr.1cewld.top/397D-0CA8-F1BF-0091-B2E4

http://p27dokhpz2n7nvgr.1js3tl.top/397D-0CA8-F1BF-0091-B2E4

http://p27dokhpz2n7nvgr.1ajohk.top/397D-0CA8-F1BF-0091-B2E4

If this page cannot be opened click here to get a new address of your personal page.

If the address of your personal page is the same as before after you tried to get a new one, you can try to get a new address in one hour.

Fig. 3. The Cerber malware displays a ransom notice after encryption, listing five Tor2Web gateways leading to the same hidden service payment site.

malware for the first 14 days in June 2017. As we see, the wallet 17gd1msp5FnMcEMF1MitTNSsYs7w7AQyCt has exclusively participated in 7 exchanges with other bitcoin addresses, transferring amounts in the order of a tenth of a bitcoin out, and within minutes receiving the same amount minus the transaction fee back from this temporary wallet. All of the temporary wallets are only used for this single exchange, and are never engaged otherwise in any transaction nor keep a balance after the retour.

Step 2: While the temporary wallet is a "throw-away" account of no further significance, its name, the public key, actually is of significance. The first 6 characters of the wallet identifier indicate the domain name the client should connect to as the next element on the path to the control server. While the ransomware authors have first used a mixture of top-level domains such as .win, .com, .pw or .top, all ransomware strains since December 2016 have focused exclusively on the .top TLD. This top-level domain name server is then asked for the server handling the domain name obtained from the blockchain.

As we can see in figure 2, the domain through which the server can be reached is rotated in irregular time intervals, which coincides in some cases with the destination and IP addresses being blocked. Not all of these rotations seem ad-hoc, as we find that new domain names are sometimes registered days in advance before the corresponding transaction is made in the blockchain that sends client traffic to them.

It is important to note that the domain lookup process in step 1 and 2 basically derails much of the fundamental principles of the existing DGA detection approaches. Solutions testing for an abnormal number of NXDomains will find no flood of non-existent domains being queried, as both the domain name listed in the blockchain always exists and the single request will thus always hit.

Step 3: The IP address referred to by the .top domain name however does not host the actual Cerber server, but a gateway which forwards incoming connections to a hidden service located within the onion router ecosystem. The gateway maintains a connection to the Tor network, and acts as a proxy forwarding to the address of the hidden service, the ".onion domain", which is encoded as a subdomain of the 6-character .top domain name. We will refer to the combination of the onion domain and the .top domain name as the "full gateway domain". Tor2Web gateways are not limited to forwarding to only a specific hidden service, but act as a proxy to access any Tor .onion domain.

As the actual location of a hidden service is unknown, a take-down of the server instance and the database is difficult to accomplish, thereby increasing the overall lifespan of a single control instance. During the installation and encryption of the victim's hard drives, the infected PC will contact the control server to dump information about the compromise, which also hosts a website where the victim can pay in order to recover the data. The victim is directed to this payment site through an HTML document created on the infected host, figure 3 shows an example of such ransom notice which lists links using 5 Tor2Web gateways all pointing to the same Tor hidden service. In this type of setup, gateways contain no sensitive information or state, and are thus highly expendable. As we will further explore in the next section, this part is also the component with the highest amount of turnover or churn in the ecosystem.

Step 4: The location information of the hidden service is also hard-coded in the malware, and occasionally changes between versions of the Cerber ransomware binary. This relatively slow churn of the most sensitive part is obviously possible because of the low exposure and attack surface the system has. Even with the address being public knowledge, it is not possible to stop the Tor network from serving requests to this URL or redirect connection attempts to a sinkhole as would be normal practice with seized domain names used in cyber criminal activities.

From the above description, we see that four components are used in the service provisioning of the control server. First, the hard-coded wallet that signals clients which domain name is currently in use, second, a hidden service with a hard-coded domain name at which the perpetrator provides information about the payment, third, the gateway domain which acts as a bridge to the Tor network, and fourth, the IP address from which the gateway is served. In the following section, we will look at the relationship and evolution of these indicators.

From a security perspective, this alternate mechanism is somewhat difficult to detect as it breaks with previous practices of domain generation. The dynamic lookup mechanism implemented by Cerber requires only a single call to a website to retrieve a transaction snapshot of a wallet, and afterwards connects directly to the control instance. This procedure hence does not create any NXDomain responses and burst query patterns, which in traditional DGAs are a clear giveaway of malicious activity and an infected host. At the same time, this mechanism also has the capability to be extremely dynamic, as a single transaction in the bitcoin network immediately steers all clients to a different gateway.

IV. The Coordination of the Cerber Ransomware

The case of the Cerber ransomware is particularly interesting as it is marketed through a "ransomware-as-a-service" (RaaS) model, which means that individual cyber criminals may participate in the distribution of the malware, the infection and extortion of victims, without the need to actually run the Cerber infrastructure themselves. This affiliate concept has been observed in other criminal activities before, such as the selling of counterfeit goods [12], and affiliates receive a cut of the profits from the extortion.

In order to understand the evolution of the Cerber ransomware coordination, we analyzed malware samples and monitored the activity of assets from a period of approximately 15 months, spanning from July 2016 until October 2017. For this time frame, we observed some 3701 indicators, wallet addresses, .onion domains, gateway domains and IP addresses, that were involved in various Cerber campaigns. We have obtained these indicators by analyzing malware samples, installing the ransomware and capturing network traffic, following bitcoin transactions, and linking network resources back to domains that connect to them. This section will present various aspects from this analysis in detail.

The big picture of these relationships is shown in figure 4. The three-dimensional plot visualizes the interconnection and dependency between resources along the x-axis, starting with the hard-coded wallet addresses on the far left to the individual IP addresses used by a gateway on the far right. The vertical bar shows the activity period of a particular resource, for example we see that the orange hard-coded wallet has seized to be active in March 2017, where the red wallet began operation in late April 2017 and has remained active until the end of our study. To simplify reading of the figure and visualize some high-level relationships, the color of the bars and lines throughout the figure is matched with the color of the hard-coded wallet. In other words, the orange hardcoded wallet had connections to the all the orange Tor hidden services (in this case one), and these orange hidden service domains were reached by the orange full-gateway domains and so on. If some gateway servers were reused across campaigns, their corresponding IP addresses are not color-coded but displayed in black. The y-axis contains an index for easier readability of the data in the chart, and has no further meaning.

Already from a high-level inspection of the resource interactions and relationship of infrastructure, we see the extremely complex turnover rate of "open" assets such as domain names and IP addresses which can be easily traced back and blocked, in comparison to the long-lived, almost invariant wallet and hidden service addresses which cannot be interfered with. At a closer look we can also spot times of transition in the setup and operation of the infrastructure, most easily visible in middle column of "full gateway domain". While before January 2017 full gateway domains were only in use for extremely short periods of time, from January 2017 their service life stabilizes. On March 2017, we observe almost an entire replacement of involved domain gateways, linked as we can infer through the colors with the advent and proliferation of two new hidden service control instances. While these behaviors propagate also to the rightmost part of the figure, the plethora of assets make a visual analysis difficult. In the following subsections, we will thus look at various aspects of the Cerber ecosystem separately in detail.

A. Cerber Evolution

To put our study into context, we will briefly summarize the main evolution steps of the Cerber malware. Antivirus companies usually distinguish 6 versions of the Cerber malware, which successively introduced new features:

v1 In Feb 2016, the initial version of the ransomware surfaced, which was distributed via 2 exploit kits and encrypted files based on AES [13].

v2 An update in August 2016, mainly distinguished by the use of a new suffix for the encrypted files. We find that since this version the victim is referred to a hidden service via a Tor2Web gateway.

v3 Minor adjustments around September 2016.

v4 The new version from Oct 2016 obfuscates file names and displays the ransom note in HTML. It targets new file types.

v5 The release from November is equipped with a new delivery mechanism.

v6 In June 2017, Cerber receives a major overhaul, anti-sandboxing and anti-VM methods are added [14].

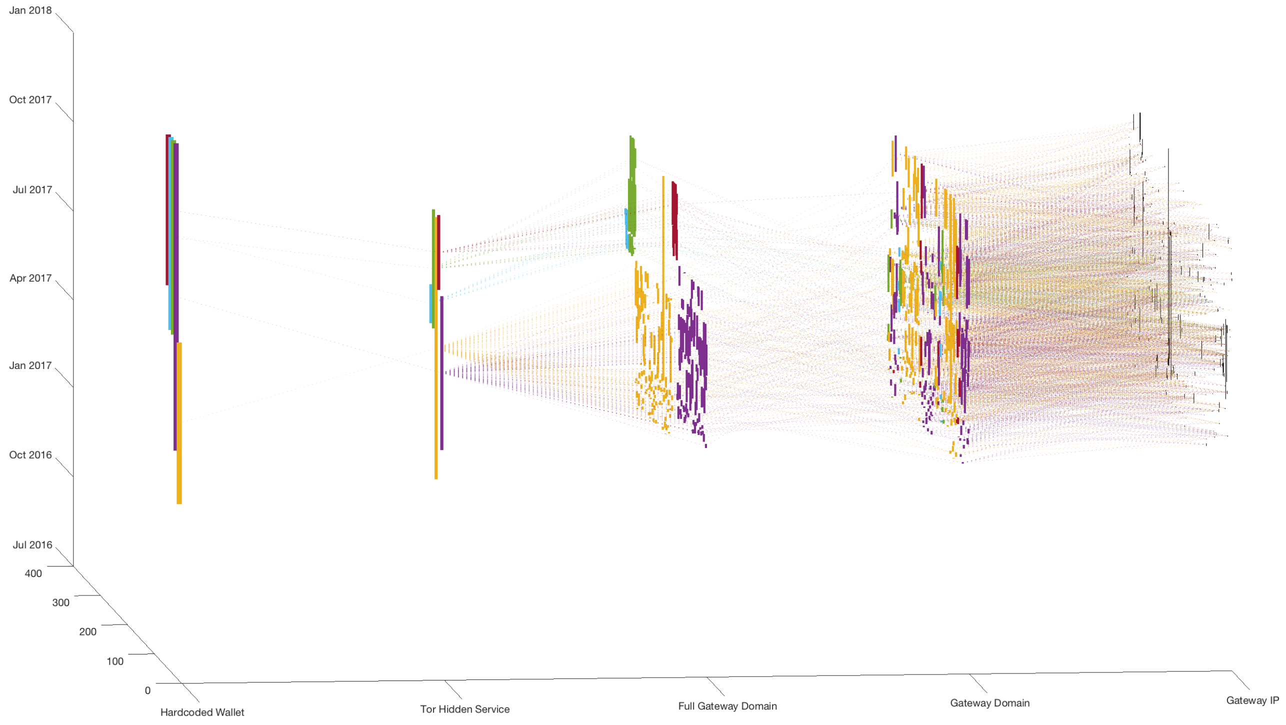


Fig. 4. Interactions and relationship of infrastructure used in Cerber ransomware campaigns.

Some more changes however occurred "under the hood". In mid December, we found the addressing scheme introduced in v2 to change. From this time frame on, we find the Tor2Web gateway domains to be now exclusively derived from bitcoin wallet identifiers, all other addresses are phased out. The bitcoin-based addressing scheme was used already prior to December in combination with other non-bitcoin-based domain names.

Since summer 2016, we have discovered a total of 16 hidden service URLs hosting payment sites. 9 of these were only used until the introduction of the new addressing scheme, 2 during 2016 and the first month of the introductory phase of the bitcoin-based coordination, and 5 exclusively with the new mechanism. While from the perspective of externally-visible software "features", no major functional changes of the malware's behavior between v5 and v6 were reported, we observe that in the backend a complete restructuring of the infrastructure took place.

B. Hosting Networks

Over the observation time, we found a total of 3670 combinations of full gateway domains and gateway hosts. The number of distinct IP addresses was significantly smaller, with only 440 unique IP addresses located in 77 autonomous systems. This was due to the fact that the placement of Tor2Web hosts did not occur randomly across networks and hosting services, but exhibited a strong preference for specific networks, in many cases even returning by accident or choice to the same IP address used in a campaign before.

As autonomous systems have a different size, a normalization is required to be able to compare the prevalence of gateways to be located within a specific network. In the following we will use the relative occurrence of gateways in comparison of the number of IP addresses owned by AS,

$$r = \frac{\text{number of gw in network}}{\text{number of IPs in network}} : \frac{\text{sum of all gw}}{\text{number of all IPs}},$$

which is a natural and easy to understand normalization, where r = 1 would indicate that a network hosts exactly as many gateways as one would suspect from a purely random distribution. Table I lists the 10 most and 10 least affected networks, in terms of their relative occurrence.

All of the networks in the list are chosen more often for gateway hosting than is statistically likely if following a random assignment[1]. Furthermore, we see that also within this list there is a drastic spread between the top and the bottom, with networks at the top being tens of thousands of times more affected by the gateways than those at the bottom. The average relative occurrence is 14108 with a standard deviation of 46230, indicating a very wide variety even within the special group of outliers.

There are two reasons why a large number of full gateway domain to gateway host combinations could be concentrated in just a few hundred IP addresses. First, some of the campaigns exhibit a high resource churn, in this case domain names, which we will discuss in the following section. Second, a

[1] Although it has to be noted that for networks with only a tiny number of gateways, the assessment is less robust.

TABLE I
TOP 10 AND BOTTOM 10 AUTONOMOUS SYSTEMS HOSTING CERBER'S TOR2WEB GATEWAYS BASED ON RELATIVE OCCURRENCE.

AS	#Gateways	Origin	Relative Occurrence r
AS36352	376	US	319499
AS48693	99	Russia	225917
AS3267	114	Russia	87398
AS9009	140	UK	73582
AS58329	30	Netherlands	45639
AS56611	34	US	38793
AS64094	8	US	36511
AS3281	12	Latvia	27383
AS197569	6	Ukraine	27383
AS135112	6	New Zealand	27383
...	...	...	...
AS24961	6	Germany	31
AS50673	6	Netherlands	29
AS20278	2	Netherlands	29
AS53755	4	US	28
AS11878	4	US	23
AS54540	1	US	18
AS54290	4	US	14
AS18978	5	US	8
AS30693	3	US	6
AS8342	1	Russia	2

significant number of IP addresses is reused across campaigns.

Out of the 440 IPs, 169 appeared in different campaigns, sometimes being active in multiple. Figure 5 shows the bipartite graph of reused IP addresses and the hidden service names that used this gateway at some point. The figure reveals a dense mesh of interconnections between individual hidden services, but at the same time different subgraphs can be discovered that share no link between them. One of these are the four top most addresses, which do reuse IP addresses among them but do not share any with for example the wallet at the bottom. The IP addresses at the top right are only linked to addresses in the middle, but not the top. These two subgraphs correspond to the different phases in the evolution of Cerber, the names at the top are exclusively services used in the control during 2016, while the services linked to the addresses on the top right are emerging after the transition period of late 2016 and active in 2017. After this transition period, the amount of IP reuse considerably drops: three quarters of reuse stems from 2016, but also in 2017 common IP addresses in use can be detected across all campaigns.

C. Detection and Resource Churn

As we have concluded in our discussion of the blockchain-based coordination mechanism in section III, there are basically two components that are traceable and thus subject to mitigation, the .top domain pointing to the gateway and the gateway itself. Indeed, as we turn back to the overview figure 4 we can identify basically no regular turnover in the hardcoded wallet and hidden service identifiers, all the churn occurs at the level of the full gateway domain (thus, hidden service + .top domain) and below. In the following, we will investigate how and how often these resources are being replaced during the 15 month observation of the Cerber ecosystem.

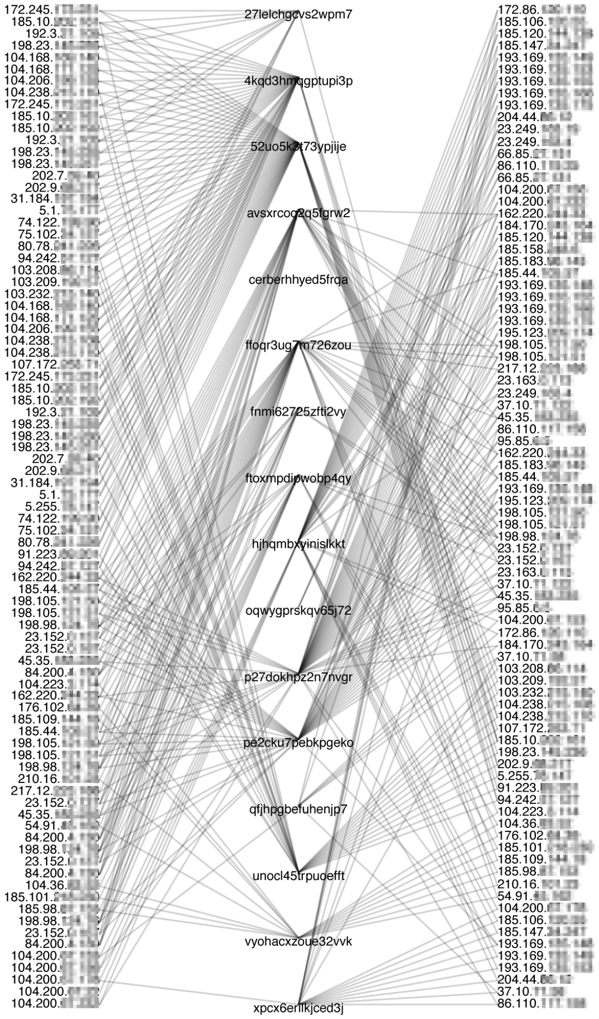
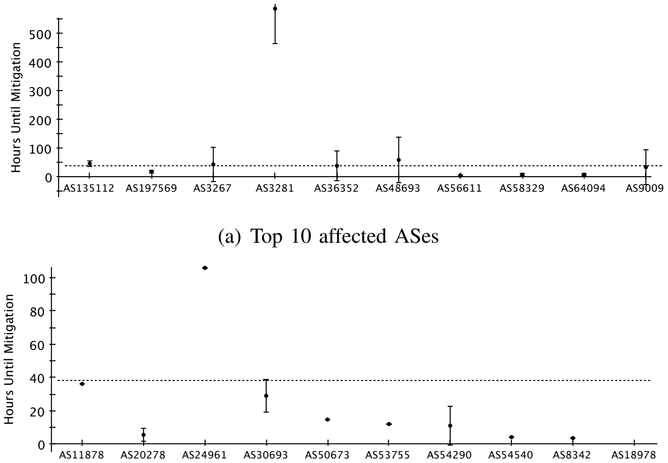


Fig. 5. Associations between reused gateways (anonymized to /16) and connected hidden services. 38% of IPs are used for more than one campaign.

From the list of affected autonomous systems in table I, a small but untested bias to Eastern European countries became visible, and from past analysis we know that malware originating from countries such as Russia does not target Russian hosts to not trigger any reaction from law enforcement [15]. Indeed, also the Cerber malware contains a block list, and does not infect the host if a geo-IP lookup asserts the computer to be located in .am, .az, .by, .ge, .kg, .kz, .md, .ru, .tm, .tj, .ua, or .uz [16], basically all countries belonging to the territory of the former Soviet Union.

This prompts the obvious hypothesis whether the malware authors specifically locate their infrastructure where they assume it to not be interfered with and thereby remain active for an extended period of time, as any forced removal will incur time and costs and potentially even loose a "sale". Such interference may be investigation and seizure by law enforcement, or simply the efforts undertaken by the autonomous system containing the gateway to locate or remove it. Over the 15 month period, we observed that gateways remained



(a) Top 10 affected ASes

(b) Bottom 10 affected ASes

Fig. 6. Mean and standard deviation of response time in removing a gateway from their network.

TABLE II

CORRELATION STRENGTH AND SIGNIFICANCE OF NUMBER OF GATEWAYS IN AN AS TO THE RESPONSE TIME TO A GATEWAY

Correlation Target	ρ	p-value
Gateway count in AS ⟷ avg. time to mitigation	0.36	0.001533
Gateway count in AS ⟷ mitigation standard dev.	0.68	4.72e-11

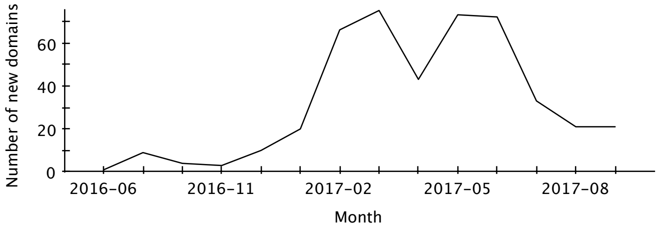


Fig. 7. Number of newly circulated domain names per month.

alive on average only for a duration of 38 hours. The large standard deviation of 112 hours however indicates a fairly different response profile across organizations. Figure 6 plots the arithmetic mean and standard deviation of the gateway mitigation time in hours, for the 10 most and 10 least affected autonomous systems based on relative occurrence as shown earlier in table I. For the examples in these two categories, we see an indicative picture emerging. The overall response time and consistency of mitigation is general much larger for the top 10 networks, compared to the 10 least affected ones.

It is clear that from the perspective of the malware owner hosting in a network with a slow response is preferable, however locating a gateway in a network with a high standard deviation in response time is also useful. A large variance in take down times will likely indicate a network with a less structured and automated approach to detecting and responding to malicious activity. As such, there might not be a formal process or policy in place processing an abuse message in a structured fashion to a service disconnection, the network might not be systematically monitored for anomalies, and no automated vulnerability scanning or a profiling of system usage in place. While a small standard deviation will mean that the gateway is consistently found within a certain reaction time, a large variation will mean some gateways can slip by for extended periods and that detection and mitigation is not guaranteed within a specific timeframe. When testing this relationship for the entire set of gateways and autonomous systems, we find a very strong correlation between the standard deviation and the average time to respond (Spearman's correlation coefficient $\rho = 0.69$, p-value < 0.0001). This would make an inconsistent network a preferred choice and indeed, this picture also emerges from figure 6. While the response time is not directly related to the country of hosting, the top four affected networks from table I are also the ones with the highest amount of variation.

From the analysis of resource reuse and figure 5 we had seen that about 38% of gateways appear across hidden services, and already a visual inspection revealed select prefixes to reappear frequently. This poses the question whether networks are chosen randomly or build up some reputation about their responsiveness, in other words if a network responds slow or inconsistent, is an adversary more likely to return to it? When we correlate the number of gateways that appeared in a particular autonomous system with the average time to mitigation over the entire time, we do find a statistically strong evidence for this relationship. Based on Spearman's correlation, 36% of the variation of gateway placement can be explained through the average time to mitigation alone at a p-value of 0.0015. Even stronger is the correlation between the number of gateways in an AS and the standard deviation of the response time at a $\rho = 0.68$ and a p-value $< 10^{-10}$ as shown in table II. The way a network responds to malicious activity does seem to directly resonate with the amount of criminal activity it will receive in the future.

A similar but less pronounced result can also be found on the churning of domain names. Figure 7 displays the number of newly registered .top domains used across all Cerber campaigns as a function of time. As can be seen in the graph, usage of new domain names is generally comparatively small, typically only a total of less than new 25 .top domains are introduced per month across all Cerber campaigns combined. This situation significantly changes however from March to June 2017 when domain churn temporarily almost quadruples. Figure 8 presents the total number of domains in use during the dynamic behavior in 2017, split up by the hidden service the domains are linked to. Here we clearly see two regimes of temporal activity and that the high number of domain introductions is entirely driven by the campaigns linked to two hidden services.

In comparison to gateways, the lifetime of domains is comparatively long, with an average life time of more than 2 weeks. Surprisingly, we note that the expiration date of domains until they are rotated drastically differs between hard-

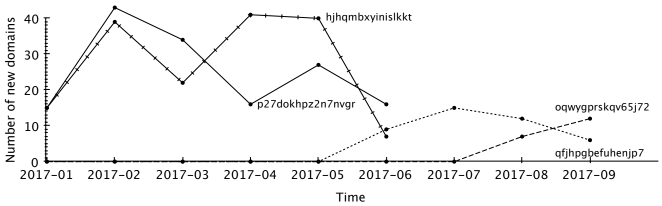


Fig. 8. Total domains in use by the 5 hidden services active in 2017.

coded wallet and hidden services, some hidden services such as the two on the right which are depicted in green and red in figure 4, accomplish a service life more than three times as long as their peers. This low turnover and high amount of continuity can also be seen in figure 4 as entries in the full gateway domain category form a continuous string, while orange and purple are repeatedly interrupted and jump across domains, and ASes. A statistical test (Anderson-Darling for difference in distribution, p-value < 0.01, t-test for different of means, p-value <0.05) between the low turnover group (green, red) to the high turnover group (orange, purple, cyan) confirms that these two groups host their gateways in structurally different AS groups, both given the total number of malicious nodes in a network as well as the relative occurrence. The dashed and solid lines in figure 8 correspond to the low and high turnover services, and give rise to the hypothesis that both sets seem to be independently managed or with different objectives. As a forward identification of malicious .top domains is equally simple for both groups, the only logical explanation for this difference is again the difference in response of the hosting networks. As an AS identifies and removes a gateway, it would also learn about the domain name pointing there and report it to domain blacklists and antivirus vendors. A faster detection of Tor2Web gateways could thus hypothetically also lead to faster churn of domain names.

D. Campaign Relationships

Throughout our analysis, we have repeatedly pointed out commonalities and differences between the various Cerber campaigns. There appeared tight relationships between some hidden services that engaged in heavy resource reuse, while other parts seemed completely detached. Interestingly, the operational behavior across some of the components seemed to differ, as we have shown above they exhibited different turnover rates and were hosted on a different portfolio of hosts. This naturally raises the question whether these commonalities imply similar or identical ownership, and whether these wallets, services and hidden services can be attributed to some abstract entity.

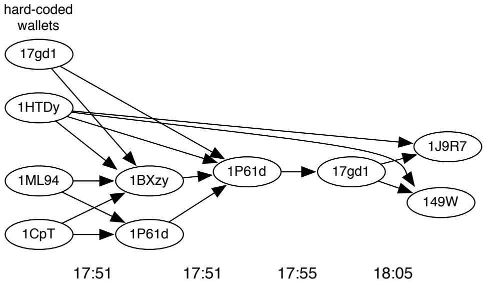


Fig. 9. Final payout scheme and close down of the network on Sep 29, 2017. The bitcoin addresses are abbreviated for better readability.

While we in general found a stringent separation of bitcoin wallets for different purposes – the hard-coded wallets are used for only coordinating the network, newly generated wallets received the ransom, and no clear transactions or other accounts existed that linked these wallets –, this tight separation smeared during the decommissioning of the system. Over the month of July through September, we observed a major decline in activity, and the domain distribution mechanism based on the hard-coded wallets successively stopped. On Sep 29, 2017, the remaining balances in the control wallets were cleared out in 3 separate rounds of transactions at 17:51h, 17:55h and 18:55h that day. Figure 9 shows the relationship of these transactions, which ultimately deposited the entire remaining balance in two new accounts. This final shutdown is astonishing as so far during the campaigns, the strict operational independence of the hard-coded wallets had been maintained. Even stronger, given the different operational characteristics of the hidden services and coordination principles associated with the first two addresses – low domain life time, large domain turnover, different hoster profile – in comparison to the hidden services attached to the bottom wallets in the figure, the hypothesis of two or more administrators would not have been unreasonable, instead it appears that the Cerber ecosystem was controlled by one entity which has conducted different operations in parallel.

It is also important to note that the operator responsible for the ecosystem in 2017 can be linked to the activities earlier in 2016, in other words there was no disruption where a new entity has taken over control of the ecosystem with the new control mechanism, again possible to establish due to common resource reuse. Recall from our discussion around figure 5 the two distinct subgraphs emerging, hidden service addresses from the beginning phases that were using an entirely different set of gateways IPs than those introduced later. Locators from the brief transitionary period however used both, thereby allowing to draw a link from the old to the new scheme.

As stated above, there exists a strict one-to-one relationship between the Tor hidden services and hardware wallets, the hidden service domains were never observed to follow the gateway sequence other than that of a single hard-coded wallet. As the hard-coded wallet is used to direct clients to gateways, the authority over the gateway can also be associated with the hard-coded wallet owner, separating these two (for example in an outsourced gateway operation) would make network operation more complex. While the second point is fairly obvious, the first one is actually astonishing. It implies that the hidden services are also operated by the same entity running the gateways, as there exists no sample that uses the Cerber gateway infrastructure but points to a different payment site,

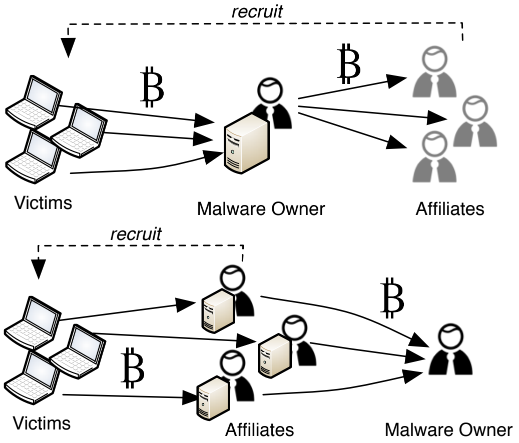


Fig. 10. The Cerber ecosystem could run the RaaS ecosystem based on commission (top) or the use of licensing fees (bottom), various indicators suggest the former.

even though the Tor2Web gateways could be used to connect to any Tor hidden service. As any beneficiary uses a distinct gateway at a time, this rules out the hypothesis that the gateway infrastructure is maintained by a third party.

A ransomware-as-a-service model as the Cerber malware could be operated in two fashions as depicted in figure 10. The ecosystem could operate based on commission, where affiliates spread the malware to potential victims who are then directed to the payment site owned and operated by the malware writer. Upon payment, the malware owner forwards a share of the profits to the affiliates, possibly through a bitcoin anonymizing network for extra protection. Alternatively, the affiliates could spread Cerber and direct victims to their own payment site. They then either pay a monthly license fee for usage or forward a percentage of every ransom. In the latter case of a license-based model, we would expect a collection of independently operated payment sites together with their own coordinating infrastructure, which further share no relationship. This is however clearly not the case: (a) the payment sites/hidden services are bundled with the coordinating infrastructure thus turning the affiliates into full service operators rather than service customers, (b) the amount of hidden services remains constant during the observation time frame which would not be indicative of an expanding and shrinking business and affiliate customer base, and (c) during the decommissioning we have seen the hard-coded infrastructure wallets to be under the control of the same entity. In a commission-based system, the malware owners could identify the affiliate to pay based on a tag or identifier embedded in the malware that is sent by the victim to the control server together with the key. This would not require any change to domain names or wallets, be easier to manage for a large customer base and not imply any major per-affiliate changes in the system. This operation model matches our findings on topology and behavior of the Cerber ecosystem, and is thus the expected modus operandi of Cerber's ransomware-as-a-service model.

V. Conclusion

Domain-generation algorithms are an established building block of malware, which helps malware owners to dynamically direct clients to the control instances while providing a solid defense against external interference. Current DGAs however leave a trace of suspicious NXDomain sequences, which allow an efficient detection of such threats. In this paper, we have reported on the discovery of a new type of malware coordination based on the bitcoin blockchain. Used in the wild within the Cerber ransomware, this mechanism allows the malware owner to instantaneously update the location of the control system, and no longer result in a single NXDomain packet, thereby complicating the detection of infections through a network-based anomaly detection.

We have followed the evolution of the Cerber ecosystem for a period of 15 months, and over the course of this discovered some 3700 indicators of compromise belonging to different types of campaigns. We were able to show how the malware owners rotate resources in reaction to detection and blockage, and could also demonstrate that autonomous systems which are slow to respond to a malicious gateway in their networks, for example because of an undeveloped or manual process, attract more malicious activity into their networks.

References

[1] S. Yadav, A. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in *IMC*, 2010.

[2] J. Abbink and C. Doerr, "Popularity-based detection of domain generation algorithms," in *International Conference on Availability, Reliability and Security*, 2017.

[3] M. Antonakakis and R. Perdisci, "From throw-away traffic to bots: detecting the rise of dga-based malware," in *Usenix Security*, 2012.

[4] M. Thomas and A. Mohaisen, "Kindred domains," in *WWW Conference*, 2014.

[5] McAfee, "W32/sality.m," 2006.

[6] G. Lawton, "On the trail of the conficker worm," *Computer*, vol. 42, 2009.

[7] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: Analysis of a botnet takeover," in *CCS*, 2009.

[8] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A comprehensive measurement study of domain generating malware," in *Usenix Security*, 2016.

[9] J. B. Grizzard, V. Sharma, C. Nunnery, B. Kang, and D. Dagon, "Peer-to-peer botnets: Overview and case study.," *Workshop on Hot Topics in Understanding Botnets*, 2007.

[10] S. Stover, D. Dittrich, J. Hernandez, and S. Dietrich, "Analysis of the storm and nugache trojans: P2p is here," in *Usenix ;login*, 2007.

[11] B. Prince, "Flashback botnet updated to include twitter as c&c," *SecurityWeek*, 2012.

[12] D. McCoy, A. Pitsillidis, G. Jordan, N. Weaver, C. Kreibich, B. Krebs, S. S. Geoffrey M. Voelker, and K. Levchenko, "Pharmaleaks: Understanding the business of online pharmaceutical affiliate programs," in *Usenix Security*, 2012.

[13] A. Jain, "The evolution of cerber ransomware," Apr 2017.

[14] C. Cimpanu, "Cerber ransomware version 6 gets anti-vm and anti-sandboxing features," May 2017.

[15] R. Unuchek, "A new era in mobile banking trojans," 2017.

[16] "Cerber configuration option file." https://gist.github.com/hasherezade/f20d48bf5a894f5c30e898087dee463b#file-404a73ba37fa813e9ab4e6e4f0480706-json.

Acknowledgements

We would like to thank VirusTotal for providing access to their API and Cerber malware samples for our research.