Unveiling the Network Criminal Infrastructure of TDSS/TDL4

DGAv14: A case study on a new TDSS/TDL4 variant.

Manos Antonakakis[‡,*], Jeremy Demar[‡], Kevin Stevens [‡] and David Dagon[*]

Damballa Inc.[‡]
Georgia Institute of Technology, GTISC [*]
{manos,Jeremy.Demar,Kevin.Stevens}@damballa.com, dagon@sudo.sh

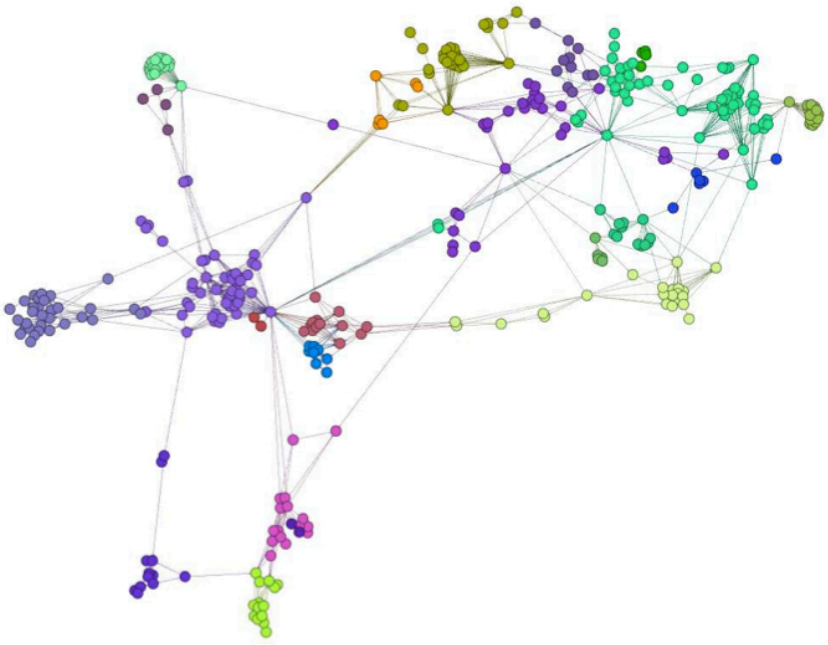

Figure 1: The TDSS/TDL4/DGAv14 extended C&C Network. This graph contain as vertices domain names and IP addresses from the extended TDSS/TDL4 C&C network. The different componet of the graph are collored with standared graph clustering techniques (Chinese Whispers).

1 Summary

In the last few months, Damballa Labs in collaboration with Georgia Tech Information Security Center (GTISC) has been tracking what appears to be a new iteration of TDSS/TDL4. This variant makes use of Domain name Generation Algorithm (DGA) tactics in order to establish its command and control (C&C) communication channel with the C&C domain names, but also to server its Click-fraud activities. The extended C&C network hosting infrastructure spans multiple different networks in Europe, US and Asia. While most of the C&C IP addresses have been associated in the past with illicit operations (i.e., RBN, BitCoin), and have affected hundreds of thousands of victims, we are not aware of a sample available to the security community that matches the network behavior. Despite this, we are able to characterize key parts of the new TDSS/TDL4 variant, its DGA, and most of the victim population. While a binary would provide a more complete explanation of this botnet, we describe in this whitepaper how network-only evidence can be leveraged to defend against the (as yet unrecovered) malware.

Currently, we are monitoring this new TDSS/TDL4 variant—which for simplicity we will refer to as DGAv14 in the remainder of the text—using Damballa's ISP visibility but also using the GTISC sinkhole infrastructure to verify what we infer about its C&C communication channels and growth. As of today we have observed close to 200,000 unique Internet hosts tying to contact the GTISC sinkhole. This number is growing. While a binary sample would let us estimate the total potential vulnerable population, we demonstrate how a network-centric view nonetheless allows us to measure and remediate this malware by working with network operators around the world.

In the remainder of this report, we will briefly discuss the similarities of DGAv14 with TDSS/TDL4 in Section 2. Next, we will continue in Section 3, where we will discuss the passive DNS properties of the network and domain name C&C infrastructure of DGAv14. In Section 4, we will discuss all observations made possible using the sinkhole data we gathered over the last few weeks. Finally, in Section 5 we will discuss the attribution aspects of DGAv14 and we will conclude in Section 6 with lessons learned from the detection and tracking efforts of DGAv14.

2 Related work

We will begin by providing some necessary background on the malware family, which we *infer* is related to the botnet we describe in this report.

Aleksandr Matrosov [8] and Golovanov et al. [8], provided a thorough write-up of all aspects of the TDSS botnet. Briefly, the authors discussed the TDSS malware family and the infection vector that the malware used at the time. Among other things, the authors also discussed the PPI, C&C and SEO aspects of the TDSS malware along with the network infrastructure (`masterhost.ru`) the malware employs. Finally, they show the main C&C protocol and their parameters, which are passed to and from the C&C as a combination of BASE64 and RC4.

DGAv14 shares several similarities with TDSS/TDL4 both on the network and on the system level. In particular, the network C&C infrastructure (as we will see in Section 3) related with DGAv14 has historically served domain names that were TDSS and TDL4 botnets. Furthermore, the C&C communication protocol that the DGAv14 victims employ is very similar to TDSS (as we will see in Section 4). Finally, from a memory snapshot we recovered from one of the hosts that regularly connects to our sinkhole, we discover significant similarity between the code base of TDL4 and DGAv14. Some of the most interesting similarities were the bitcoin module and the ad replacement methods via iframe injection.

However, we also have found dissimilarities from TDSS/TDL. The C&C communication protocol that the know TDSS/TDL4 malware family employs does not match the C&C communication patterns we observe at the GT sinkhole for DGAv14. Based on the network features alone, we believe that DGAv14 is a new malware variant that employs a lot of features from TDSS/TDL4 malware families. However, without binary that matches the exact C&C communication patterns observed both at Damballa's sensors and GT sinkhole, we cannot make any definitive claims regarding the malware family DGAv14.

Perhaps, the closest C&C communication pattern (but still not identical) that we manage to discover was the one describe by Joseph Mlodzianowski in his blog post [1] on TDL4. In this blog the author elaborates on the click-fraud properties of TDL4, the TDL4 BootKit process, the C&C domain names and some aspects of a "Domain Fluxing" activity. The TDL4 BootKit process information described in the blog matches in large extent what we manage to recover from one infected host, however they are not identical(see Section 5 for details). Furthermore, the C&C domains appears to registered under `dik_loren@`[redacted] and `bnzhaa@`[redacted] In our discovery we see that the infected by DGAv14 hosts are reaching out to a different set of domain names under registered under different email handlers `lionel.green@`[redacted] (primarily), `ludbaum@`[redacted] and `hrldedington@`[redacted] Most importantly, based on the domain names stated in the report, we can reconstruct the set of C&C hosts that were used for that version of TDL4. What we see is that from the 25 C&C hosts used by that threat only one host (91.212.[redacted]) overlaps with the extended DGAv14 network we present in this report. To that extent, we consider DGAv14 to be the logical next iteration of the TDL4 threat described in the stellar report from Joseph Mlodzianowski.

Our argument is: while binaries are useful, they are not indispensable. We demonstrate how network data can be analyzed using machine learning techniques, to produce actionable intelligence. Since many binaries "hide" from analysts, using root kits or RAM-only binaries for example, we believe this is an important contribution to this problem. In other academic venues, we have published detailed papers describing how this detection approach works in practice. The remainder of this whitepaper describes this approach at a high level, using DGAv14 as an illustrative example.

3 Passive DNS Analysis

In this section will describe the extended criminal network for DGAv14 C&C domain names. We derived this set of domain names and IPs using passive DNS analysis and the Hidden Markov Model (HMM) domain modeling technique described in [3] (Section 5.3). In other words, we obtained the C&C domain names and remote `RDATA` IPs from the DGAv14 victims. Then we project them in our passive DNS data collection in order to obtain their immediate related historic IPs (RHIPs) and domain names (RHDN). We then selected

[1] `http://sub0day.com/?page_id=365`

Table 1: Extended Criminal Network Infrastructure behind DGAv14.

# IPs	CC	ASN	CIDR	Owner
12	RU	44050	146.185.	PIN-AS Petersburg Internet
10	EU	13237	83.133	LAMBDANET-AS Lambdanet Communications
7	LV	41390	195.3.1	RN-DATA-LV RN Data,
6	RO	42741	94.63.1	CORAL-IT-OFFICE
5	RU	44050	194.11.	PIN-AS Petersburg Internet
4	RO	29568	94.63.2	POSTOLACHE
4	NL	57172	188.95.	GLOBALLAYER Global Layer
4	DE	197043	46.251.2	WEBTRAFFIC Marcel Edler
3	RU	44050	95.215	PIN-AS Petersburg Internet
3	RO	UNK3	94.60.1	COVER-SUN-DESIGN
3	NL	49981	109.236	WORLDSTREAM WorldStream
2	US	19194	63.223.	JOVITA - Sentris
2	RU	UNK2	91.212.2	ZHIRK
2	RU	44050	46.161.	PIN-AS Petersburg Internet
2	RO	UNK1	141.136	SC-MORE-SECURE-SRL
2	NL	50673	46.249.	SERVERIUS-AS Serverius Holding
2	NL	49981	217.23	WORLDSTREAM WorldStream
2	EU	5577	62.122.	ROOT root SA
1	US	30058	50.7.19	FDCSERVERS - FDCservers.net
1	US	174	38.0.	COGENT Cogent/PSI
1	UA	50192	194.247.	UDNET
1	UA	20489	195.234.	KOSMOTEL
1	RU	44780	195.28.	Neryungrinskoye Obschestvo Internet-polzovateley
1	RU	12695	89.208.1	DINET-AS Digital Network
1	NL	47869	94.228.2	NETROUTING-AS Netrouting Data
1	KR	3786	27.255.	LGDACOM LG DACOM
1	DE	8928	91.199.	INTEROUTE Interoute Communications
1	CN	56040	120.197	CMNET-GUANGDONG-AP China Mobile

all the domain names that matched the HMM model for DGAv14. The resulting set of resource records comprises the extended TDSS/TDL4/DGAv14 C&C network. Using the `RDATA` extracted from our passive DNS, we then provide a complete picture of the extended DGAv14 C&C components.

In Figures 2 and 3, we observe the network agility of the extended TDSS/TDL4 C&C network infrastructure. In these figures we can see how the botmasters behind TDSS/TDL4 moved and updated their impressive C&C network infrastructure from `03/03/2011` through `07/18/2012`. From Figure 2, we can see that multiple C&Cs hosts were active at the same time, especially towards the last few months of our analysis period. In Figure 3, we can see that a steady number, typically above 15 distinct hosts, were always active and facilitated C&C operations for the TDSS/TDL4 botnets, with the exception of the first few months of the analysis period in 2011.

In Table 1, we show the extended criminal network behind DGAv14. We were able to identify 85 hosts that appear to be related to the actors behind DGAv14, and were used over the last 18 months. Some researchers have noted through host-based analysis of executing malware that a few of these hosts were connected to malicious activities like TDSS/TDL4. In particular, the host `94.63.` located in Romania is a known TDSS host [2]. Similarly, the node `63.223` is also a known TDSS host that is also associated with BitCoin mining activities [3]. Finally, the nodes in AS44050, have been historically associated with TDL4 and RBN activities [4].

As we can see from Table 1, the three main countries that facilitated the C&C hosting infrastructure for this criminal network are Russia (26 hosts), Romania (15 hosts) and the Netherlands (12 hosts). Currently, we see active C&Cs in the networks of RN-DATA-LV (AS41390, Latvia), ROOT SA (AS5577, Poland) and LAMBDANET-AS (AS13237, Germany).

[2] `http://blog.dynamoo.com/2011/10/some-tdltdss-rootkit-sites-to-block.html`

[3] `http://lists.emergingthreats.net/pipermail/emerging-sigs/2012-January/017250.html`

[4] `http://doc.emergingthreats.net/pub/Main/RussianBusinessNetwork/RBN_IP_List_Update_7-21-2011.txt`

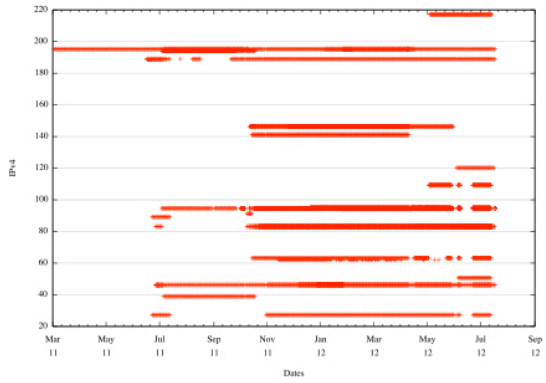

Figure 2: TDSS/TDL4 Network Agility.

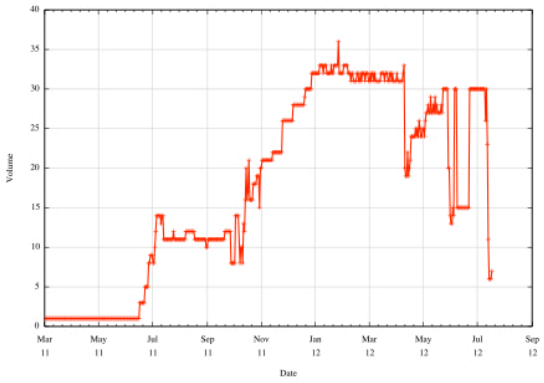

Figure 3: Counting TDSS/TDL4 Network Agility

Table 2: Email Credentials

Volume of C&C Domains	Registration Name	email address	Affiliation
152	Ludolf Baumschlager	ludbaum[redacted]	TDSS
122	Yamir J ayantilal	yantilal@[redacted]	TDSS
63	Harold Edington	hrldedington@[redacted]	TDSS
40	Nmajjd Nbvjaa	bnzhaa@[redacted]	Pending Renewal Status
20	Lionel Green	lionel.green@[redacted]	DGAv14
11	Dik Loren	dik_loren@[redacted]	TDSS
4	Whois Agent	gmvjcxkxhs@[redacted]	Unknown
3	Nick Krachek	kr_niccky@[redacted]	Unknown
3	Unknown	gfgvfddgvdfddsd@[redacted]	Unknown

We identify 418 unique domain names by looking for domains that historically resolved to hosts in the criminal network infrastructure for DGAv14. These 418 are related to the TDSS/TDL4 C&C hosts. Table 2, shows the frequency count of these C&C domain names with respect to their registration information. It worth noting that all the domain names were registered under the `BIZCN.COM INC` registrar. Despite our efforts, BIZCN refused to help us (and to our knowledge, any other security researcher, ever) in any effort to remediate this abuse, or collect any further information regarding these malicious domain names.

We were able to associate the 20 domain name under the `lionel.green@`[redacted] email credential with DGAv14. However, several of the DGAv14 victims also looked up domain names from the `ludbaum@`[redacted] `yantilal@`[redacted] and `hrldedington@`[redacted] until this day. At this point we should also note that the emails under the `hrldedington@`[redacted] are linked with TDSS. All of them matches the DGAv14 DGA pattern (according to the HMM model for DGAv14, see Pleiades [3], Section 5.3), however we do not see hosts making use of the DGAv14 also looking them up.

In other words, this criminal infrastructure facilitated the registration and glue support of new domain names attributed to TDSS email accounts. Figure 4 and 5 shows the direct correlation between the TDSS/TDL4 affiliated accounts with the email address `lionel.green@`[redacted]. This particular email account is highly correlated with the latest active C&C domain names for DGAv14, which we were able to detect using Damballa's ISP-level sensors. At this point it is safe to assume that the actors behind the TDSS/TDL4 could be associated with the DGAv14 botnet. Also, there are some networks like LAMBDANET-AS and RN-DATA-LV that systematically facilitate criminal hosting infrastructure for TDSS/TDL4 and DGAv14 botnets.

We speculate this is an artifact of RIPE's LIR allocation policies, if not tolerance by the network operators. Regardless, these networks illustrate an important invariant property in botnet C&C: no matter how many rebrandings of networks, no matter how many off-shore shell companies handle the traffic, and no matter

Figure 4: Email Aliases to ASNs

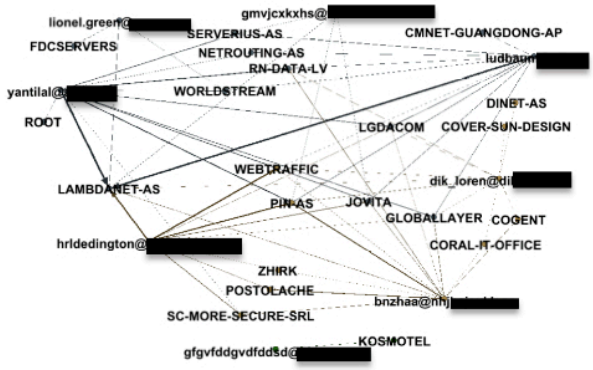

Figure 5: Email Aliases to OrgNames

Figure 6: DGAv14 NXDomain Samples.

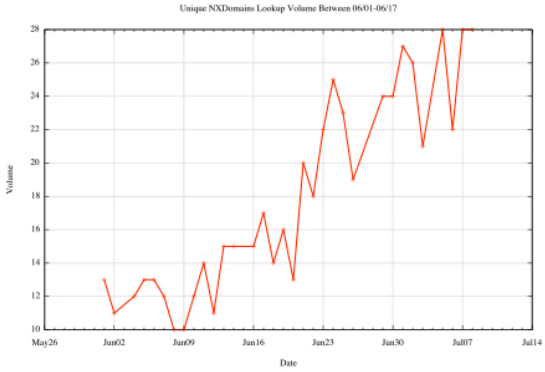

Figure 7: Unique NXDomains volume (single DGAv14 infection) between 07/01 until 07/17.

how many botnets "scrub" C&C traffic through victim proxies, there remains a limited number of networks that will tolerate and enable C&C networks on the Internet. While some operators may argue that individual episodes reflect isolated abuse episodes (cf. McColo), they form a pattern which is illuminated by machine learning techniques. In turn, this lets us associate and cluster domain names, in the zero hour, with their corresponding botnet.

We believe our machine learning approach has the rigor to withstand current and anticipated evasion techniques. This assertion is more fully described in our paper [3], and related works [1, 2]. To give an informal sense of this approach, we further describe the detection of DGAv14's DGA.

NXDomain Analysis: Up to this point we presented the information we were able to collect on the DGAv14 extended criminal network. Next we will discuss how we discovered DGAv14. This discovery was made possible via machine learning techniques [3] on a large pool of ISP-level unsuccessful DNS resolutions from hosts infected with the DGAv14 variant. We were able to employ this particular detection system due to the number of `RCODE=3` messages (NXDomains) the DGAv14-infected hosts generate every day. Currently, we believe that the DGAv14 malware uses a domain name generation algorithm that generates observable NXDomains every day. We should note that due to the lack of a malware sample that directly corresponds

to DGAv14 behavior, we cannot be certain of the properties of the DGA (i.e., seed, DGA cycle, etc.). We are however successfully blocking such domains, and sharing this knowledge with the security community.

In Figure 6, we present a small sample from the NXDomains the DGAv14 DGA generated over time. It appears that every 48 hours a few new NXDomains are generated by the infected hosts. Using this observation, and in collaboration with Georgia Tech Information Security Center (GTISC), we managed to get a glimpse of the botnet worldwide infection levels. We further discuss the analysis of the data collected in the sinkhole in Section 4.

In order to provide some insight on the levels of NXDomains generated by a host infected with the DGAv14 variant we collected all NXDomains generated over a period of 17 days, between June 1^{st} and June 17^{th}. In that period of time, we observed more than 7,000 unsuccessful resolution. At this point we should note that these 7,000 NXDomains are not unique. As we can see from Figure 7, the number of unique NXDomains observed per day is not static. We happen to see different number of unique NXDomains per day between the ranges of 10 and 30. However, we should note that only a small number of 42 overall unique NXDomains were observed between June 1^{st} and June 17^{th}. So the very high number of NXDomain events (over 7,000 NXDomains), and the relatively small number of unique NXDomains, means that the DGA most likely proceeds to unstructured repetitions of even older NXDomains. Finally, another interesting observation is that the infected hosts appears to generate NXDomains (attempt to connect to the C&C) as long as it is connected to the network.

We note that the proper use of statistical techniques (e.g., singular value decomposition) are capable of detecting faint signal in noisy (high SNR) networks. Thus, botmaster evasions that reduce the rate or volume of `RCODE=3` traffic will likely fail or even become more detectable in some circumstances. DGAs, *as a species of abuse*, must generate *some* non-resolvable queries. While automata induction remains NP-hard, the detection of such strings in a DGA is approachable. We claim this behavior will prove detectable even in small (evasive) amounts.

4 Sinkhole Analysis

Table 3: (Left) Sample of DGAv14 Domain Names Sinkholed. (Right) Observed User Agents.

# TCP Con. Attempts	Domain Name	Unique IPs	Unique Domains	UA
21,578,806	udf-szhubu[redacted]	74,793	16	Windows NT 5.1; (W.XP)
18,751,345	ad9btvkonin[redacted]	58,374	16	Windows NT 6.1; (W.7)
18,054,071	v-qk5nvogzt[redacted]	48,428	16	Windows NT 6.0; (W.V)
15,612,548	fjg56xwou[redacted]	56	16	Mac OS X
14,597,317	dklfebjexiab[redacted]	47	16	Windows NT 5.2; (W.S.03)
13,695,584	0ubpccgk[redacted]	6	16	Windows CE; IEMobile
13,174,612	ikh9w-3vd[redacted]	4	5	iPhone; CPU iPhone
10,419,785	nsya1vsb[redacted]	3	4	Linux; U; Android
9,971,869	d8kkkbla[redacted]	17,545	16	Various Windows and Others
9,504,591	qiqxqoedn[redacted]			
8,459,496	sbvv2b59ps[redacted]			
6,574,668	hrai41zpyw7[redacted]			
4,465,259	nihawelnopj[redacted]			
3,925,492	nxadrmwfgp[redacted]			
2,337,619	yhftaw6wxl[redacted]			
679,098	fwjudokrk[redacted]			

Without a binary available to verify our findings, we instead turn to sinkholes. That is, to verify that our DGAv14 cluster accurately described related botnet traffic, we directed some of the victim lookups to a simple sinkhole.

In this section we will discuss some of the most interesting observations made possible from the sinkholed data. We sinkholed the first domain names on July 11^{th}, without attempting to disrupt the entire botnet. Our main goal was to obtain more information on DGAv14, simply because the number of infected hosts

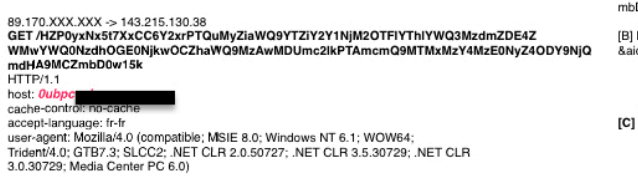

Figure 8: C&C communication.

Figure 9: C&C communication decoded.

according to our ISP visibility was growing and no anti-virus company had any corresponding malware sample. In Table 3, we can see the 16 most significant domain names we proactively registered. We manage to identify and select these domain names using Damballa's ISP visibility and the DGA-based detection system we employ at the ISP levels [3]. The table shows the number of TCP connection attempts that matched the C&C protocol of DGAv14, which we describe next.

In Figure 8, we can see the HTTP GET request from host `89.170.XXX.XXX`, to our sinkhole. The particular host makes the request using the domain name `0ubpc`[redacted]. From the HTTP communication observed at the sinkhole, we observe that the GET request resembles the C&C communication protocol the TDSS [7, 8] malware employs.

However, if we look a bit closer we do not need the RC4 component to decode the information communicated to our sinkhole (or any active C&C host). In Figure 9, we can see ([A]) that the URI is effectively a simply BASE64 encoded message, which can be translated ([B]) to a URI concatenated with a unknown binary prefix.

If we now isolate what appears to be the variables in the URI ([C]), we can see that the C&C communication variables resembles but they do not completely match the TDSS kit described by Matrosov [8] and known TDL4 variants. This clearly shows the string ties between the TDSS/TDL4 malware families and DGAv14.

Given the similarities with the known TDSS and TDL4 malware family we can speculate on the parameter passed as part of the C&C GET commands observed at the sinkhole. To that extent, we can assume that the *clk* value is the version of the C&C kit used to craft the malware sample. Throughout the entire period we operated the sinkhole, 93,9% of the clk values were `Version 4.3`, 4.3% were `Version 4.2` and 1% were `Version 4.1`. The versions `4.4, 3.9` and `3.8` had a 0.2%, 0.07% and 0.006%, respectively.

The *bid* value appears to be a binary identifier, while the *aid* apparently reflect some sort of leasing code, where the binaries that report the same *aid* value have been "delegated" to a particular handler. By examining the relation between the TCP connection attempts and the bid and the aid (sublease) values, we conclude that a small number of subleases have the majority of the bids under their control.

Another interesting observation we made from the sinkholed data was the user agents (UA) reported by the hosts issuing the HTTP GET requests. At this point we should clarify that the UA is very easily spoofed, and without the malware sample for DGAv14 we cannot be absolutely certain whether the malware spoofs the UA fields or not. In the left half of Table 3, we can see the distribution of the most notable UA values observed at the sinkhole. We should note that these UAs were collected if and only if the URI reported by the remote host matched all the C&C protocol parameters presented in Figure 9([C]). According to the first three UAs, it appears that the vast majority of the hosts infected by the DGAv14 malware are Windows XP, Windows 7 and Windows Vista operating systems. Unfortunately, we cannot explain why a small select set of hosts reported UAs related with Mac OS X, or even user agents related with mobile devices such as Windows CE, iPhone and Android. Looking a bit closer on the exact network locations the hosts that reported mobile-related UAs, we saw that the hosts were indeed in networks that support mobile devices.

Next we will present some generic statistics from the sinkholed data. We will begin by presenting the distribution of TCP connections attempts to the sinkhole that matches the DGAv14 C&C protocol. As you can see from Figure 10, we average almost 50,000 such TCP connection attempts every 15 minutes.

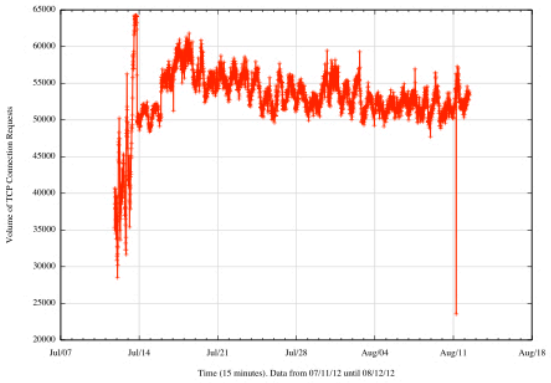

Figure 10: Volume of TCP connection attempts.

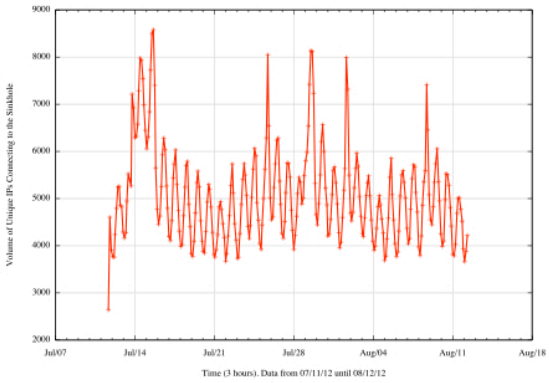

Figure 11: Volume of unique IPs.

Interesting observation is that the connection attempts are not declining, which means the domain names we registered are still in the malware's C&C communication cycle. Again, due to the lack of a malware sample we cannot provide an explanation as to why this is happening. We should note that the single decrease on the TCP connection attempts was due to a system issue at the DNS authoritative server for the sinkholed domain names and not to the botnet itself.

Currently, we have seen 199,256 unique IPs that have contacted the sinkhole with a HTTP GET request that matches the DGAv14 communication protocol. In Figure 11 you can see the number of IP addresses attempting valid (according to the C&C protocol) HTTP GET requests from the sinkhole over a time window of 3 hours. We can see that the diurnal patterns is not very clear. This is happening because the IPs we see are not distributed over all continents, rather they are highly concentrated in North America and parts of Europe. In particular we recored so far 75,412 (37.8%) IPs from the US, 70,678 (35.4%) IPs from Germany, 17,145 (8.5%) IPs from Great Brittan, 11,015 (5.5%) from Canada and 10,480 (5.2%) from France. These five countries comprise the bulk (92.6%) of the infections. We are in constant discussion with the ISPs, private companies and others affected by DGAv14 in an effort to improve the situational awareness and remediate the hosts affected by the DGAv14 botnet.

4.1 Ad-click Replacement C&C Protocol

As we discussed in the earlier parts of Section 4, the main C&C communication protocol used by this DGA is simply encoded (Base64). This was not the only type of traffic we saw at the sinkhole. A small portion of the domain names (See left portion of Table 4) that we obtained received mix traffic — that is — the C&C communication described in Figures 8 and 9, but also encrypted traffic that resembles the traditional TDSS C&C communication tactics. This second C&C communication protocol, however, is related to the Ad-click and BlackHat-SEO illicit activities by the malware.

Next we describe this Ad-click protocol. Looking in Figure 12, we can see what the decoded and decrypted HTTP GET request for the domain name `cudkk`[redacted] looks like. Briefly, by simply decoding observed HTTP GET request and then decrypting the derivate HEX output (using RC4 and the domain name as the key), we obtain the plain text communication (Figure 12(I)). Despite the fact that this encryption tactics have been use in the past by TDSS [5], this is a new usage scenario for TDSS/TDL4 malware family.

Briefly, in Figure 12(II) we can see the six main components of the Ad-click replacement protocol. The `Version` parameter seems to be the same (1.8) across all victims. (As noted in Section 5, the string `"1.8|%s|%s|%s|%s|%s|%s"` is hardcoded in the memory dump.) The `BID` looks to be the same on both C&C communication channels. In other words, we will see a remote IP reporting the `BID` value in the C&C protocol described in Figures 8 and 9, and then the same IP information and `BID` value will appear

[5] `http://resources.infosecinstitute.com/tdss4-part-2/`

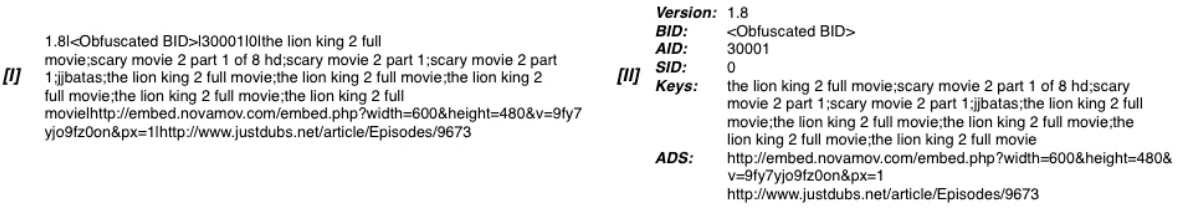

Figure 12: Ad-click C&C communication protocol.

in the Ad-replacement data. The `SID` value appears not to be really used, according to our data, since it is constantly zero. Lastly, the `Keys` and `ADS` parameters are used to report back the search terms and the Ads being replaced by the malware. The `Keys` parameter is a semicolon delimited field of recent search terms. As shown in Figure 17, the malware would capture up to 10 of the most recent user searches, and send them to the C&C via an encrypted GET parameter path.

We believe that the Ad-hijacking module in this particular TDSS/TDL4 variant uses this encrypted C&C communication channel so the botmaster can control and provision the status of the click-fraud campaign. This permits the botmaster to sell off victim traffic to other blackhat SEOs and monetize the entropy and browsing sessions harvested from users. In the right portion of Table 4, we observe the top domain names from the URLs recovered for the sinkhole. This indicates that Ads with URLs under these domains are being targeted by the particular TDSS/TDL4 variant. Over 1,461,213 instances of Ad-replacement events have been recorded until the writing of this report.

Table 4: (Left) Sample of DGAv14 Domains Sinkholed part of the Ad-click C&C Communication Cycle. (Right) Domains in the Ads being replaced.

# TCP Con. Attempts	Domain Name		Occurrence Volume	Domain Name
369,189	1jndmf93[redacted]		256,632	facebook.com
367,729	cudkkm[redacted]		175,981	doubleclick.net
359,000	anz7sj[redacted]		109,257	youtube.com
349,328	vpmybke[redacted]		100,513	yahoo.com
5,315	3kish71ei[redacted]		84,453	msn.com
4,752	h3avgk[redacted]		35,782	google.com
1,128	9gkftyw[redacted]		33,503	jeetyetmedia.com
887	rvplrlw[redacted]		32,608	atdmt.com
884	f0ix-fvlh[redacted]		30,611	adnxs.com
858	yurballv[redacted]		28,078	exoclick.com
838	grnuadbo[redacted]		26,925	rubiconproject.com
289	dkhshr[redacted]		26,572	pubmatic.com

4.2 Host C&C Interarrival

Most anti-DGA analysis focuses on the periodicity of domain generation. Some botnets e.g., Torpig, use domains for different periods of time. That is, some domains are used repeatedly for a month, others for only a week, and some only for a day. Without access to a binary, is it still possible to infer the periodicity of the DGA, by reference to simple spectral plots. Since victims often have skewed clocks, the sinkhole traffic is a mass of inter-leaved C&C attempts.

We can make sense of this traffic by noting the inter-arrival period between hosts, as well as their containing CIDRs. This counts are of course skewed by NAT and DHCP churn. But we can estimate the amount of skew by comparing the per-host inter-arrival to the inter-arrival of victims in the same CIDR. Figure 13 shows the time in seconds between victim arrivals in the sinkhole, sorted by both host and CIDR,

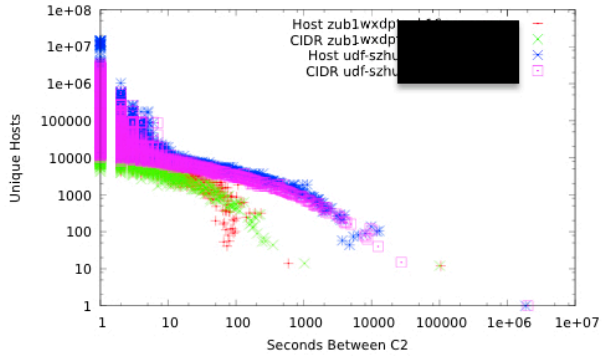

Figure 13: Per-Host and per-CIDR inter-arrival times, for most and least popular C&C domains.

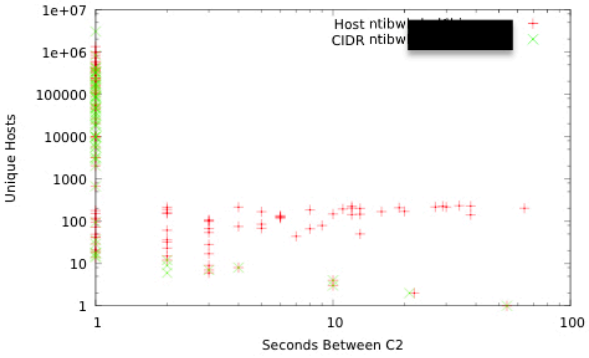

Figure 14: Limited use C&C domain.

for the most popular C&C domain (`udf-szhu`[redacted]), and one of the least popular C&C domains (`zub1wxdp`[redacted]). The separation between the Host and CIDRs views shows the impact that NAT and DHCP churn has on this spectral measurement. The actual DGA period must be the larger (in seconds) of the two.

The graph in Figure 13 also show that different domains are used for different periods. Although there is some skew in both, clearly the domain `udf-szhu`[redacted] is used for a longer period ($\sigma = 250713$, or ≈ 69 hours.) We expect that, when a binary is found for "DGAv14", a tiered DGA will be discovered (similar to Torpig), which uses domains for different lengths of time. Of course, one cannot say for certain without access to the malware. But this insight lets one plan DNSBL blocking actions, allocate resources for network defenses, and prioritize different domains for remediation or research.

Figure 14 also shows the inter-arrival period for a very short-lived domain. Almost *all* the hosts reaching `ntibw`[redacted] did so within a few seconds, and none visited the C&C for this domain more than a few times, usually taking no more than a minute total. We speculate this domain was intended for a short-lived campaign (e.g., a PPI, click-fraud, or fake AV campaign), that has little periodic-recurring traffic. Without access to the malware, we cannot be certain. However, calculating the period for host inter-arrival lets identify this domain as a one-off "event" for the botnet, as contrasted to the long-term recurring visits for other domains.

Traditional time series plots of query volumes will show some details, particularly those of short-lived or single-use domains, but will obscure the DGA periods for longer-lived domains. Consider for example Figure 15, which shows the same data graphed as traditional timeseries, with sinkhole volume over time. For domains that last more than a single day, diurnal user patterns and time zone skews will obscure important details. This is particularly true for this botnet, which has a highly curated victim population, making traditional botnet time series analysis [5] difficult.

4.3 User DNS Paths

Operating a sinkhole at GTISC also gave us the opportunity to collect *iterative* DNS information about the victims, and compare this to the DNS stub visibility we use for statistical analysis. Since we operated name-servers for the C&C domains, we were able to selectively manipulate resolution sessions, and learn more about victims' DNS settings.

We first created a multi-threaded "302 only" responder that answered all incoming victim http sessions with an RFC 2616 §14.30 "Location" response [6], directing hosts to a wildcarded child label generated from their HTTP TCP session. The logic of this approach is similar to the IPv4 based DNS scanning demonstrated by others [4]. Figure 16 shows a conceptual view of this sinkhole design. Because these records had no cache value, we appropriately chose a TTL of zero.

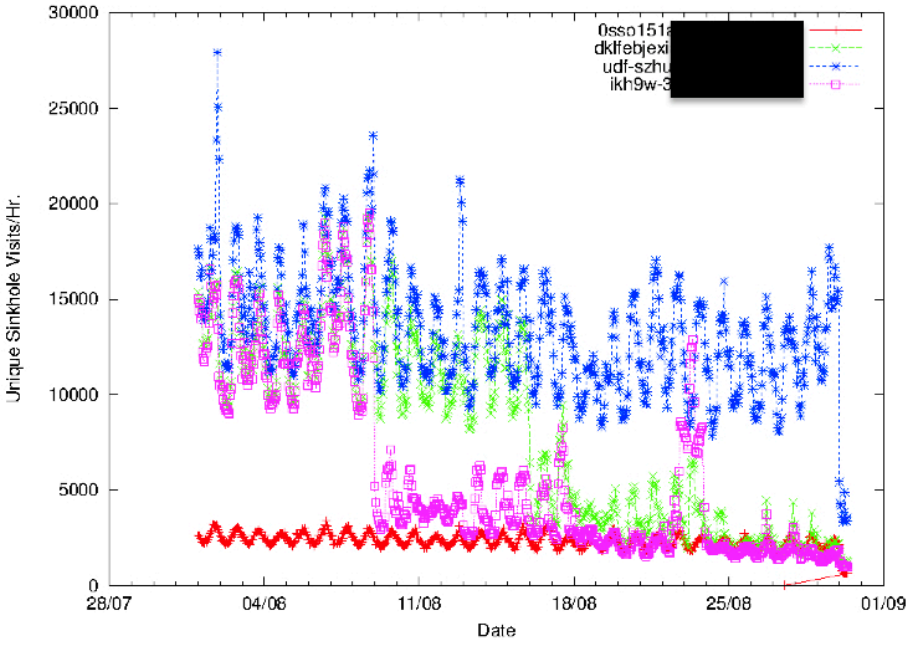

Figure 15: Basic Time Series Analysis of C&C Domains.

By forcing the victims to resolve unique child labels, we capture several important properties of their DNS settings: DNS path, RTT latency (e.g., for users that have alienated their DNS resolution to distant networks), use of so-called Cloud DNS services, SPR, use of DNS-0x20, and other DNS health metrics. Since the child label is unique it also prevents malicious query injection (e.g., spoofed queries to add noise to our observations).

Figure 18 shows the DNS resolution paths taken by victims, for a given 15 minute sample of data. Red dots indicate DNS resolvers, and blue lines connect the victims from the geolocation to a particular resolver. To make the graphic easy to see, Figure 18 only plots cross-domain resolution. That is, it only shows users who have DNS settings configured to resolve outside of their network. (Since most users apply the local DNS settings of their network, the graph would otherwise have very few blue lines, and a sea of red dots for local resolvers.)

We noted that many users rely on so-called Cloud DNS services that offer security filtering. Since these DNS services were still permitting users to contact the sinkhole, we contacted as many of them as we could find, and offered a list of DGAv14-related domains.

The DNS data and the 302 sinkhole let us make some observations:

- First, we noted that many networks performed DNS lookups and did so regularly and in synchronization with other victim groups, but were able to block outbound HTTP connections. Very likely, such networks had DPI or edge-based protection systems or HTTP proxies that spotted TDSS-like URI signatures in the TCP stream. However, these networks were evidently not filtering DNS for known TDSS domains.

  When we switched these C&C lookups from `RCODE=3` to actual `RDATA`, many edge devices "woke up" and saw traffic they could identify. Thus, merely by running a sinkhole, we helped many 3^{rd} party networks better use the edge network protection/detection systems they deployed. In our future works, we will study how to better manage `RCODE=3` traffic in protected networks and use split horizons to leverage networks that perform HTTP filtering.

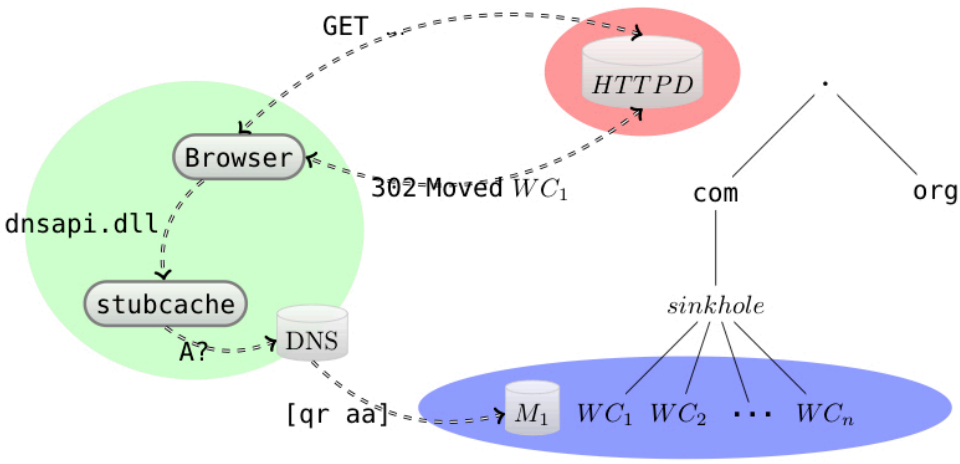

Figure 16: Conceptual view of 302 Sinkhole responder. Users were given `302 Moved` responses to wildcard zones unique to their http session. By monitoring the authorities for these wildcard zones for `[qr aa]` responses at M_1, we create a table mapping all botnet victims, DNS paths, and browser properties.

- Second, we noted that a small number of networks did not perform basic DNS hygiene (e.g., port randomization), or used unpatched old resolvers, or used open recursives.
- Third, just based on the DNS traffic alone, we were able to identify numerous researchers. We note that RFC1262 says little about how security researchers are to inter-operate or cooperate, and welcome the discussion about how sinkhole management can be cooperatively managed.

5 Attribution

In order to confirm the malware behind the activity we began searching for a binary. Our goal was to find an example that was communicating with the sinkhole, we found an example at a customer who provided us with a memory dump of the injected process. Because this was a memory dump no useful hash is available to identify the sample. By viewing the process dump we were able to determine that the injected process was a version of the TDSS/TDL4 cmd.dll module. The process was injected into the systems "explorer.exe".

We were able determine that TDSS/TDL4 had installed itself to its standard locations seen in Figure 19[I]. We were then able to extract portions of the cfg.ini file from the dump and confirm that the sections were found matched what was seen in the sinkhole. The `[main]` section (Figure 19[II]) didn't have any values that appeared to be major changes from what has been reported in the past [6].

The `[cmd]` section (Figure 19[III]) however had some values that stood out. The version in this section is higher than the 0.28 that was seen in reporting that most closely matches the other values from previous reports. We were also able to locate parts of the BitCoin mining config. We couldn't determine the domain it was using to communicate to the BitCoin tracker but we did find the username and password arguments passed to the BitCoin mining module. The format strings for various communication channels were found can be seen in Figure 19[IV]. The strings (1) and (5) were seen in use for the Click-fraud abuse.

One interesting item that was found in the memory snapshot was related to the BitCoin mining software controls. The purpose of this code seems to facilitate BitCoin mining activities when there is no user present. Searching for the some strings found in the data we found leads to a Russian language programming blog [7]. There is no indication that this programmer is involved in any way but the strings were method names,

[6] `http://www.securelist.com/en/blog/559/TDSS_Bitcoin`
[7] http://blog.yastrebkov.com/2011/07/isenslogon-atl.html

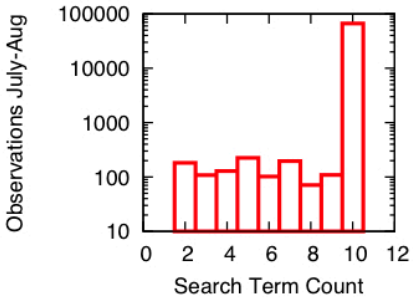

Figure 17: **Number of Search Terms Exfiltrated.** Most victims had their ten most recent search queries uploaded.

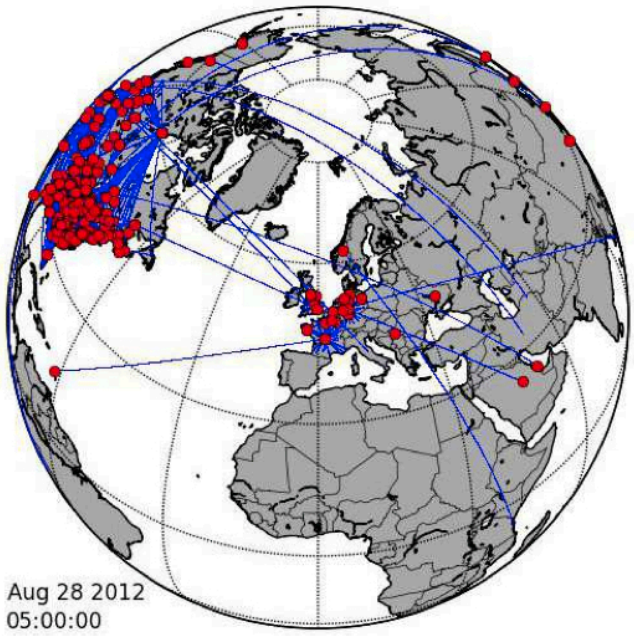

Figure 18: **DNS Paths of DGAv14 Victims** Red dots indicate recursive resolvers used by DGAv14 victims, blue lines indicate the start of the user's DNS resolution path, ending in a given resolver.

which can be seen at Figure 19[V]. Besides that blog, no further references to these strings were available in the Internet.

The primary use for the binary seems to be related to Ad-clicking and Click-fraud activities. In the memory space we were able to find the code used to inject the actual Ads. The actual iFrame injection code used can be seen at Figure 19[VI]. Finally, the strings in the binary used for this abuse can be seen at Figure 19[VII].

We believe that the call home requests we are seeing from the infected hosts to the DGA domains is primarily part of the Click-fraud traffic. We also believe that that the DGA domains serve the Click-fraud activities in the same concept as TDSS. Some of the email addresses that we have seen used in DGAv14 were also used to register domains that were used with older versions of TDSS. This helps determine that the same criminal groups are involved in both TDSS and DGAv14. The link between the email credential used significantly adds to the evidence that this threat is most likely a new iteration of TDSS.

The actual version of the Click-fraud module that we are seeing in the GET requests to the sinkhole are versions 3.8 through 4.4. It is interesting to note that we have seen similar requests, as stated earlier in the paper, in previous versions of TDSS. These versions used click module versions prior to 3.8 and had less parameters in the GET request. In Figure 19[IV] you can see line 2 matches mostly with:

```
ver=4.2&bid=<Obfuscated SHA1-like
Value>&aid=50018&sid=0&rd=1307260520&eng=www.bing.com&q=celebrity.
```

Which is a real Base64 decoded request that was sent from a TDSS infected machine and comes from the paper [8]. The difference is now that the type of OS, from the infected machine, is being reported x86=%d(32 or 64) and some other field that has a string value rf=%s. If this holds true then line 2 is what is searched for

[8] http://research.microsoft.com/en-us/um/people/saikat/pub/sigcomm12-clickspam

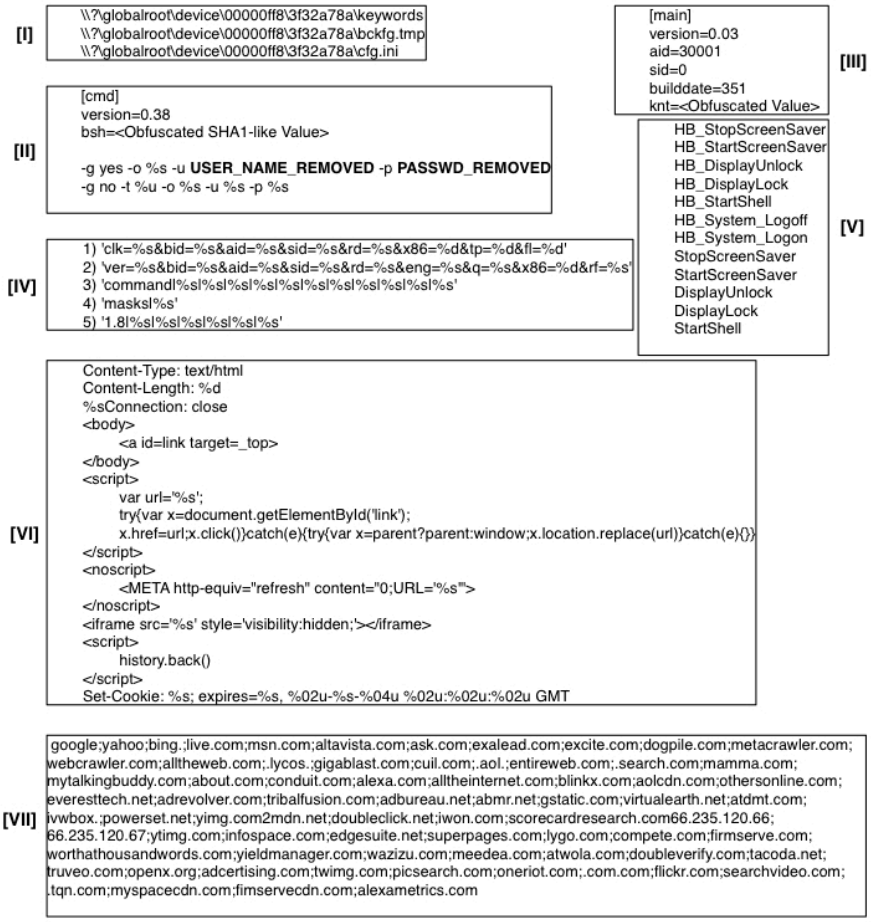

Figure 19: Forensic Evidence from Memory Snapshot

and is sent to the C&C server. Then a XML should be returned based on the recent research by Microsoft. The results, in line 5, is probably what was clicked on and being returned to the ad server for book keeping by the botmaster. An example of this is Figure 12.

6 Lessons learned

The cat-and-mouse game between security researchers and botmasters will continue, so long as network defenders respond with incremental improvements. In this whitepaper, we describe a *next-generation* improvement over existing defenses.

Using statistical properties of botnet behavior, we are able to detect stealthy DGA-based botnets in ISPs. Moreover, we are able to map key properties of the botnet (C&C domains, DGA period, victim DNS and browser properties), before any malware sample has been collected. Indeed, to the best of our knowledge, no security company has a copy of the malware used by DGAv14 TDSS/TDl4 variant.

Our techniques are robust against evasion, and we believe they target *an invariant* of the botnet, since at some point all victims must make some contact with the botmaster (e.g., to install additional malware, to be sold off into a fake AV, ad replacement campaign, BlackHat SEO etc.). Botmaster efforts to diversify

domain ownership, change C&C hosting infrastructure or scrub traffic through proxies will only yield *more signal* that machine learning can leverage. Additionally. botmaster efforts to improve binaries, hide from host inspection and defeat host-based defenses are irrelevant, since we do not require malware samples of any kind. We claim this is a significant change in how botnets can be managed on protected networks.

We list several important lessons learned in this exercise:

- Sinkhole operation remains a complicated topic, with numerous policy concerns. We implemented a simple sinkhole to verify our findings. Beyond that, however, we've designed our system to work *without* sinkhole input.
- The security community lacks clear guidelines about how researchers share information and coordinate remediation efforts. We believe an update to RFC 1262 is in order, so that security research and remediation is transparent, and non-harmful to 3^{rd} party networks.
- We note that, by virtue of our returning `RDATA` for selected C&C domains, we likely elicited a better response from DPI devices that now had TCP traffic to inspect. We also are taking steps to share data gathered by our sinkhole. We believe that the security community must improve the science of remediation. Damballa will continue working with ISPs and customers to remediate infected hosts. However, Internet public health solutions must be extended to all victims. We call for a scientific study of remediation techniques (victim notification, blocking, etc.) to evaluate which approach works best.

References

[1] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a dynamic reputation system for DNS. In *the Proceedings of 19th USENIX Security Symposium (USENIX Security '10)*, 2010.

[2] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, and D. Dagon. Detecting malware domains in the upper DNS hierarchy. In *the Proceedings of 20th USENIX Security Symposium (USENIX Security '11)*, 2011.

[3] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware. In *the Proceedings of 21th USENIX Security Symposium (USENIX Security '12)*, 2012.

[4] D. Dagon, N. Provos, C. P. Lee, and W. Lee. Corrupted DNS resolution paths: The rise of a malicious resolution authority. In *Proceedings of Network and Distributed Security Symposium (NDSS '08)*, 2008.

[5] D. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *Proceedings of Network and Distributed Security Symposium (NDSS '06)*, 2006.

[6] R. Fielding, J. Gettys, J. Mogul, M. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – http/1.1. `http://tools.ietf.org/html/rfc2616`.

[7] S. Golovanov and V. Rusakov. TDSS. `http://www.securelist.com/en/analysis/204792131/TDSS?print_mode=1/`, 2010.

[8] A. Matrosov. TDSS part 1 through 4. `http://resources.infosecinstitute.com/tdss4-part-1/`, 2011.