Detecting Globally Malicious Events with Local Records: A Case Study

Max Potasznik
Department of Computer Science
University of Maryland
College Park, MD 20742
+1 301-405-1071
maxp@cs.umd.edu

ABSTRACT

On or about August 25th 2013, the name servers supporting the country code Top Level Domain (ccTLD) ".cn" were attacked and brought offline[2, 6–8, 11]. As local DNS caches expired, this attack eventually affected the internet traffic of most users attempting to reach Chinese websites because the authoritative DNS servers for those sites ceased working. While the attack itself was widely reported in tech circles, there are very few technical details publicly available about the attack. In this paper, we follow a series of deductive hypotheses: each leading closer to the actual malicious actors and eventually revealing the nature of the attack on the Chinese DNS to be a dictionary based NXDOMAIN attack.

Keywords

NetFlow, DNS, DDoS, DNS Amplification, NXDOMAIN

1. INTRODUCTION

This paper follows a chain of hypotheses beginning very generally and following the conclusions offered in the data at each step to deduce more specific conclusions and further hypotheses. Before delving into the questions themselves, we'll discuss the systems we're investigating as well as the methods we use to investigate them.

1.1 NetFlow

Network flows are an aggregated means of monitoring traffic which traverses specially configured routers on a network. In contrast to full packet capture utilities and many IDS/IPS systems, flow based tools look only at IP layer header information. This information is then aggregated according to protocols so that all traffic between two hosts in a given direction using a single protocol is considered one flow. The actual details contained in a flow record depend on both the vendor implementation and the level of logging determined at the router, however UMD uses Cisco's proprietary NetFlow v9[9] and we'll refer to our flows henceforth as NetFlows. The full range of flow recording is detailed in the IPFIX specification[10].

Given the increased use of payload encryption and increased privacy concerns from network users as well as its light weight compared to full packet tools, flow based tools continue to see both a rise in use in both a network administrative and security setting. Our group's previous work has involved using NetFlow to detect malicious browser redirects[13].

1.2 Domain Name System

The Domain Name System (DNS) is, in many ways, equivalent to the internet's phone book: if you want to know a web address's number (IP address) you query DNS and receive an answer. When a part of the system goes down you can still access an IP address provided you already know it, but you can't look it up. This is the same as if you lost a phone book; you may still make a call but only if you remember the number to dial or have it written down.

This simple analogy captures the essence of DNS functionality but not the details. In reality, DNS is broken up into a hierarchy[21] of subdomains beginning at the "root." Below the root are Top Level Domains (TLDs) such as ".com," ".gov," and ".cn." Below TLDs are further divided subdomains such as "google," "whitehouse" and "baidu." Lastly, individual services at each subdomain may have their own record in the DNS such as "www" for web servers or "ftp" for file shares.[20]

The beauty of DNS is that an individual DNS server need only store the DNS records for the websites and/or domains that lie below it in the DNS hierarchy. If a local computer queries for a website's IP address which the server doesn't know, it can recursively call through the DNS tree beginning at the root and progressing deeper until a name server returns the requested record. Once the local server receives a record, it can store this response locally in a cache for use the next time that website's IP is requested. The danger here is that if a particular name server is unavailable, there is a chance that records for which it is responsible for (i.e. for which it is the authoritative name server) will be unavailable unless cached locally at other servers.

1.3 DDoS on .cn

According to numerous reports[2, 6–8, 11], the registrar responsible for running the ".cn" ccTLD was attacked around midnight local time on August 25, 2013. Reports indicate that the attack was the "largest ever"[2, 11] and was able to knock the DNS out for the world's fourth largest ccTLD for periods up to six hours at a time. Apart from the astonishing lack of details in any of the reports and a surprising lack of follow up reporting on the actual events, what makes this attack surprising is that the CNNIC, the DNS operator for ".cn," is well regarded in the community and had recently been designated as a backup registry for any new gTLD that fails or goes out of business.[17] The success of an attack on a sophisticated ccTLD operator which provides DNS for roughly 10 million web sites indicates that the attack was either complex, or extremely large scale. Since quote

from the network operator indicate that they suspect the latter, we begin our investigation there.

The remaining sections detail the investigation of the attack including our hypothesis and analytical approach at each step. Where we include data other than NetFlow data captured at the border router of UMD, it will be noted. The final sections offer our conclusions and potential for future research.

2. CASE STUDY

Our first assumption is that the attackers perpetrated the Distributed Denial of Service (DDoS) against .cn were focused more on attaining a high attack volume and less on evading detection. This assumption is borne from the previously cited news reports indicating an unsophisticated attack with high volume. As such, we believe it is a good candidate for exploration via NetFlow because NetFlow is particularly useful for looking at traffic volumes and other aggregated metrics.

2.1 Can we find the attack?

A1: *If the attack is large enough to take down the world's 4th largest ccTLD, it likely had a global footprint.*

Assumption A1 encodes our suspicion that any large scale DDoS likely involves the use of a custom built or contracted global botnet. Indeed large scale botnets, specifically built for launching DDoS attacks (or any other purpose) can be purchased quite readily[14]. While the ability exists for attackers (or botnet purchasers) to localize their botnets in specific locations, there are many reasons why an attacker would want a globally distributed botnet as opposed to a focused one, especially as it relates to DNS. For instance, if an entire botnet falls in a specific subnet and a DDoS victim determines the attacker's subnet, they will be able to swiftly block the attack. On the other hand, a more distributed botnet will be more difficult to block because the attack will appear to come from everywhere.

A2: *If a network event is globally distributed, a subset of that event's activity will be evident in the UMD NetFlow records.*

Assumption A2 stems from our knowledge that the UMD network is generally permissive and rarely blocks traffic or removes infected hosts. Based on our conversations with network administrators and our previous work on the campus network, we are confident that there are numerous hosts on the campus network belonging to a variety of malware variants.

From our assumptions about the UMD network and H1 and H2, we get our first experimental hypothesis:

H1: *Since we should see evidence of global events in UMD NetFlow records and the attack on .cn was likely of global scale, we should see evidence of this attack in the UMD NetFlow records.*

2.1.1 Finding the Attack

The first step in our case study on the Chinese ccTLD is to determine if we can see anything which resembles attack behavior in the campus NetFlow. To do this we aggregate total traffic volume in bytes for each hour of the two weeks preceding and following the attack. Our assumption here is that whatever the baseline level of activity on the UMD campus to the CNNIC name servers, the volume of traffic during the attack will be much larger and thus easily spotted. Figure 1 shows that our suspicion is correct as the large spike in traffic is several orders of magnitude higher than the mean traffic volume for all other time periods in the sample observed. Additionally, by closely looking at the times associated with the spike in Maryland traffic, we can see that our traffic to the victim IPs corresponds directly to the times of the reported outages (Figure 2). These times are midnight August 25th (noon August 24th local time) and 6 hours later.

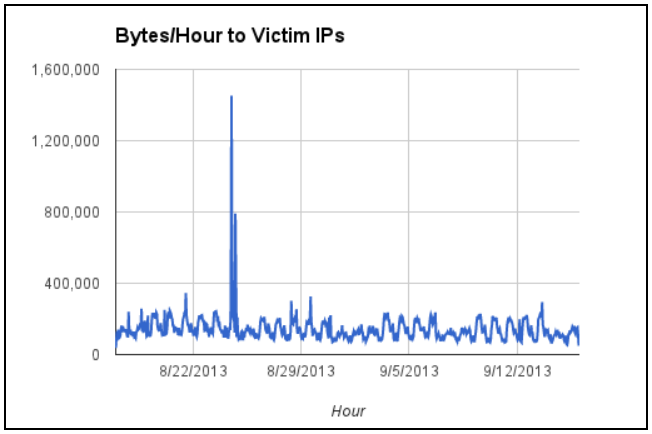

Figure 1 - The large spike is an order of magnitude above the mean traffic volume for non-attack periods.

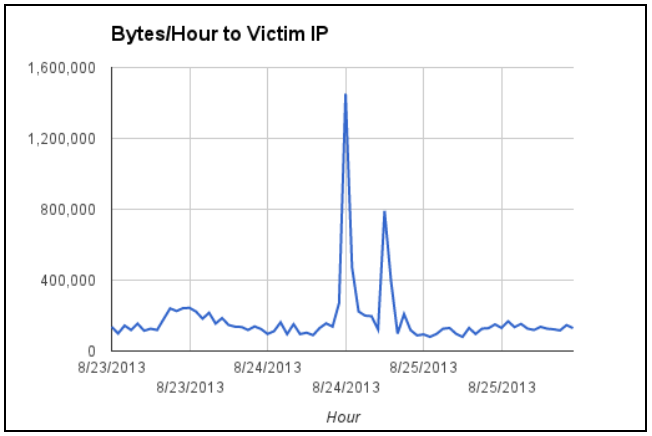

Figure 2 - Traffic spikes in the UMD traffic occur at precisely the reported attack times in China

The figures above show that traffic from UMD is in some way affected by the attack on the Chinese ccTLD. We've shown that during the attack timeframe, network behavior at UMD significantly deviates from normal behavior and thus validated our hypothesis that evidence of the global event will be visible in our NetFlow records. However, it is important to note that we haven't shown who is responsible for the change in behavior on the network, what type of DDoS attack this might have been or even if the observed traffic is part of the malicious activity or a side-effect of it.

2.2 Can we find the attackers?

A3: *The number of hosts on the UMD network infected with whatever malware controls the botnet which launched the DDoS is small relative to both the number of uninfected hosts and the number of hosts regularly contacting the victim IPs.*

Assumption A3 summarizes our suspicion that while it is likely some members of the UMD network participated in the attack, their numbers are likely small relative to the size of UMD's network. If the number of participating infected hosts was a significant portion of the local network, their combined illicit

activity may have caused local network disruptions or caused network administrators to take action, neither of which occurred. Further, we assume that of all hosts who contact the victim IPs very few will actually have been part of the attack.

Taking this assumption into account, we formulate our second experimental hypothesis:

H2: *Since the number of UMD hosts contributing to the attack is likely much smaller than the number of UMD hosts which legitimately communicate with the victim, we should be able to locate a small number of talkative UMD hosts that are likely malicious.*

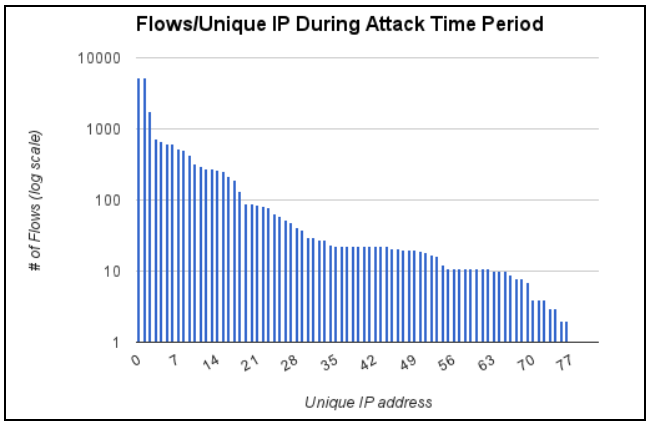

Figure 3 - During the time of increased traffic two distinct IPs account for a majority of the traffic.

2.2.1 Finding the Attackers

In order to find suspicious hosts on the UMD network, we isolate our search to the attack time period and count the number of flows from each distinct IP which communicated with the victim during that time. Figure 3 shows the result of this aggregation with a log scale on the vertical axis as the majority of hosts sent fewer than 50 flows but the talkative hosts each sent over 5000. For privacy related purposes, we have assigned an index number along the horizontal axis for each IP encountered in our dataset.

2.2.2 Who did we find?

In some sense, the figure above indeed validates H2 because we were able to identify some highly talkative hosts, but it also reveals some flaws and hidden assumptions in our models. Some deeper digging related to the specific, talkative UMD IPs reveals that we have in fact detected UMD name servers; part of the DNS hierarchy themselves! Considering that a crucial part of DNS functionality involves communication between DNS servers, it isn't surprising that the most of the network traffic from UMD is from UMD DNS servers. This finding suggests that there was an implied assumption in H2, namely:

A4: *The DDoS perpetrated against the victim was a direct DDoS with no indirection (i.e. the botnet hosts communicate directly with the victim).*

Although we did partially validate H2 by finding a small number of hosts, we discredited A4 by showing that UMD hosts did not directly communicate with the victim in numbers sufficient to constitute an attack. Instead, any bots residing on the UMD network must have used the campus DNS servers as a launching pad for their attack. We also confirm that the DNS servers were responsible for the rise in traffic volume during the attack by monitoring only the name server IPs and observing the same rise in traffic there as seen in the set of all IPs. The finding that the UMD name servers are in fact involved in the attack requires us to consider different attacker models.

2.3 Is this a DNS Amplification attack?

2.3.1 DNS Amplification

A DNS amplification attack is a specific type of DDoS which is particularly effective against DNS servers and has been documented being used in the wild as far back as March 2006[16]. In a DNS Amplification attack, each host in a botnet sends a request to an open recursive DNS server on the internet. The requests are specifically tailored to ask for records which are extremely large files. These files can either be deposited maliciously at other previously exploited DNS servers, or they can simply be large "zone" files which DNS servers to keep track of the larger DNS hierarchy. In either case, the attacking bots request the large file but spoof their own IP to be that of the victim so that the response (in this case the very large file) is sent to the victim.

This attack is effective against DNS servers because the attackers also spoof their port to be port 53, which is where the DNS service runs. From the victim's perspective, it appears as if they are receiving a response to a request they never issued. Because the malicious UDP datagram is large enough to fragment in transit, the victim's network stack must spend resources to reconstruct the full datagram from several IP packets and only then can it discard it. As more attack datagrams come in, the DNS servers system resources are fully consumed reconstructing fragmented packets and eventually it is unable to process valid request traffic. While several approaches to detecting or mitigating these attacks have been proposed[16, 19], it is unclear how effective the approaches are or how widely they have been adopted in industry.

2.3.2 DNS Amplification Attack Signature

If we assume that the attack is a DNS Amplification attack, then in order to fit our observations to the model, we must also assume that (**A5**) *the UMD name servers are serving as the naïve, open-recursive servers from the attack scenario: blindly responding to requests forged to look as if they are from the victim.*

Given A5, there are at least distinctive behaviors which we might be able to find evidence for in our NetFlow records. The first is a large number of connections *from* the victim during the attack time. This might be counterintuitive but stems from the fact that the malicious requestors spoof their own IP to look like the victim. If any bots from outside the MD campus attempt to use the UMD name servers as their DNS, when their traffic crosses the border router (where our NetFlow collector is) it will be recorded as coming from the victim's IP. We attempted to look for this behavior, but instead found that traffic from the victim dropped to zero during the attack indicating that the attack succeeded. While this is a negative result, it does not rule out the possibility that UMD hosts still spoofed the victim's IP while making requests. If this were the case, we would not see this traffic as it never crosses the border router and, as such, is missing from our NetFlow records.

The second distinctive behavior which would support the theory that the attack is a DNS Amplification is an increase in bytes per packet during the attack time frame. The "amplification" in the attack's namesake comes from the fact that well behaved DNS traffic is both small and proportional to the request size whereas a

response to a request from this type of attack is much larger and disproportional to the size of the request. Thus, the attacker utilizes very little network resources making the request, but his request size is *amplified* many times in the response sent to the victim. This characteristic of the attack can be summed up in the following hypothesis:

H3: *Since we are assuming the attack is a DNS Amplification attack, there should be a significant rise in bytes/packet during the attack timeframe.*

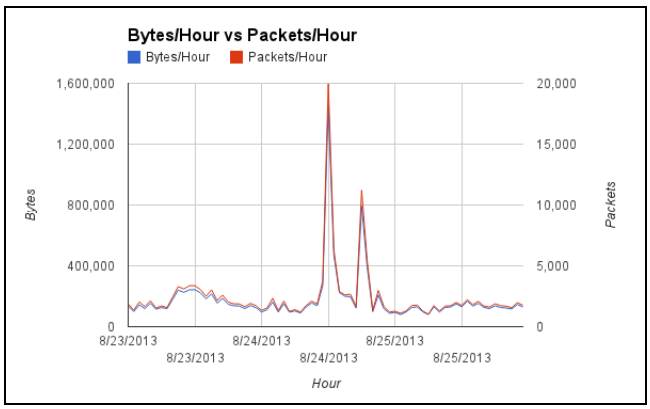

Figure 4 - Packets per hour overlaid on top of the bytes per hour from Figure 2. No rise in bytes per packet.

2.3.3 Dismissing DNS Amplification

Unfortunately, we are able to dismiss H3 due to the findings summarized in Figures 4 and 5. Figure 4, above, overlays the packets per hour on top of the bytes per hour seen in Figure 2. If this were a DNS Amplification attack, we would expect to see the increase in bytes per hour outpace the increase in packets per hour. While it is hard to tell in this figure, the opposite actually occurs. Figure 5, below, shows the average bytes per packet as a function of time. Here we see that this number stays relatively stable between 70 and 75 bytes per packet whereas we expect to see a rise into the thousands of bytes per packet in the case of a DNS Amplification attack.

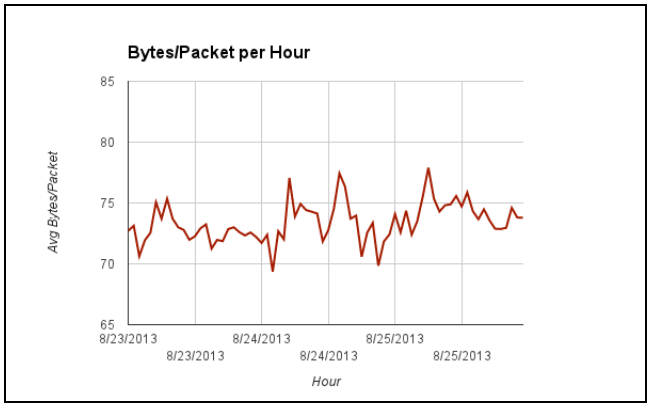

Figure 5 - Small variations in Bytes/Packet are nowhere near the expected increases of a DNS Amplification attack.

2.4 Is this an NXDOMAIN attack?

2.4.1 NXDOMAIN Attacks

The term "NXDOMAIN" is DNS short hand for name errors in a DNS response[1]. This type of response exists solely to communicate the absence of a record in a particular name server's database. The fact that DNS servers must communicate this information in order for DNS to work properly leaves a vulnerable attack vector open to any malicious actor willing to exploit it.

An NXDOMAIN attack consists of a single host or many hosts in a botnet, each querying the target server for non-existent URLs for which the target DNS server should be the authoritative name server. In addition to performing a guaranteed worst-case search of both its cache and the zone database, once the server generates the response, it will replace a valid cache entry with the newly created NXDOMAIN entry. This dual threat, flushing the cache while consuming valuable CPU and disk resources generally overload DNS servers fairly quickly and lead to this threat being called "the most advanced form of attack against DNS services."[15]

2.4.2 Limits of UMD NetFlow

Unfortunately, in searching for evidence of an NXDOMAIN attack we've run into the limit of how much knowledge we can gain through the use of NetFlow data alone. The limit exists because we only have NetFlow records for traffic which crosses the border router and in order to prove or disprove anything about the existence of NXDOMAIN attacks, we'll need information on the traffic between the UMD DNS and its clients. Since the UMD DNS only services requests it receives from internal hosts, we never see the DNS requests in the NetFlow records. Using NetFlow to investigate this event is also limited in that we can't see what domains were requested (or by whom).

In an effort to aid our study of this event in light of the limits of the campus NetFlow, Symantec Research Labs was kind enough to provide us with an additional dataset. The data provided consists of DNS server logs of DNS servers run by the anti-virus company for its clients. The logs span two days, August 24-25 and are filtered to only include NXDOMAIN records. Each record consists of a four-tuple including a timestamp, server ID, client IP address and the requested domain (which resulted in a NXDOMAIN response). With the new dataset in hand, we were able to begin the process of determining whether the attack may have been an NXDOMAIN attack.

2.4.3 Finding the Attack with DNS Logs

Similar to our approach in initially finding the attack with NetFlow, we first seek to determine if the attack is visible at all with only the DNS logs. Since the servers producing the logs only service queries from customers of Symantec anti-virus products, there is at least some reason to believe that these DNS clients might be protected from whatever malware that controls the botnet which perpetrated the attack. As such, we expect to find *less* evidence of the global event in these DNS logs than we would in similar logs from servers open requests from any public hosts.

We begin our analysis of the DNS logs in much the same way we began our analysis of the NetFlow records. Using assumptions similar to A1 and A2, we propose a hypothesis trivially similar to H1 albeit adjusted to find events in the Symantec DNS data.

H4: *Since we should see evidence of global events in DNS logs from distributed servers and the attack on .cn was likely of global scale, we should see evidence of this attack in the DNS logs.*

As we mentioned that these DNS service only select client requests, we add the caveat assumption (**A5**): *there exist some clients of the Symantec DNS which are infected by whatever*

malware is responsible for controlling the botnet which perpetrated the attack on the Chinese ccTLD.

Below, Figure 6 shows the number of requests for domain names ending in .cn as well as how popular that TLD is compared to all other TLDs. This is useful because as one might expect, .cn is not the most popular TLD by any measure and the logs are dominated by requests which either mangle the TLD in their request or else ask for non-existent .com URLs. Although we don't have enough data before or after this event to determine whether the change we observe is part of a long term pattern or evidence of a continued campaign against the ccTLD, there does appear to be a change in behavior of some sort on exactly midnight August 25th. Beginning on August 25th, the average number of requests per hour roughly triples and the popularity of .cn increases from around 50 to become the 20th most popular ccTLD in the data set.

Since the timeline does not quite matchup with our NetFlow data, we'll introduce a new assumption to explain why the behavioral change occurs at that time. Without the following assumption, it becomes more difficult to explain why the behavioral change occurred when it did.

A6: *The timestamps found in the Symantec DNS logs correspond to local time at the DNS server itself.*

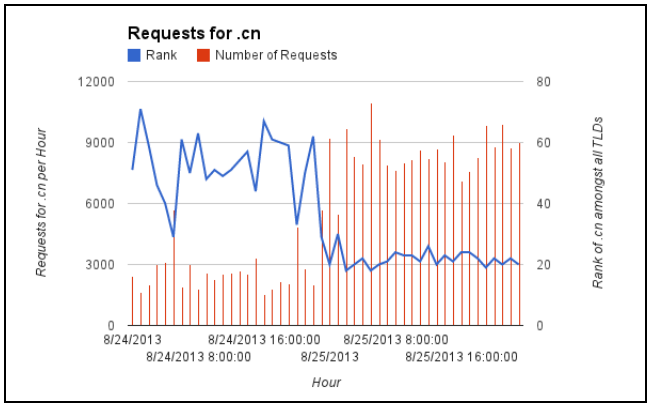

Figure 6 - As requests for records ending in .cn increase, the Chinese ccTLD increases in popularity amongst all TLDs.

2.4.4 Is it Consistent with an NXDOMAIN attack?

Our previous hypothesis explored whether there was an observable change in the data during the time period coinciding with the attack on the .cn name servers. Here, we'll assume the validity of our previous hypothesis with the assumption (**A7**) *that the DNS logs do show evidence of the globally malicious event.* We can combine this assumption with facts we already know about NXDOMAIN attacks and DNS to yield a new hypothesis. Recall that in order for an NXDOMAIN attack to work, the attacker must continually request *new* non-existent domains of the target system or else the local DNS will cache its previous response and fail to pass the request to the target.

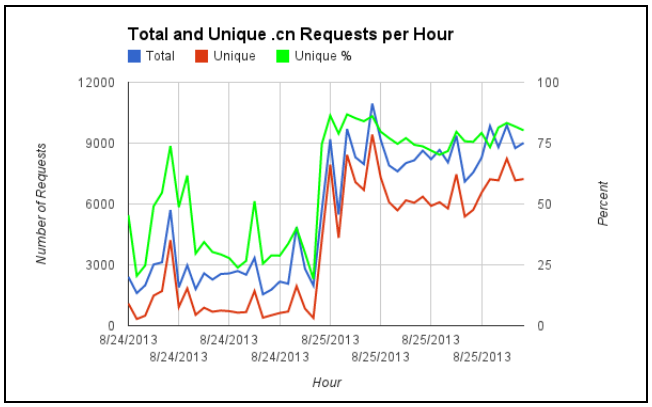

Figure 7 - The rise in traffic is accounted for almost entirely by a rise in *unique* domain requests.

H5: *Since the DNS logs contain attack related requests and NXDOMAIN attacks require issuing requests for unique domains, the DNS logs should contain in increase in requests for unique domains which corresponds to the attack related traffic.*

Figure 7 above, illustrates that the increase in requests is due almost entirely to an increase in unique requests. Whereas pre-attack DNS requests tended to be less than half unique, requests made during the attack period are more than 75% unique. If the rise in traffic was not from an NXDOMAIN, we would expect to see a proportionate rise in both unique and non-unique domain requests. This indicates that the behavior change is consistent with an NXDOMAIN attack and validates our hypothesis H5.

2.4.5 Can DNS Logs Reveal More About the Malware?

We've shown that the behavior captured in the DNS logs is consistent with expected behavior of an NXDOMAIN attack. Are there other characteristics of the attack which might be revealed through the DNS logs? One assumption we can make is that in order to quickly produce a high volume of unique requests, whatever malware is responsible must have an automated URL generator.

A7: *The malware responsible for the attack has an automated URL generator.*

With this assumption in mind, we asked ourselves, "If I were writing a URL generator, how would I do it?" The easiest way to do it would be to pick a string of predetermined length and then fill it with random characters, appending ".cn" to the end of whatever string was generated. While each string generated in such a manner is likely to be unique with a given high probability, there is one characteristic which remains constant across all strings: the length. This observation leads us to our final experimental hypothesis.

H6: *Since the unique URLs are generated algorithmically, they are likely to share a common length which will be reflected in the DNS logs.*

There are many algorithms for generating URLs which would cause this hypothesis to fail. Notably, one could perform a dictionary search and append ".cn" to every word (or combination of two words) in the dictionary. This would yield non-uniform string lengths across generated URLs but we might subject URL lengths to frequency analysis coinciding with the frequency of word lengths in the dictionary.

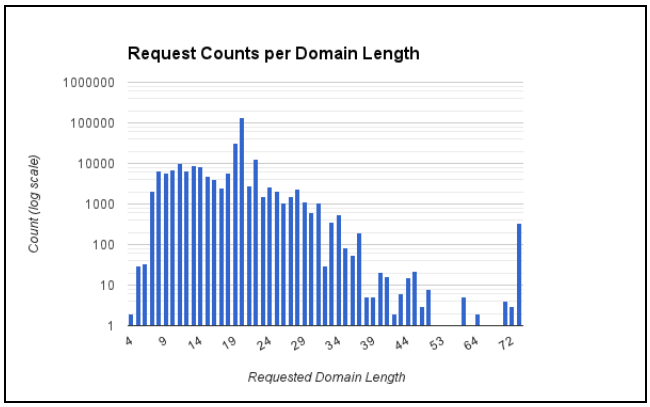

Figure 8 - The request count for domains of length 20 is almost five times as many as the next most requested length (19).

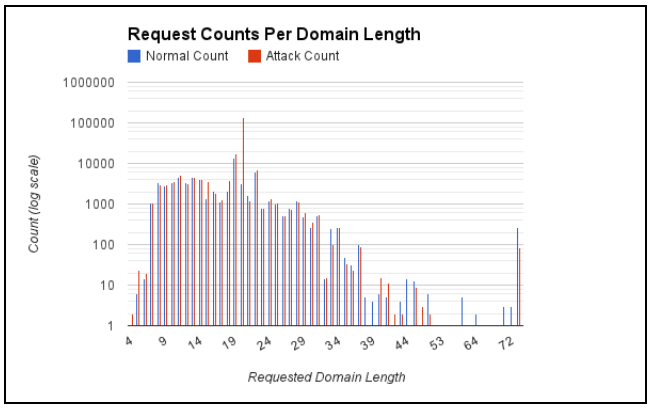

Figure 9 - Most domain lengths are requested at similar rates before and after the event. Only length 20 changes drastically.

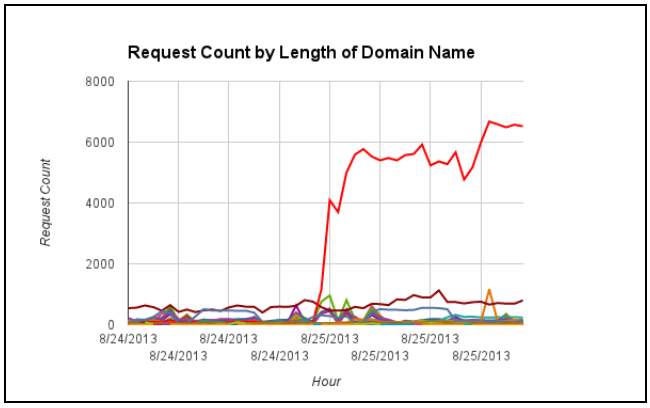

Figure 10 - The red line shows the number of requests per hour for domains of length 20, dwarfing requests for other domain lengths.

2.4.6 Magic Length = 20

On the opposite column are figures 8, 9 and 10. They detail our initial discovery about the auto-generated domains. Specifically, we validate our hypothesis that URLs generated algorithmically are likely to have similar lengths.

What is interesting, however, is that the algorithm generating these URLs is not generating random strings. After manually inspecting many DNS records, it appears that the algorithm used to find unique URLs was to dig through a database of valid .com domains, choose URLs whose total length was 17 and then append ".cn" to the original URL. We also verified manually that these URLs do point to existing domains under the .com TLD but do not exist under the .cn TLD. A sample of the generated domains can be seen in the table below.

Table 1 - Selected generated URLs

www.putmanfps.com.cn
www.papagoinn.com.cn
www.shrilanka.com.cn
www.hifivideo.com.cn
www.adobegold.com.cn
www.rhinobldg.com.cn
www.onthepark.com.cn
www.bellashoe.com.cn

3. CONCLUSIONS AND FUTURE WORK

Our case study into the attack on the Chinese ccTLD was revealing. Not only were we able to find evidence of the attack from NetFlow records on the UMD network, but we were also able to utilize an additional dataset to uncover more about the nature of the attack. We should be careful to note that while these findings are interesting, they still seem to leave us with more questions than answers. In particular, why does the traffic spike drastically on the UMD network but on the DNS logs it appears to increase once and remain at the higher rate? How long does this behavior continue for? Finally, although we have a good idea how the attack works, we have no ground truth from the operators of the ccTLD themselves. While we don't foresee the CNNIC becoming a collaborator on future research, it would be nice to know the reality of the attack from the victim's perspective. In a similar vein, performing this analysis on UMD DNS logs would validate our conclusion that UMD hosts participated in the attack. However, without the actual DNS logs (which have been erased by now) we'll never know for sure.

4. RELATED WORK

In dealing with both DNS and flow records, several works inform and motivate our own. First, flow data analysis is quickly becoming a popular method for detecting botnets and other forms of malicious activity. Bilge et al. [4] cover detecting botnets through NetFlow analysis and the extraction of six novel features for clustering. Iliofotou et al. [18] and Nagaraja et al. [22] both utilize flow records to build graphs of network traffic and use graph based algorithms to find malicious activity. While the previous two works utilize flows captured across ISPs, Coskun et al. [12] also use flow-derived graphs to detect botnet activity but only at the edge router of enterprise networks.

Regarding DNS, Antonakakis et al. has demonstrated that it is possible to detect malicious domains that are automatically generated, since the typical automated alphanumeric string is not well-formed [3] and this may explain motivations for the attacker to use a dictionary based approach as opposed to random string generation. Additionally, Bilge et al. [5] utilize passive DNS analysis to detect malicious domains.

5. ACKNOWLEDGMENTS

I would like to extend my thanks The University of Maryland Division of Information Technology for allowing us to use the campus network and data for my research. I'd also like to thank Symantec Research Labs for offering their support of my research and providing me with their DNS logs.

6. REFERENCES

[1] Andrews, M. 1998. *RFC 2308: Negative Caching of DNS Queries (DNS NCACHE)*.

[2] Announcement: 2013. *http://www.cnnic.net.cn/gywm/xwzx/xwzxtzgg/201308/t20130825_41322.htm*.

[3] Antonakakis, M. et al. 2011. Detecting malware domains at the upper DNS hierarchy. *Proceedings of the 20th USENIX Security Symposium, USENIX Security* (2011), 27.

[4] Bilge, L. et al. 2012. DISCLOSURE : Detecting Botnet Command and Control Servers Through Large-Scale NetFlow Analysis. *Proceedings of the 28th Annual Computer Security Applications Conference* (New York, New York, USA, Dec. 2012), 129–138.

[5] Bilge, L. et al. 2011. EXPOSURE : Finding Malicious Domains Using Passive DNS Analysis. *NDSS*. (2011), 1–17.

[6] China suffers major DDoS attack on .cn domain | ITworld: 2013. *http://www.itworld.com/internet/370449/china-suffers-major-ddos-attack-cn-domain*.

[7] China's Internet hit by DDoS attack; sites down for hours | Security & Privacy - CNET News: 2013. *http://news.cnet.com/8301-1009_3-57600083-83/chinas-internet-hit-by-ddos-attack-sites-down-for-hours/*.

[8] Chinese internet server jammed in largest ever mainland hacker attack: 2013. *http://www.scmp.com/news/china/article/1299654/mainland-internet-server-jammed-hacker-attack*.

[9] Claise, B. 2004. *Cisco systems NetFlow services export version 9*.

[10] Claise, B. 2008. *Specification of the IP flow information export (IPFIX) protocol for the exchange of IP traffic flow information*.

[11] CNNIC hit by "largest ever" denial of service attack: 2013. *http://domainincite.com/14300-cnnic-hit-by-largest-ever-denial-of-service-attack*.

[12] Coskun, B. and Dietrich, S. 2010. Friends of An Enemy : Identifying Local Members of Peer-to-Peer Botnets Using Mutual Contacts Categories and Subject Descriptors. *Proceedings of the 26th Annual Computer Security Applications Conference* (2010), 131–140.

[13] Frank Hemingway et al. 2013. A Flow-Based, Blacklist Approach for Associating Web Browser Redirection with Malicious Activity. *LASER Workshop* (Arlington, VA, 2013).

[14] Franklin, J. et al. 2007. An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants. *Proceedings of the 14th ACM conference on Computer and communications security*. (2007), 375–388.

[15] Holmes, D. 2012. The DDoS Threat Spectrum. F5 Networks, Inc.

[16] ICANN Security and Stability Advisory Committee 2006. *SSAC Advisory SAC008, DNS Distributed Denial of Service (DDoS) Attacks*.

[17] ICANN selects new gTLD backup providers: 2013. *http://domainincite.com/12539-icann-selects-new-gtld-backup-providers*.

[18] Iliofotou, M. et al. 2007. Network Monitoring using Traffic Dispersion Graphs (TDGs). *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement* (2007), 315–320.

[19] Kambourakis, G. et al. 2008. Detecting DNS Amplification Attacks. *Critical Information Infrastructures Security*. Springer Berlin Heidelberg. 185–196.

[20] Mockapetris, P.V. 1987. *RFC 1034: Domain names - concepts and facilities*.

[21] Mockapetris, P.V. 1987. *RFC 1035: Domain names - implementation and specification*.

[22] Nagaraja, S. et al. 2010. BotGrep : Finding P2P Bots with Structured Graph Analysis. *Proceedings of the 19th USENIX Conference on Security* (2010).