Detection of Malicious Payload Distribution Channels in DNS

A. Mert Kara †‡, Hamad Binsalleeh †‡, Mohammad Mannan ‡, Amr Youssef ‡, and Mourad Debbabi †‡
† National Cyber Forensics and Training Alliance Canada
‡ Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada
{ab_kara, h_binsal, mmannan, youssef, debbabi}@ciise.concordia.ca

***Abstract*—Botmasters are known to use different protocols to hide their activities. Throughout the past few years, several protocols have been abused, and recently Domain Name System (DNS) also became a target of such malicious activities. In this paper, we study the use of DNS as a malicious payload distribution channel. We present a system to analyze the resource record activities of domain names and build DNS zone profiles to detect payload distribution channels. Our work is based on an extensive analysis of malware datasets for one year, and a near real-time feed of passive DNS traffic. The experimental results reveal a few previously unreported long-running hidden domains used by the Morto worm for distributing malicious payloads. Our experiments on passive DNS traffic indicate that our system can detect these channels regardless of the payload format.**

I. Introduction

A common approach to bypass network defense borders is tunneling the communication through existing protocols. Such tunneling can effectively defeat traditional firewalls and intrusion detection systems (IDSs). Botmasters often prefer tunneling to keep their communications under the radar. In the early days of botnets, botmasters mostly used Internet Relay Chat (IRC) channels to operate and control their activities [12]. The advancement of newer protocols such as HTTP and P2P largely outdated the use of IRC channels [4], [10], [19]. As a natural extension to exploiting common protocols for tunneling, DNS comes into play due to its wide availability. DNS is a query and response protocol, which responds to each query with corresponding pre-defined resource record. The simple but robust architecture of DNS attracts botnets to abuse the system for different malicious activities [3], [6], [7], [8].

Botmasters take advantage of DNS tunneling to conduct malicious activities such as command and control (C&C) or payload distribution. In payload distribution channels, for instance, botmasters use DNS query and response packets to carry out malicious instructions and payload updates to individual bots. Recently, few malware families such as Morto [16], Katusha [19], and Feederbot [8], have been identified as using the DNS protocol to hide their communications.

Compared to other protocols, the inherent nature of DNS renders the protocol quite inefficient as a payload distribution channel [24]. On the other hand, DNS is widely available, which explains the recent exploitation of DNS as an attack channel despite the narrow transmission bandwidth. However, DNS abuse by malware has not been comprehensively studied so far, as compared to P2P botnets [10], and previous work on DNS abuse mainly focused on specific malware families (e.g., [8]).

In this paper, we propose a detection mechanism for DNS payload distribution channels by leveraging some features of DNS to distinguish between malicious and non-malicious domains. We use this mechanism to analyze a significant amount of DNS traffic in order to understand the extent of DNS abuses in the wild. We detect a few previously unknown long-lasting malware domains and different types of payload distribution channels. This result is significant, considering the fact that DNS payload distribution channels are still exploited relatively rarely. Our proposed technique, which is based on access counts of resource records, shows promising results regardless of the syntax formats of payload distribution channels.

Our contributions of this paper can be summarized as follows:

- Comprehensive analysis of malicious payload distribution channels using a 1-year malware dataset covering Jan.-Dec. 2012.
- Detection of payload distribution channels using passive DNS traffic. We propose a method to detect payload distribution channels based on access counts of resource records.
- Evaluation of the proposed method with near real-time passive DNS traffic for a 30-day period.

The rest of the paper is organized as follows. Section II provides background on payload distribution through DNS. Related work is reviewed in Section III. Our system is described in Section IV. Section V explains our datasets and Section VI demonstrates the effectiveness of our proposal via an experiment on near real-time traffic. We discuss the limitations of our work in Section VII, and Section VIII provides the concluding remarks.

II. Background

In this section, we introduce payload distribution channels in DNS. Then, we discuss the use of these channels both for legitimate and malicious purposes. Finally, passive DNS is briefly explained.

A. Payload Distribution via DNS Hierarchy

Recently, DNS has become a target to distribute malicious payloads for two main reasons. First, DNS traffic is often allowed to pass without inspection in corporate networks as it is considered to be a core element of Internet activities. Second, the DNS protocol includes some fields that are defined to be more flexible for future usages of the protocol. Malicious payloads can be stored in different resource records (e.g., TXT, or CNAME). However, the TXT resource records are a more

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

viable option compared to CNAME resource records, because CNAME is less flexible as it requires the domain name syntax. The payload data can be cached in DNS resolvers, and they can be accessed even if command and control servers are down. RFC 1464 paved the way for payload distribution by opening the possibility of storing arbitrary information within DNS messages [23]. In particular, it is recommended in RFC 1464 to store key-value pairs to pass only some operational data between servers. The feasibility of using DNS resource records to distribute payload has been proven by the DNS tunneling technique, which shows that DNS can be used for transmitting any type of information after simple encoding operations.

B. Existing Uses of Payload Distribution

The DNS protocol is not intended to be used as a payload distribution mechanism. However, there are very limited number of legitimate uses that are introduced by some organizations to channel parts of their operational data through DNS.

Legitimate Uses: In 2007, Trend-Micro Inc. proposed a method to distribute malicious code signature updates through the DNS protocol [14]. The intention of this technique is to feed anti-virus client software with signature updates through DNS, as an alternative update mechanism. The signature updates are divided into several chunks, which can be identified by an index number. These pieces are encoded with Base64, and assigned to the zone file as TXT resource records of a specific domain name. When the client needs an update of a malicious code signature, it sends a query with an index number of the signature in the sub-domain label. Then the server responds with the corresponding anti-virus signature in TXT records. In general, each signature update can span over many TXT records, which makes the client generate many queries to retrieve the whole update. Finally, the client combines all TXT records to form the actual update.

In 2009, Devicescape Software Inc. introduced a system for public hotspot authentication systems for mobile devices [9]. In their model, there are public WiFi hotspots, which are placed across many places such as coffee shops and restaurants. The authentication system for these hotspots is managed through a central server. The DNS protocol is used as a channel to transfer authentication parameters between mobile devices and a credential server. The client prepares a DNS query, which consists of six sub-domain labels to carry different parameters, e.g., the MAC address of the client's machine. When the name server receives the query, it forwards the parameters embedded in these labels to the back-end credential server. Based on the parameters, the credential server prepares the corresponding credentials to be transported back to the client. Finally, the client verifies the response and then submits it to the authentication server in the local hotspot network.

Malicious Uses: The crucial component of any malicious network is the control and communication method within the network. In 2011, Dietrich et al. [8] reverse engineered the Feederbot botnet that used DNS as a command and control channel. Another example of the abuse of the DNS protocol is the Morto worm, which uses DNS TXT records to transmit an encrypted URL to the real attack payload [16].

C. Passive DNS

Passive DNS is a technique to replicate DNS activities in order to investigate the DNS traffic in near real-time. The inconsistency between A and PTR records [25], which is caused by dynamic IP addresses, requires a technique to track changes in resource records. Therefore, passive DNS is introduced to establish a real-time replication mechanism [25]. It is designed to be deployed on a local DNS resolver to observe the DNS traffic.

III. Related Work

The use of DNS as a payload distribution channel is relatively new and research activities on this topic are somewhat limited. Existing studies are scattered, they can be roughly grouped under three categories: detection of malicious channels in DNS, feasibility of using DNS in malicious activities, detection of DNS tunnels, and using DNS traffic for detecting other malicious activities.

Feasibility of using DNS for Malicious Activities: Xu et al. [27] introduce a resilient mechanism for bots to create covert channels through DNS for command and control communications. They design a stealthy C&C that supports two different modes. The *codeword mode* creates a uni-directional communication channel that pulls the attack payload. The *tunneled mode* creates a bi-directional communication channel between bots and the command and control server. In fact, their proposed methods are already used by some malware families such as Feederbot and Morto [8], [16]. Similarly, Raman et al. [21] propose a network penetration technique that uses DNS tunneling to infiltrate the attack payload. Their technique is based on establishing a tunnel by exploiting a software on the host machine. The exploit code sends DNS queries to receive the real attack payload. There are several studies on building a covert channel using DNS query and response packets [5], [15], [18], [24]. In these studies, the authors discuss the possibility of sending and receiving data through DNS query response packets, and provide performance results for existing DNS tunneling tools.

Malicious Channels in DNS Protocol: Dietrich et al. [8] are the first to discuss the existence of botnets that tunnel command and control channels through DNS. They discovered a malware family (Feederbot) that exfiltrates data within DNS query sub-domain labels, and infiltrates attack payloads in DNS response packets. Their detection method introduces the extraction of several features from response data. While their work shows promising results, it is limited to the detection of DNS tunnels with an extensive traffic for command and control channels. Other malware families are using more resilient methods for receiving the attack payloads through DNS by combining DNS with other protocols [16]. Also, their work focuses on the assumption that there is a certain volume of traffic while some families use DNS to receive a very limited amount of payload such as the Morto family [16]. Also, the solution relies on the fact that the channel data is always encoded. However, malware might not receive Base32 or Base64 encoded payload, rather clear text.

Detection of DNS Tunnels: There are some proposed methods for detecting DNS tunnels within a network using the n-gram analysis [11], [20]. The proposed methods present promising

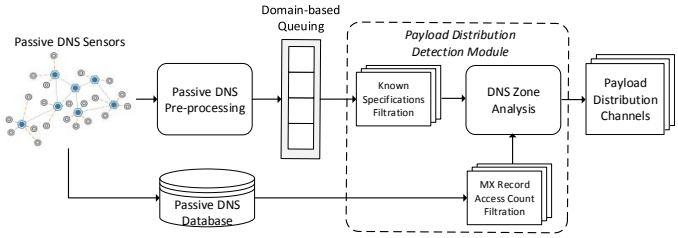

Fig. 1: System overview.

results in terms of detecting DNS tunnels. However, malicious payload distribution channels often do not have extensive upstream data; thus they do not show this characteristic feature of DNS tunnels. Therefore, any string based analysis on queries might not reveal enough differences between regular and malicious queries to detect these channels.

IV. System Description

Our system monitors DNS queries and responses in passive DNS and detects payload distribution channels established within DNS messages. As shown in Figure 1, the system consists of one main module, which we label as the payload distribution detection module. Initially, the system pre-processes the passive DNS traffic by extracting DNS messages that have TXT resource record activities, and then divides the captured DNS traffic stream into epochs (e.g., $epoch$ = 1 day). For each epoch, it aggregates the DNS query and response messages of a given domain name to be added to the domain-based queue. In parallel, the passive DNS traffic is also stored in a passive DNS database which collects historical data about DNS messages for off-line analysis. After the pre-processing phase, the collected messages are fed to the payload distribution detection module, which identifies the payload distribution channels. In the following section, we provide details on the detection module.

Each DNS message has a domain name d that consists of a set of labels. The rightmost label is called the *top-level* domain, the two rightmost labels are called *second-level* domain, and the rest of the labels are referred in the same manner. The services provided by these labels are represented in a *zone* file and stored in the corresponding authoritative name server. Name servers are capable of handling any DNS query and returning the corresponding responses, which are taken from the zone file of each domain name. As name servers are the key players in DNS, malicious networks must have access to a name server for managing the payload distribution. When a name server is appointed to be the authoritative name server for a malicious domain name, botmasters prepare the zone file of the domain to store all attack payloads to be delivered via DNS. Since DNS zone files reflect the provided services of domain names, we decided to observe DNS zone files to analyze domain names behavior.

DNS Zone Analysis: First, we analyze the behavior of domain names by observing DNS zone activities in the passive DNS traffic. Within the zone file of each domain name, there are different types of resource records. Each resource record indicates specific services or operations associated with the domain name. An important feature of the passive DNS database is the aggregation of how many times each record has been requested, called *access count*. Our intuition is that, domain names which are solely used for payload distribution, show different behavior compared to regular domains. Regular domains receive queries for different resource records. On the other hand, malicious domain names, which are mostly used for payload distribution through DNS, are only accessed to receive attack payloads. Therefore, botmasters only focus on using specific resource records that are known to be used in payload distribution channels such as TXT records. Moreover, these domains do not heavily use the resource records that are normally used by regular domain names, such as A, AAAA, and MX resource records. Thus, by observing the resource records and their access counts, we can profile the DNS zone activities of a domain name.

Extraction of DNS Zones: In payload distribution channels through DNS, name servers are considered as the payload distributors. Since domain names can have multiple zones, we must recognize the responsible zones that are associated with payload distribution.

Since a domain might have multiple labels that point to different zones, the system traverses the labels from second-level to the left-most label. For each label, the NS resource record is requested to see whether that label is a zone or not. If a sub-domain label has an NS record, it is a zone under that second-level domain. In the next step, the system profiles DNS zone activities of this zone.

Profiling of DNS Zones: Understanding whether a zone is used for payload distribution purposes can be achieved by analyzing its resource record activities. These activities can be calculated as a function of access counts. By using the passive DNS database, we extract all accessed resource records and their access counts. The passive DNS is built in a way such that it counts the accesses to each resource record for a certain period.

Let $R = \{A, NS, CNAME, \ldots, TXT\}$ be the set of all resource record types that can be defined in a DNS zone file, and $P = \{p \mid p \in R \setminus \{NS, CNAME\}\}$ is the set of all the resource record types that are commonly used by payload distribution channels. Since the TXT resource record is known to be the most suitable for payload distribution, we define a set $T = \{r_{TXT} \mid r_{TXT}$ is a TXT record$\}$ that holds any TXT record in a given zone.

For every resource record type from sets P and T, the corresponding ac_T and ac_P are retrieved from the passive DNS database. Then, these access counts are aggregated to determine μ as follows:

$$\mu = \frac{\sum ac_T}{\sum ac_P} \tag{1}$$

From Equation 1, μ reflects the relation between the access ratios of T and P records and thus acts as an indicator of payload distribution activities. Payload distribution channels receive a significant access counts from T, which results in larger μ values. However, non-payload distribution channels receive more access counts on the resource records from P, and hence result in smaller μ values.

Filtration Steps: There are some legitimate cases that can behave as payload distribution channels. In fact, there are specifications that are using TXT records to apply some

Passive DNS	Period	30 days
	DNS messages	≈ 20 Billion
	TXT records	≈ 40 Million
Passive DNS Database	Period	3-year
Malware database	Period	1-year
	No. of samples	≈ 15 Million
	No. of samples with TXT activities	≈ 18 Thousand

TABLE I: Dataset statistics.

security measure for mail servers such as SPF [26], DKIM [2], IKE [22], and DNSBL [13]. Since these specifications are designed for mail servers, a DNS zone file should reflect the existence of MX resource records. These legitimate services can be recognized using two different filtration steps: specifications recognition (i.e., SPF, DKIM, IKE, or DNSBL), and MX resource record activity. The first filtration process takes each domain, and selects the most accessed TXT resource record using the passive DNS database. Then, we apply a regular expression in the TXT record based on the defined syntax of specifications [2], [26], [22], [13] to determine any possible specification string (e.g., SPF). When a TXT record data matches any defined specifications, we consider the domain name as non-payload distribution channel. In the second filtration step, we investigate the activities of MX resource records. When a domain name is associated with MX resource record activities, it is considered as non-payload distribution channel.

V. Datasets

We utilize three datasets to evaluate our system: a near real-time passive DNS traffic, a passive DNS database, and a malware database.

Passive DNS Traffic: We evaluate the system using a one-month passive DNS dataset, which spans between March 19, 2013 and April 19, 2013, provided by Farsight Security Inc. [1] We only process the packets with TXT responses. According to the system logs, the total number of packets processed by our system is around 40 million packets with an average of about 1.3 million packets daily (Table I).

Passive DNS Database: Our system also builds a passive DNS database which stores all the data coming from the passive DNS traffic. This database contains the last three-year period of the passive DNS traffic.

Malware Database: We observe malware samples provided by a major security vendor over one-year period. We receive the malware feed on a daily basis and then analyze each sample in a sandbox to generate dynamic behavioral analysis reports. In our analysis, we just consider malware samples that conduct activities using the TXT resource record. Table I shows some of the statistics about the malware feed recorded between January 2012 and December 2012.

VI. Experimental Results

In this section, we report the results of our experiments that we perform to test the effectiveness of our system for detecting payload distribution channels in passive DNS dataset. We begin by demonstrating the results of the DNS zone analysis module using the passive DNS dataset. Then, we elaborate on the long-running hidden domains used by the Morto worm to distribute attack payloads. Finally, we conclude the section by

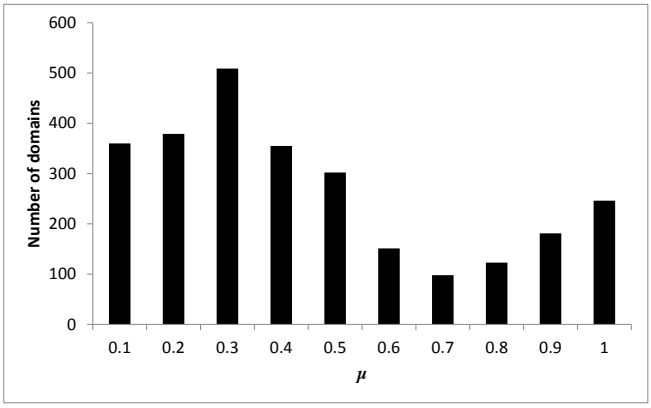

Fig. 2: Distribution of rating values of the detected 2707 domains.

Number of observed domains	2707
Domains with MX records	2506
Domains follow known Specifications	2613
Domains remained after applying both filtrations	37

TABLE II: Statistics of detected domains within 30-day.

showing that, on contrary to the common wisdom (e.g., [16]), some of the attack payloads are in clear-text without any encoding or encryption. This indicates that our system can detect these channels regardless of the syntax of the distributed data.

During our experiments, we processed domain names that are accessed for TXT records in a time-based window; we set the window to be one day. When the window expires, the packets are fed to the DNS zone analysis module, to build the DNS zone profile for each zone for detecting payload distribution channels.

DNS Zone Analysis: When the query and response messages of domains are captured, they are inspected by the DNS zone analysis module. The access counts of each resource record are gathered from the passive DNS database. Equation 1 determines the μ values of each domain based on the access counts. During our experiments on the passive DNS traffic, we have captured 2707 domains that have TXT resource record activities. Figure 2 shows the distribution of domain names with different μ values. According to Equation 1, the bigger μ value is the more a domain name is involved in payload distribution.

In Table II, we show the number of detected domain names during the 30-day period and the effect of each filtration mechanism. The number of detected domains is 2707 before any filtration is applied. However, some of these domains might be mainly accessed to receive specifications related data in TXT records. Applying both filtration mechanism reduces the number of domains to 37.

To validate the effectiveness of the filtration process, we observe our malware dataset and passive DNS database to investigate the difference between payload distribution channels and regular domains. As regular domains, we use the top 500 domains from Alexa top sites. By using our one-year malware dataset, we extract malware domains, which are

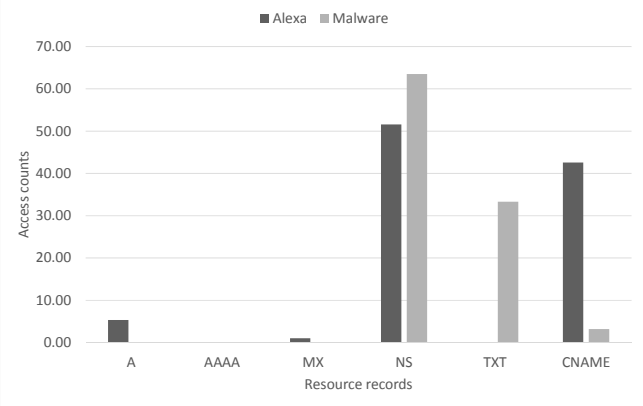

Fig. 3: Alexa and malware domains DNS record access counts.

used for payload distribution. We retrieve the access counts for all resource records of each domain from regular and malware domains. These access counts are a good measure to understand the individual resource record activities of any given domain. In Figure 3, the distribution of the access counts for these resource records is given. Domains from Alexa receive DNS queries for different resource records. The reason for this can be attributed to the fact that these domains utilize DNS for enabling access to different services. On the contrary, malware domains receive an extensive number of DNS queries for TXT records. These records are used to distribute the payload as it is the most suitable resource record type within the protocol. We also investigate the access to the CNAME records in malware domains. They are used to redirect queries between malicious domains as botmasters maintain a network of malicious payload distribution channels. On the other side, malicious domains are not associated with any MX resource records, which support the second filtration process.

The Resilient Morto Domains: Morto is a malware family that targets the Remote Desktop Protocol (RDP) to gain access to host machines. It is one of the malware families that use DNS as a payload distribution channel [16]. By utilizing the passive DNS database, we detected domains that are used by the Morto family for more than 18 months with an average to TXT records over four million times. Past work [16] has revealed that Morto receives a Base64 encoded or encrypted URL, which points to the second payload. We noticed that Morto domains also distribute IP addresses in clear-text inside TXT records. A reverse lookup to one of these IPs in passive DNS database reveals that it is shared with other malicious domains. As mentioned in Section VI, the malware authors also link different domains to each other through CNAME records to maintain a malicious network.

Source	No. of domains	Avg. μ value
Devicescape [9]	12	0.998
Tunneling [17]	1	0.991
Morto [16]	3	0.994

TABLE III: Detected payload distribution domains.

Detection Regardless of Syntax: As our method discovers the Morto domains, it also detects the legitimate payload distribution channels as discussed in Section II-B. It indicates that regardless of the syntax of the payload distribution channel, the DNS zone activity metric is a strong feature to detect domains, which are used for these channels (Table III). If botmasters start using a syntax similar to the legitimate services to blend in their traffic, they might not be detected by network monitors. However, our system may still detect it since the system monitors the DNS zone activities of payload distribution channels rather than investigating their message syntax.

DNS Tunneling Detection: Our experiments reveal some DNS tunneling activities from a single domain (a DNS Tunneling app for Android [17]). As our system is configured to monitor TXT records, it successfully detects any DNS tunneling activities on TXT records. If the tunnel is established by using another resource record type, we expect that our system would still detect as the detection is not based on the content of the resource record, but the access counts of resource records.

In Table III, we summarize our results with use cases and average μ values. After applying both filtrations, we are left with a few domains per day. Therefore we could investigate their traffic manually. The manual investigation resulted in 16 domain names that are used as payload distribution channels. The remaining domains are used for transmitting some domain specific data in TXT records.

VII. Discussion and Limitations

During our observation of the malware dataset, we find domain names that are used for payload distribution channels. These malware samples use different methods to retrieve the malicious content as discussed in Section II-B. One of the interesting ways is that they use indexed queries to receive attack payload in multiple response packets. Due to the size restriction on TXT resource records, the payload is chunked into parts and each part is placed in another TXT record. Bots start querying this series of packets in a sequential manner until the last packet is received. Some of these payloads are chunked up to thousand of packets. Surprisingly, this method is very similar to the patent from Trend Micro [14]. However, our results showed that this method is not seen in our passive DNS dataset. There are two possible interpretations of not observing this behavior in our recent dataset. First, botmasters realize the significant exposure of using this mechanism, which generates a large number of messages, and then decide to stop using it. Second, these domain names are directly resolved by their own name servers or other open resolvers, which are not captured by passive DNS sensors.

The closest work to our proposed solution is the detection of DNS traffic of a specific malware family by applying features that are based on previously seen malicious traffic syntax [8]. Our solution uses another approach by investigating the DNS zone activities of malicious domains. Even if malware changes the message syntax, the use of DNS remains the same. Our method detects the malicious traffic regardless of the syntax and malware family.

The proposed solution can be used in real-time scenarios where there is an access to the DNS zone activities of domains. For example, it can be used by a domain name registrar to detect registered domains that are used for payload distribution. In this case, the system can be deployed on the authoritative

name servers of the registrar so it can observe the zone activities of domains.

Finally, limitations of our current design include the following. First, there are domain names that use TXT resource record for legitimate services along with other activities. Malware can play the same role by operating different services on the same domain name. When a domain is used for different malicious activities (i.e. spam, phishing) as well as for payload distribution, then it will be accessed for different resource records, e.g., A record for phishing scam websites. Since our system uses the fact that name servers of payload distribution channels mostly receive requests for TXT records, it might consider this domain as non-payload distribution domain. Second, the passive DNS replication [25] is a unique way to collect the global DNS traffic by sensors. However, it has a shortcoming that might affect our results. Malware might not use the caching resolver of a network, and alternatively send queries directly to an open resolver. In this case, the traffic would not pass through the sensors that collect DNS traffic, and would not be captured. However, this is likely to remain a limitation in all solutions that are based on passive DNS datasets.

VIII. Conclusions

In this paper, we shed some light on the abuse of the DNS protocol by malware for distributing attack payloads. We designed a system to detect the payload distribution channels within passive DNS traffic. Our system observes the DNS zone activities of a channel by gathering access counts of each resource record type, and determines payload distribution channels. Our experiments on near real-time passive DNS traffic show that our system can detect several resilient malicious payload distribution channels, which were active for more than 18 months. We found that most malware instances with DNS-based payload distribution are using a resilient pattern to retrieve their attack payloads, apparently to blend within regular network traffic. Moreover, our system is able to detect payload distribution channels regardless of their syntax format.

Acknowledgments

The authors would like to thank Paul Vixie for his valuable comments.

References

[1] Security Information Exchange (SIE), Farsight Security Inc. https://www.farsightsecurity.com.

[2] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, and M. Thomas. Domainkeys identified mail (DKIM) signatures. RFC 4871, proposed standard, May 2007.

[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 *USENIX Security Symposium*, Bellevue, WA, USA, Aug. 2012.

[4] H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi, and L. Wang. On the analysis of the Zeus botnet crimeware toolkit. In *Conference on Privacy Security and Trust*, Ottawa, Ontario, Canada, Aug. 2010.

[5] S. Bromberger. DNS as a covert channel within protected networks. Technical report, National Electronic Sector Cyber Security Organization, 2011. http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/DNS_Exfiltration_2011-01-01_v1.1.pdf.

[6] D. Dagon, M. Antonakakis, P. Vixie, T. Jinmei, and W. Lee. Increased DNS forgery resistance through 0x20-bit encoding: security via leet queries. In *ACM Computer and Communications Security (CCS)*, pages 211–222, Alexandria, VA, USA, Oct. 2008.

[7] D. Dagon, N. Provos, C. P. Lee, and W. Lee. Corrupted DNS resolution paths: The rise of a malicious resolution authority. In *Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, Feb. 2008.

[8] C. J. Dietrich, C. Rossow, F. C. Freiling, H. Bos, M. van Steen, and N. Pohlmann. On botnets that use DNS for command and control. In *European Conference on Computer Network Defense*, pages 9–16, Gothenburg, Germany, Sept. 2011.

[9] J. Gordon. Systems and methods for identifying a network. Patent no. US8353007 B2, Filed October 13th, 2009, Issued January 8th, 2013.

[10] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling. Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm. In *Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, San Francisco, CA, USA, Apr. 2008.

[11] Kenton and D. Gustafson. Detecting DNS tunnels using character frequency analysis. In *Annual Security Conference*, Las Vegas, NV, USA, Apr. 2010.

[12] J. Kristoff. Botnets. In *32nd Meeting of the North American Network Operators Group*, Reston, VA, USA, Oct. 2004.

[13] J. Levine. DNS blacklists and whitelists. RFC 5782, proposed standard, Feb. 2010.

[14] J. Li, B. K. Chandrasekhar, and K. Y. Chan. Updating of malicious code patterns using public DNS servers. Patent no. US8171467 B1, Filed July 3th, 2007, Issued May 1st, 2012.

[15] A. Merlo, G. Papaleo, S. Veneziano, and M. Aiello. A comparative performance evaluation of DNS tunneling tools. In *Conference on Computational Intelligence in Security for Information Systems*, pages 84–91, Torremolinos-Málaga, Spain, June 2011.

[16] C. Mullaney. Morto worm sets a (DNS) record. Technical report, Symantec, 2011. http://www.symantec.com/connect/blogs/morto-worm-sets-dns-record.

[17] Nijhof. Element53. http://www.nijhof.biz/pages/project/171/Element53, DNS tunneling app for Android.

[18] L. Nussbaum, P. Neyron, and O. Richard. On robust covert channels inside DNS. In *Information Security Conference*, volume 297, pages 51–62. Pafos, Cyprus, May 2009.

[19] OpenDNS.com. The role of DNS in botnet command & control. Technical report, 2012. http://info.opendns.com/rs/opendns/images/OpenDNS_SecurityWhitepaper-DNSRoleInBotnets.pdf.

[20] C. Qia, X. Chenb, C. Xud, J. Shia, and P. Liub. A bigram based real time DNS tunnel detection approach. In *Information Technology and Quantitative Management (ITQM)*, Suzhou, China, May 2013.

[21] D. Raman, B. D. Sutter, B. Coppens, S. Volckaert, K. De, P. D. Bosschere, and E. V. Buggenhout. DNS tunneling for network penetration. In *International Conference on Information Security and Cryptology (ICISC)*, Seoul, Korea, Nov. 2012.

[22] M. Richardson and D. Redelmeier. Opportunistic encryption using the internet key exchange (IKE). RFC 4322, informational, Dec. 2005.

[23] R. Rosenbaum. Using the domain name system to store arbitrary string attributes. RFC 1464, experimental, May 1993.

[24] T. van Leijenhorst, D. Lowe, and K. Chin. On the viability and performance of DNS tunneling. In *Conference on Information Technology and Applications*, Cairns, Queensland, Australia, June 2008.

[25] F. Weimer. Passive DNS replication. In *17th FIRST Conference on Computer Security Incident*, Singapore, June 2005.

[26] M. Wong and W. Schlitt. Sender policy framework (SPF) for authorizing use of domains in e-mail, version 1. RFC 4408, experimental, Apr. 2006.

[27] K. Xu, P. Butler, S. Saha, and D. Yao. DNS for massive-scale command and control. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 10(3):143–153, 2013.