Autocorrelation Analysis of Financial Botnet Traffic

Prathiba Nagarajan[1], Fabio Di Troia[1], Thomas H. Austin[1], and Mark Stamp[1]
[1]*Department of Computer Science, San Jose State University, San Jose, California*
prathiba.rajan@gmail.com, fabioditroia@msn.com, thomas.austin@sjsu.edu, mark.stamp@sjsu.edu

Keywords: Botnet; autocorrelation; periodicity; Citadel; SpyEye; Zeus; Tinba.

Abstract: A botnet consists of a network of infected computers that can be controlled remotely via a command and control (C&C) server. Typically, a botnet requires frequent communication between its C&C server and the infected nodes. Previous approaches to detecting botnets have included various machine learning techniques based on features extracted from network traffic. In this research, we conduct autocorrelation analysis of traffic generated by several financial botnets, and we show that periodicity in the network traces can be used to distinguish these botnets from each other.

1 Introduction

Periodic patterns can often shed light on the characteristics of an underlying process. For example, periodicity of network traffic has been used to analyze network congestion (He et al., 2009). The focus of this paper is on the analysis of periodicity features extracted from botnet traffic. Specifically, we consider features related to each of DNS, HTTP, and TCP traffic collected from several botnets. We perform autocorrelation analysis on these features and show that over a sample of four financial botnets, these features are highly distinguishing—to the point that we could create a distinct periodicity profile for each of these four well-known financial botnets.

The remainder of this paper is organized as follows. Section 2 includes background information, with the emphasis on botnets and their associated communications strategies and protocols. In Section 3, we briefly discuss relevant related work. Section 4 provides details on a variety of experiments that we have performed and the results that we obtained. Finally, our conclusions and suggestions for future work can be found in Section 5.

2 Background

In this section, we discuss various relevant aspects of botnets. In particular, we focus our attention on botnet communication, as this is the feature analyzed in the research discussed in this paper.

2.1 Botnet Basics

Also known as a "zombie army" (Hachem et al., 2011), botnet-infected nodes are typically controlled by a command and control (C&C) server, which acts as the so-called botmaster. The C&C server is used to send commands to direct the infected nodes to perform malicious activities such as distributed denial of service (DDoS) attacks, collecting sensitive information from infected hosts, financial theft, and so on. Botnets are capable of inflicting significant financial harm (Sood et al., 2016; Bottazzi and Me, 2015).

Botnets utilize a wide variety of techniques to propagate the bot infection (Bailey et al., 2009). In a typical scenario, a botnet uses malware as a means to recruit additional nodes. The nodes that become infected communicate with the C&C server without the legitimate user's knowledge. After infection, the botmaster may send commands to the infected computers or may poll the bots—or the bots may poll a C&C server—on a regular basis. The C&C server might also provide software updates to the infected bots, as necessary.

C&C servers play a critical role not only in the spread of botnets, but also in the long-term survival of botnets (Tiirmaa-Klaar et al., 2013). If a C&C server is taken down, or the communication channel is interrupted, the botnet army is likely to become useless for malicious activities.

Botnets can adopt either centralized or distributed networking models for their activities. Centralized models can be implemented using hierarchical or star topologies, with single or multiple server at the core.

While there is essentially no communication latency in a centralized model, the downside is that the server is a single point of failure. Distributed (e.g., peer-to-peer) topologies do not suffer from this single point of failure issue, but message delivery to the infected bots is much more challenging.

2.2 Botnet Communication

The IRC and HTTP protocols are popular for botnet communication. IRC facilitates communication between clients in the form of text, where a chat server acts as an intermediary to transfer messages between clients. Due to the scalability and flexibility of the IRC protocol (Butts and Shenoi, 2011), it would seem to be ideal for a botnet application.

The HTTP protocol is also commonly used by botnets. One advantage of an HTTP botnet is that its traffic will tend to blend into the vast background of HTTP traffic. Furthermore, HTTP bots frequently exploit bugs or compromised websites to communicate with infected nodes (Hachem et al., 2011).

Regardless of the protocol used, a botmaster can communicate with his bots in either a push or pull mode (Hachem et al., 2011). In a push communication mode, the botmaster sends commands to bots, typically via broadcast methods. In contrast, for pull communication, an infected node must contact the botmaster, which typically involves periodically polling the botmaster. For botnets, an advantage of push communication is that it is easier to coordinate attacks, while an advantage of pull communication is that it is likely to be considerably stealthier. Of course, a botnet could employ a combination of both push and pull communication.

3 Related Work

In (Tyagi et al., 2015), the focus is on detecting periodicity of related content in a particular flow using n-gram analysis and deep packet inspection. The authors cluster flows and compute a similarity score based on a distance metric and timing analysis. These techniques are tested on a synthetic dataset based on the Zeus botnet, and achieve a 98% detection accuracy. While these results are intriguing, the reliance on simulated data and the small size of the experiments are significant limitations of this research.

The authors of the work in (Stevanovic and Pedersen, 2015) and (Beigi et al., 2014) extract more than 20 network-based features related to DNS, TCP and UDP traffic and use these features to classify data from 40 botnets. The authors apply a variety of classifiers based on random forests and in the best case obtain a detection accuracy of 99%.

The research in (Jin et al., 2015) consists of creating a database of authoritative name server records (i.e., DNS TXT records) and then using this information to differentiate traffic. This research achieves a “hit rate” of 19% per day. However, this research relies on fixed IP addresses, which may not stay constant over an extended period of time.

A decision tree classifier is applied to byte-based, duration-based, behavior-based, and packet-based features of botnet traces in (Beigi et al., 2014). These authors achieve detection rates that range from 75% to 99%.

In (Adamov et al., 2014), the authors conducted a study of popular botnets, based on the type of files downloaded, protocols used, and encryption information, along with various characteristics of the bot commands. The focus of this work is on anomalous botnet traffic. While these results are interesting, the general applicability of the method is not clear.

Various periodicity characteristics of network features in botnets are discussed in (Garcia et al., 2014). Using Markov chains based on the transport layer protocol, they achieve an F-measure of 93%.

The work in (Eslahi et al., 2015) uses a variety of metrics, including range of frequencies, and time sequencing to determine periodicity in HTTP botnets. A decision tree is used for classification.

4 Experiments and Results

When examining botnet packet captures, we found network communication in TCP and DNS to be common across all botnets that we considered. Furthermore, the HTTP protocol under TCP has a special significance, as many botnets use HTTP for communication (via HTML). Hence, this research focuses on features extracted from DNS, HTTP, and TCP.

4.1 Feature Selection

The features we consider have been selected while keeping in mind techniques such as tunneling and domain generation algorithms (DGA). Note that DGAs are often used by botnets to generate different domain names periodically, which can serve to make it far more difficult to shut down the C&C server. In addition, botnets frequently use tunneling to piggyback data from one protocol on top of another. By careful use of tunneling techniques, botnets can often evade firewall defenses.

The specific DNS, HTTP, and TCP features we have selected for analysis are summarized in Table 1. Note that these features capture a wide variety of characteristics of each of these protocols.

Table 1: Feature sets

Protocol	Feature	Description
DNS	query type	type of query
	response type	type of response
	frame length	length of data frame
	frame delta yime	time elapsed
	query name	SLD name
	response TTL	response time to live
	answer count	number of answers
	response code	response code
	response length	length of response
	length of domain	length of SLD
	domain digits	digits in SLD
	bigram query score	domain bigram score
HTTP	request method	method used
	content type	type of content
	response code	status code
	content length	length of content
	URL length	length of URLs
	cache time	time of cache
	cache type	type of cache
	header elements	no. header elements
	bigram score	data bigram score
	request interval	timings
	time since request	elapsed time
TCP	keep alive status	keep-alive flag
	segment length	TCP segment length
	flags status	no. of flags set
	iRTT	initial estimate RTT
	length of connection	duration
	port	port number

4.2 Autocorrelation

There are various methods available to identify periodicity in a time domain signal. For example, the discrete Fourier transform (DFT) maps a time domain signal into the frequency domain, where periodicity features are easy to distinguish.

For the research presented in this paper, we rely on autocorrelation plots for each of the multitude of features listed in Table 1. Autocorrelation consists of the cross-correlation of a signal with itself. Autocorrelation plots enable us to easily determine the presence of periodic cycles in a signal via a simple visual inspection, and we can also determine the dominant periods in periodic signals.

Figure 1 gives an example of a periodic signal and the corresponding autocorrelation sequence. From the autocorrelation plot, we see that we can easily deduce the dominant period (or periods) of the underlying sequence. Another significant benefit to autocorrelation analysis is that it works well, even in the presence of considerable background noise.

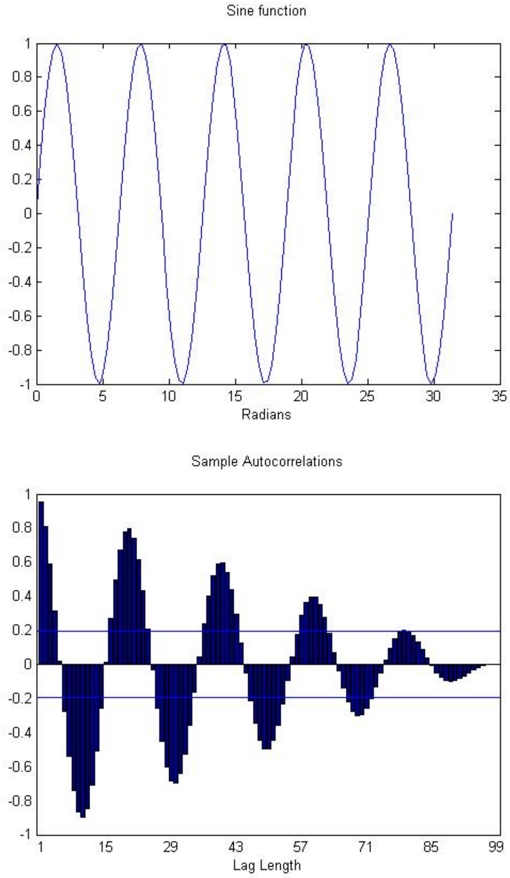


Figure 1: Autocorrelation of a periodic signal

For comparison, we have given an example of an aperiodic sequence in Figure 2, along with its autocorrelation plot. In this case, the autocorrelation sequence reveals that there is no periodic information present in the original signal.

4.3 Data

The datasets used in our experiments are all publicly available. The Contagio malware dump (Parkour, 2015) consists of a large number malware samples, as well as botnet packet captures and binaries. The Stratosphere IPS dataset (Garcia et al., 2014) is a sister project of the malware capture facility project (MCFP), and includes packet capture and binaries of more than 200 botnet traces. The ISCX dataset (Beigi et al., 2014) contains 13 traces of botnet packet captures, and also includes benign data. In this paper,

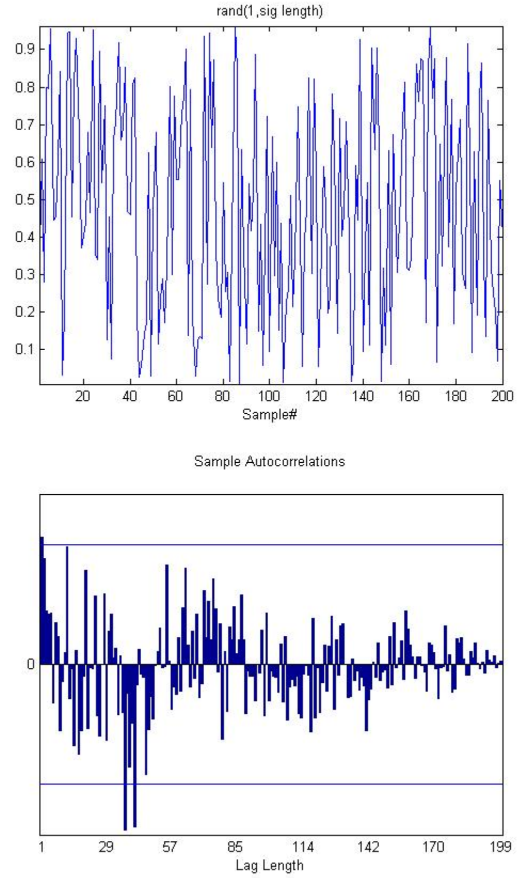


Figure 2: Autocorrelation of an aperiodic signal

we report on periodicity experiments involving the financial botnet data, as found in the aforementioned datasets.

Specifically, the experiments reported here were performed on the following four financial botnets: Citadel, SpyEye, Zeus, Tinba. These botnets have been used to steal credentials, enable banking fraud, and target financial institutions, among other malicious activities. Here, we provide a brief summary of each of the financial botnets considered in this paper.

- Citadel is a sophisticated botnet that is considered on offshoot of Zeus. This botnet includes a variety of stealth features and can be used to steal credentials via keystroke logging, screen capture, and video capture. The Zeus botnet is available as a kit that was reported to sell for more than $3000 in 2012 (Segura, 2012). As of 2013, Citadel had been implicated in the theft of more than $500M (BBC News, 2013).
- SpyEye includes such advanced features as keystroke logging, encrypted config files, an "authorization grabber", and a "Zeus killing" feature, which serves to eliminate any possible competition from the popular Zeus botnet (Coogan, 2010). In an all-too-rare success for law enforcement, the developers of SpyEye were caught and recently sentenced to long prison terms (Ribeiro, 2016).
- Zeus (also known as Zbot, among many other names) is one one of the oldest and most successful financial botnets. As a result of its success, Zeus has spawned a vast number of variants. The Zeus botnet is primarily focused on stealing credentials and it employs a wide variety of means to do so. Interestingly, some versions of this botnet include sophisticated hardware-based license protections to prevent unauthorized redistribution (Stevens and Jackson, 2010).
- Tinba uses fake messages and fake web forms to try to convince users to divulge their banking credentials. In spite of being considered one of the smallest examples of malware ever created, Tinba includes a significant number of resiliency features designed to make it difficult to defeat. For example, Tinba uses public key cryptography to ensure that any updates come from an authorized botmaster (Bach, 2015).

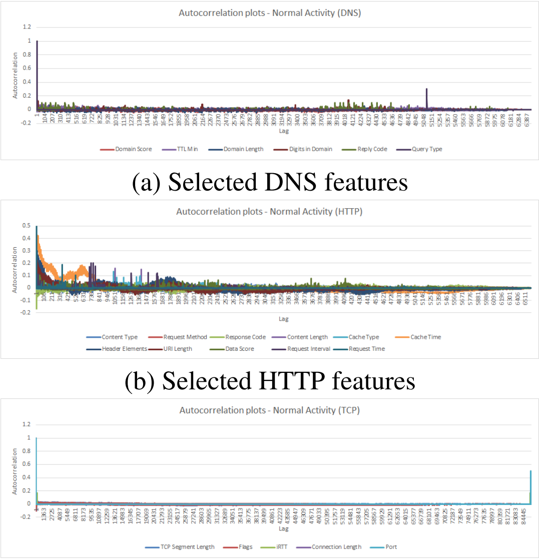
The collected network traffic from botnet binaries is in the form of pcap data. Thus, for comparison, we collected benign (i.e., non-botnet) activity also in the form of pcap files. Of course, we also analyzed botnet and benign traces obtained from the datasets mentioned above. In all cases, protocol dissectors written in Lua and executed in TShark (tshark, 2017) were used to extract the relevant features.

4.4 Experimental Results

In this section, we summarize some of our main results. First, we present autocorrelation results for background traffic, followed by a similar analysis of four financial botnets. Then we discuss the relevance of these results to botnet analysis. Note that a large number of additional autocorrelation results—and related results—can be found in the report (Nagarajan, 2017).

4.4.1 Background Network Activity

The raw data for the benign features considered here is primarily from the Stratosphere IPS dataset. Figure 3 (a) shows the autocorrelation graphs for selected DNS features, while Figures 3 (b) and (c) are analogous plots for selected HTTP and TCP features, respectively. Note that these autocorrelation graphs do not reveal any significant periodicity with respect to any of these features in the background data.

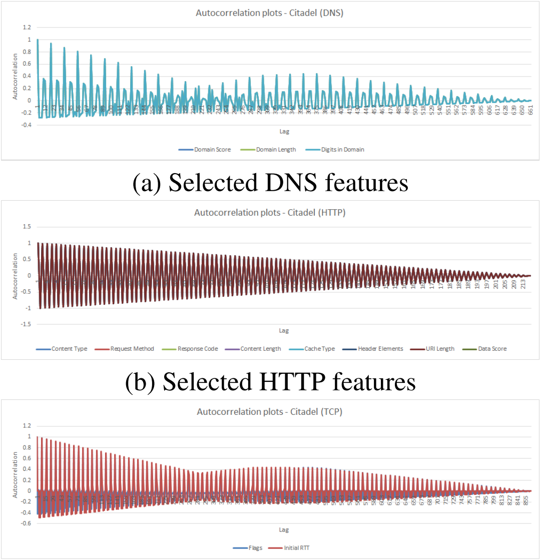


(a) Selected DNS features

(b) Selected HTTP features

(c) Selected TCP features

Figure 3: Background activity autocorrelation plots



(a) Selected DNS features

(b) Selected HTTP features

(c) Selected TCP features

Figure 4: Citadel autocorrelation plots

Next, we present autocorrelation plots for traffic extracted from each of the four financial botnets under consiseration. In all cases, we observe highly periodic results for multiple features within one or more of the types of traffic under consideration.

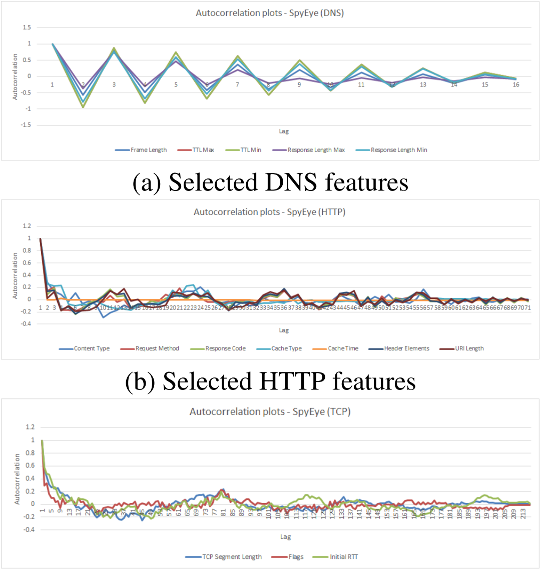
4.4.2 Financial Botnet Network Activity

In this section, we consider the periodicity of network traffic generated by a representative sample of financial botnets. In each case, we present DNS, HTTP, and TCP autocorrelation plots, based on selected features from those listed in Table 1. We also specify the dominant period of each periodic feature.

Citadel: For the Citadel botnet, we consider network traces obtained from the Contagio dataset. All three of the Citadel autocorrelation plots in Figures 4 show strong periodicity. The DNS and TCP autocorrelation in Figures 4 (a) and (c), respectively, provide clear evidence of a short period of about 11 and (somewhat less clear) evidence of a longer period of about 350. The HTTP plot in Figure 4 (b) shows a strong and unambiguous period of 2. In this latter case, an examination of the corresponding packets reveals similar requests and responses (but with different data) being sent repeatedly.

The periodicity of Citadel traffic is striking. It is appears that periodicity would likely be a useful feature for classifying traffic from this particular botnet.

SpyEye: Like Citadel, selected DNS features of SpyEye show periodicity. However, in the case of SpyEye, the DNS periodicity is 2, as can be deduced from the autocorrelation plot in Figure 5 (a). The HTTP autocorrelation plots for SpyEye given in Figure 5 (b) show a weak periodicity of about 23, which is surrounded by fairly significant noise. The TCP features in Figure 5 (c) seem to have a weak periodicity at around 81 packets, but more data would be needed for confirmation.



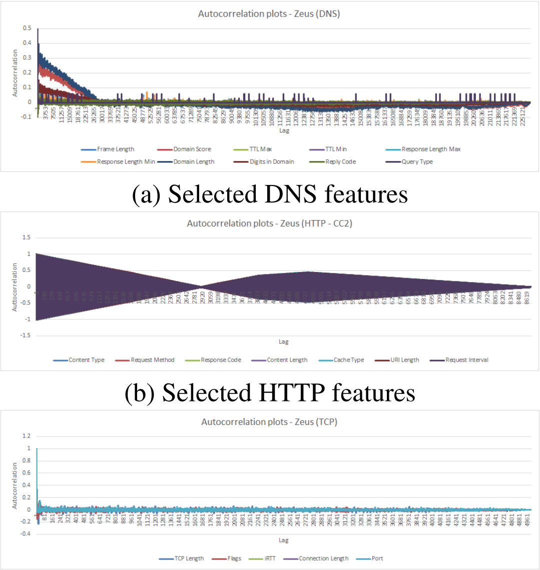
(a) Selected DNS features

(b) Selected HTTP features

(c) Selected TCP features

Figure 5: SpyEye autocorrelation plots

While Citadel shows strong periodicity for selected DNS, HTTP, and TCP features, the results in Figure 5 show that SpyEye only provides similarly



(a) Selected DNS features

(b) Selected HTTP features

(c) Selected TCP features

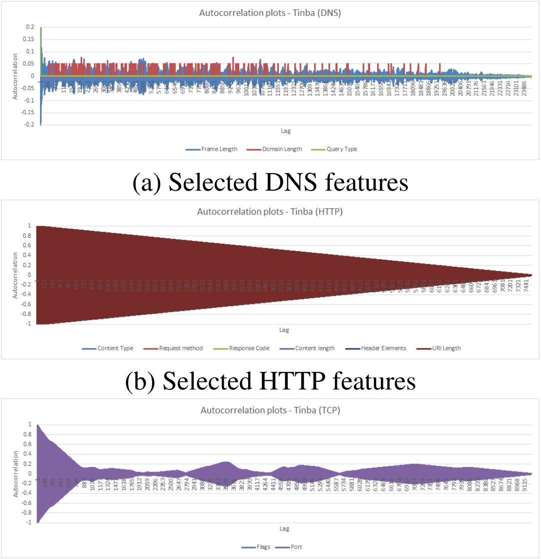
Figure 6: Zeus autocorrelation plots

strong results for its DNS traffic, with only weak periodicity in its HTTP and TCP traffic. Nevertheless, the periodicity in the DNS traffic for SpyEye is quite different than what we expect to see in background traffic, as shown in Figure 3. Thus, periodicity analysis also appears to be a strong feature for distinguishing the SpyEye botnet from Citadel.

Zeus: The raw Zeus traffic considered here was obtained from the Stratosphere IPS dataset. As can be seen from the autocorrelation plots in Figure 6 (b), the Zeus botnet is highly periodic with respect to selected HTTP features (with period 2). Based on the plots in Figure 6 (a) and (c), we do not observe significant periodicity in DNS or TCP features of Zeus.

Analogous to SpyEye, for Zeus we observe strong periodicity in one type of traffic, but not for the other two types. But, for SpyEye the strong periodicity was in the DNS traffic, whereas the strong periodicity in Zeus is in the HTTP traffic. In any case, strong periodicity in any one type of traffic indicates that we have a potentially useful feature for distinguishing traffic generated by this particular botnet.

Tinba: Our analysis of Tinba is based on raw traces taken from the Stratosphere IPS dataset. From the autocorrelation results in Figures 7 (b) and (c), we see that Tinba is highly periodic with respect to all of the selected HTTP and TCP features. Based on Figure 7 (a), it appears that there may be a weak (and long) periodicity in Tinba DNS traffic, but this cannot be reliably determined without substantially more data.



(a) Selected DNS features

(b) Selected HTTP features

(c) Selected TCP features

Figure 7: Tinba autocorrelation plots

While the overall periodicity in Tinba is not as strong as Citadel, it is stronger than Zeus or SpyEye. In fact, the periodicity found in each of the botnets is distinct, and could serve to distinguish the bots from each other. We discuss the issue further in the next section.

4.5 Discussion

Each of the four financial botnets analyzed here shows strong periodicity for multiple features in at least one of the types of traffic considered (i.e., DNS, HTTP, and TCP). As a results, we can go beyond the discussion above to construct a specific “signature” for each botnet based on its various periodicity features. The relevant information for each of the four financial botnets under consideration is summarized in Table 2, where we have shorthanded various features, e.g., “request” corresponds to any of the request-related features listed in Table 1. Based on the results in Table 2, it is clear that periodicity analysis provides a fingerprint for each of the four financial botnets considered in this paper. Significantly, these fingerprints are sufficient to distinguish traffic generated by any of these four botnets from all of the others.

The point here is that we can clearly distinguish each of these four financial botnets from each other based on a periodicity analysis of a subset of the features in Table 1. Furthermore, if we account for the period lengths, it would appear that we obtain highly discriminating signatures in each case.

Table 2: Financial botnet traffic-based signatures

Botnet	Periodic features		
	DNS	HTTP	TCP
Citadel	domain	content	flags initial RTT
SpyEye	frame TTL response	—	—
Zeus	—	content request response cache URL	—
Tinba	—	content request response header URL	flags port

5 Conclusions and Future Work

We considered four financial botnets and analyzed the periodicity of their DNS, HTTP, and TCP traffic, based on autocorrelation plots. In each case, we found strong periodicity features in at least one of these traffic types, while background traffic did not exhibit any significant level of periodicity. Thus, autocorrelation analysis of botnet traffic would enable us to distinguish between the network activity of the four financial botnets under consideration. That is, we can construct highly discriminating signatures for each of these four financial botnets based on periodicity features, and their periods.

For future work, we will analyze the effectiveness of the periodicity-based analysis presented in this paper in a realistic networked environment. In such a case, there will be a large volume of background noise, and our goal is to determine how well we can distinguish botnet traffic (based on periodicity features) in such a noisy environment. We believe the features discussed here will prove strong by themselves and, of course, we can consider combinations of periodicity features with other aspects of botnet behavior. This problem seems ideally suited to the application of machine learning techniques and we plan to apply a wide variety of such techniques. Hidden Markov models (HMM) (Stamp, 2004), profile hidden Markov models (PHMM) (Durbin et al., 1998), support vector machines (SVM) (Berwick, 2003), and neural networks (Mukkamala et al., 2002) would appear to be obvious candidates for application to this particular problem.

REFERENCES

Adamov, A., Hahanov, V., and Carlsson, A. (2014). Discovering new indicators for botnet traffic detection. In *Proceedings of IEEE East-West Design Test Symposium (EWDTS 2014)*, pages 1–5.

Bach, O. (2015). Tinba: Worlds smallest malware has big bag of nasty tricks. https://securityintelligence.com/tinba-worlds-smallest-malware-has-big-bag-of-nasty-tricks/. Accessed 2017-10-15.

Bailey, M., Cooke, E., Jahanian, F., Xu, Y., and Karir, M. (2009). A survey of botnet technology and defenses. In *2009 Cybersecurity Applications Technology Conference for Homeland Security*, pages 299–304.

BBC News (2013). FBI and Microsoft take down $500m-theft botnet Citadel. *BBC News*, http://www.bbc.com/news/technology-22795074. Accessed 2017-10-15.

Beigi, E. B., Jazi, H. H., Stakhanova, N., and Ghorbani, A. A. (2014). Towards effective feature selection in machine learning-based botnet detection approaches. In *2014 IEEE Conference on Communications and Network Security*, pages 247–255.

Berwick, R. (2003). An idiots guide to support vector machines (SVMs). http://www.svms.org/tutorials/Berwick2003.pdf.

Bottazzi, G. and Me, G. (2015). *A Survey on Financial Botnets Threat*. Springer International Publishing, Cham.

Butts, J. and Shenoi, S. (2011). *Critical Infrastructure Protection V: 5th IFIP WG 11.10 International Conference on Critical Infrastructure Protection, Revised Selected Papers*. Springer Berlin Heidelberg.

Coogan, P. (2010). SpyEye bot versus Zeus bot. https://www.symantec.com/connect/blogs/spyeye-bot-versus-zeus-bot. Accessed 2017-10-15.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.

Eslahi, M., Rohmad, M. S., Nilsaz, H., Naseri, M. V., Tahir, N., and Hashim, H. (2015). Periodicity classification of http traffic to detect http botnets. In *2015 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pages 1–5.

Garcia, S., Grill, M., Stiborek, J., and Zunino, A. (2014). An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123.

Hachem, N., Mustapha, Y. B., Granadillo, G. G., and Debar, H. (2011). Botnets: Lifecycle and taxonomy. In *2011 Conference on Network and Information Systems Security*, pages 1–8.

He, X., Papadopoulos, C., Heidemann, J., Mitra, U., and Riaz, U. (2009). Remote detection of bottleneck links using spectral and statistical methods. *Computer Networks*, 53(3):279–298.

Jin, Y., Ichise, H., and Iida, K. (2015). Design of detecting botnet communication by monitoring direct outbound dns queries. In *2015 IEEE 2nd International Confer-*

ence on Cyber Security and Cloud Computing, pages 37–41.

Mukkamala, S., Janoski, G., and Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In *Proceedings of the 2002 International Joint Conference on Neural Networks*, volume 2 of *IJCNN'02*, pages 1702–1707. IEEE.

Nagarajan, P. (2017). Analysis of periodicity in botnets. Master's Project, Department of Computer Science, San Jose State University. `http://scholarworks.sjsu.edu/etd_projects/544/`.

Parkour, M. (2015). Collection of pcap files from malware. *Contagio Malware Dump*. Accessed 2016-11-20.

Ribeiro, J. (2016). SpyEye botnet kit developer sentenced to long jail term. *PCWorld*, `https://www.pcworld.com/article/3059557/spyeye-botnet-kit-developer-sentenced-to-long-jail-term.html`. Accessed 2017-10-15.

Segura, J. (2012). Citadel: A cyber-criminals ultimate weapon? `https://blog.malwarebytes.com/threat-analysis/2012/11/citadel-a-cyber-criminals-ultimate-weapon/`. Accessed 2017-10-15.

Sood, A. K., Zeadally, S., and Enbody, R. J. (2016). An empirical study of http-based financial botnets. *IEEE Transactions on Dependable and Secure Computing*, 13(2):236–251.

Stamp, M. (2004). A revealing introduction to hidden Markov models. `https://www.cs.sjsu.edu/~stamp/RUA/HMM.pdf`.

Stevanovic, M. and Pedersen, J. M. (2015). An analysis of network traffic classification for botnet detection. In *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, pages 1–8.

Stevens, K. and Jackson, D. (2010). ZeuS banking trojan report. `https://www.secureworks.com/research/zeus`. Accessed 2017-10-15.

Tiirmaa-Klaar, H., Gassen, J., Gerhards-Padilla, E., and Martini, P. (2013). *Botnets*. SpringerBriefs in Cybersecurity. Springer London.

tshark (2017). tshark — the Wireshark network analyzer. `https://www.wireshark.org/docs/man-pages/tshark.html`. Accessed 2017-10-15.

Tyagi, R., Paul, T., Manoj, B. S., and Thanudas, B. (2015). A novel http botnet traffic detection method. In *2015 Annual IEEE India Conference (INDICON)*, pages 1–6.