Large Scale DNS Analysis

Samuel Marchal and Thomas Engel

SnT - University of Luxembourg, Luxembourg,
firstname.lastname@uni.lu

Abstract. In this paper we present an architecture for large scale DNS monitoring. The analysis of DNS traffic is becoming of first importance currently, as it allows to monitor the main part of the interactions on the Internet. DNS traffic can reveal anomalies such as worm infected hosts, botnets or spam participating hosts. The efficiency and the speed of detection of such anomalies rely on the capacity of DNS monitoring system to treat quickly huge quantity of data. We propose a system that leverages distributed processing and storage facilities.

Keywords: Security monitoring, DNS data, storage facilities

1 Introduction

The DNS service [10, 11] is the glue that holds and drives the current Internet. In addition to a basic and essential function that maps names to IP addresses, attackers also leverage DNS to enhance the impact of attacks. For instance Fast Flux networks and Double Flux networks are the typical crimeware supporting infrastructures. Fluxing techniques allow to relocate hosts providing malware rapidly, while keeping the same contact reference: the domain name. In order to detect such a behaviour and to be able to identify domain names implicated in criminal activities, the fine choice of the DNS features is paramount. Behind the concern of malicious domains, the question arises of automatically discovering the activity type of domain. The other big issue is the storage facilities to use for these features to have a fast access to them.

2 Passive DNS Analysis

2.1 Early phase of research

Studies on several malicious activities showed that they can be characterized by statistical properties of basic protocol fields (TTL, Request type, domain names). This is a good starting point that still needs to be extended with derived metrics suited for detecting deviations from a normal profil. Major issues with data mining for DNS data are related to identify relevant features that can be used to detect abnormal behaviour.

In [9], we present the first steps of our work showing the utility of DNS features in the identification of domain activities. This presents a passive DNS security monitoring framework called DNSSM whose architecture for DNS data capture (see Figure 1) is based on the technique of Weimer [14]. We collect DNS responses sent to a recursive server and extract ten features for each domain to build a representative pattern of it. We use common features of DNS data mining such as the TTL or the number of IP addresses associated to a single domain. Additionally we introduce new features such as an entropy based IP dispersion index that depicts the number of different networks in a pool of IP addresses and thus shows the potential geographical dispersion of physical machines.

These patterns can be further used for comparison with other domain patterns in order to raise singular anomalies, or to group them corresponding to their similarities. We present a method that offers two methods: first, through a Web Interface where the analyst can easily see the characteristics of a domain and compare them to the average or to extreme values of numerical features. Second, we present an automatic analysis leveraging the k-means clustering method [5] based on ten DNS features. Data mining 300 MB of DNS responses allows to group together domains participating to CDNs, domains that performs user tracking services or very popular domains such as *google.com* or *facebook.com.* Through this analysis of a small dataset, we show that DNS features are significant in domain activity.

2.2 Distributed data storage

A critical requirement for such an architecture is the large quantity of data that has to be processed and stored. Our experience in deploying a passive DNS monitoring tool showed that for a regional backbone network, the daily quantity of data can easily reach 1 GB per day. In our first trial, we have implemented the storage system using a relational database system (MySQL). Since tracking malicious domains over a monthly basis can lead to deal with data quantities in the tenths of GBs such that we have to leverage distributed storage and retrieval solutions. Existing approaches leverage efficient key-value storage systems (see `Cassandra` [6] or [7]).

Most activities related to DNS security monitoring require small processes running over a very large database. For instance, looking for the IP address corresponding to a domain is an easy task, but requires to mine a huge volume of data. Thus, the paradigm has shifted from a highly computational to a data-intensive problem. We propose to use the popular `Hadoop` framework [15] in order to distribute both the computations and data storage.

`Hadoop` implements the `MapReduce` [4, 8] design pattern, which is designed for data-intensive problems and to distribute computing of large datasets on clusters of computers. However, for achieving the same task as the centralized approach of MySQL, the design of our DNS analysis algorithm has to be rethought, because mapping is applied onto each piece of distributed data, on each machine. Basically data is processed to extract required features, with the domain name being the key. Later, these key features are used to aggregate results, because all outputs of the mappers with the same key are sent to a unique reducer in

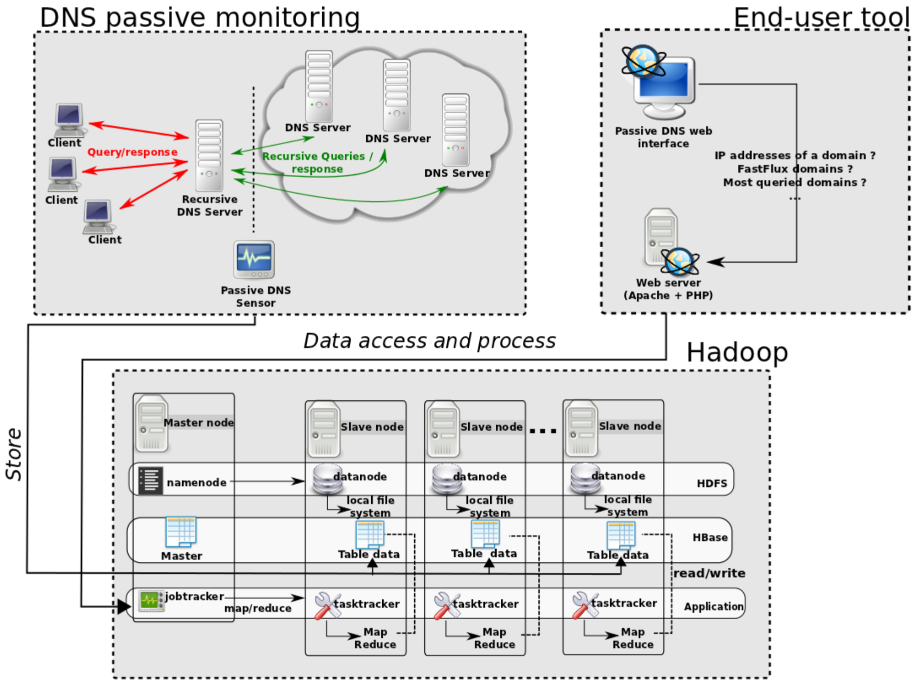

Fig. 1. Hadoop-based DNSSM architecture

charge of producing the final result. This is an issue we are addressing to build the architecture proposed in Figure 1.

3 Related Work

The first work of Florian Weimer [14], opened the door to DNS analysis with a first proposal to analyse passively collected DNS data. Based on this work, a lot of research has been done to detect DNS anomalies such as DNS tunnel in [3] or FastFlux and botnet in [12, 1] and [2]. These papers present data mining techniques leveragning several DNS features and mainly supported by machine learning in automatic detection. The difference to our work is that DNSSM doesn't relies on labeled data as it uses unsupervised clustering techniques. Dealing with large quantities of DNS data has already been addressed in [13], where Plonka et al. use binary trees to store DNS data using mapped IP addresses as keys.

4 Conclusion and Future Work

In this paper, we present a method to identify and compare domain names activities leveraging features that are extracted from DNS response packets. We propose an architecture to collect and mine data manually or automatically. We showed in a small scale analysis (around 300 MB of data) the ability of our framework to group domain names regarding their activities using the k-means clustering method. We are currently extending this architecture to fit with bigger volume of DNS data by using a distributed data storage and computation system (`Hadoop`).

In further work we will use this new data storage architecture to apply our technique to bigger data sets such as those from recursive DNS servers from an ISP or similar in order to detect more DNS anomalies. We also plan to extend the approach with more relevant features that can typify traffic more accurately. Finally, passive DNS data will be correlated with data gathered from honeypots to protect a targeted network from malicious domains.

References

1. Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., Feamster, N.: Building a dynamic reputation system for dns. In: Proceedings of the 19th USENIX conference on Security. pp. 18–18. USENIX Association, Berkeley, CA, USA (2010)
2. Bilge, L., Kirda, E., Kruegel, C., Balduzzi, M.: Finding malicious domains using passive dns analysis. In: NDSS'11, 18th Annual Network & Distributed System Security Symposium, 6-9 February 2011, San Diego, California, USA (2011)
3. Born, K., Gustafson, D.: Detecting dns tunnels using character frequency analysis. CoRR abs/1004.4358 (2010)
4. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. In: Symposium on Opearting Systems Design & Implementation (OSDI). USENIX Association (2004)
5. Hartigan, J.A., Wong, M.A.: A k-means clustering algorithm. Applied Statistics 28 (1979)
6. Lakshman, A., Malik, P.: Cassandra: structured storage system on a p2p network. In: Proceedings of the 28th ACM symposium on Principles of distributed computing. pp. 5–5. PODC '09, ACM, New York, NY, USA (2009)
7. Lerner, R.M.: At the forge: Redis. Linux J. 2010 (September 2010), http://dl.acm.org/citation.cfm?id=1883519.1883524
8. Lin, J., Dyer, C.: Data-Intensive Text Processing with MapReduce (Synthesis Lectures on Human Language Technologies). Morgan and Claypool Publishers (2010)
9. Marchal, S., François, J., Wagner, C., State, R., Dulaunoy, A., Engel, T., Festor, O.: DNSSM: A large scale passive DNS security monitoring framework. NOMS'12 (2012)
10. Mockapetris, P.: Rfc 1034: Domain names - concepts and facilities (1987)
11. Mockapetris, P.: Rfc 1035: Domain names - implementation and specification (1987)
12. Perdisci, R., Corona, I., Dagon, D., Lee, W.: Detecting malicious flux service networks through passive analysis of recursive dns traces. In: Proceedings of the 2009 Annual Computer Security Applications Conference. pp. 311–320. ACSAC '09, IEEE Computer Society, Washington, DC, USA (2009)
13. Plonka, D., Barford, P.: Context-aware clustering of dns query traffic. In: Internet Measurement Comference'08. pp. 217–230 (2008)
14. Weimer, F.: Passive dns replication (2005)
15. White, T.: Hadoop: The Definitive Guide. O'Reilly Media (June 2009)