Development of Bad Domain Tracking System

Cheng, Ti-Chung
August 2015

1.0 Abstract

With rise of Big Data techniques and rapid change of how hackers camouflage malicious domains. This report presents the development of Bad Domain Tracking System (BDTS). It was developed to provide automatic tracking between relationships of malicious (or bad) Domain names as well as its IP rapidly over a long period of time which was targeted as not shorter than 3 years. This system is able to obtain a record of bad domains and query through the dig functions provided on Linux systems. The results are then saved into the database. This system is also able to provide visualization with combination of the MD5 database already obtained.

2.0 Introduction and Motivation

With internet of things approaching and growing, malicious internet activities, phishing and other hacking activities has been observed throughout the internet. These methods of malicious attacks are transforming quickly and hard to imagine. To better collect sample with use of Big Data technologies. Bad Domain Tracking System (BDTS) was developed to observe the relationship between Domain Names and its corresponding IP addresses. BDTS can be covered and introduced from the following perspectives: the core scraping system, database design, intelligent exit mechanism for domains and Visualization.

Over the past years, many researchers has been putting their focus in relevant categories, however most of them are limited to the amount of data and time allowed to their research, the BDTS is currently designed to process any malicious domains continuously through a timeframe of at least three years. The team believe that with this long period of time we would be able to obtain some interesting patterns and behaviors of these bad domains and IPs. Currently the team is looking inside the System to observe bad domains with fast-flux behaviors.

3.0 Previous Work

Fast-Flux is not a new technology to the internet. Already used by many global companies to provide load-balancing to their services. Quoted from ICANN, Fast-Flux was defined as ‘rapid and repeated changes to host and/or name server resource records, which result in rapidly changing the IP address to which the domain name of an Internet host or name server resolves’ (ICANN, 2008)。In *Behavioral Analysis of*

Fast Flux Service Networks researchers observed some basic patterns of Fast-Flux domains. The observed that these domains does not switch IPs consecutive throughout the day, but can rapidly shift IP in certain time intervals. For name-servers using fast-flux technologies, the interval for the next IP change can be as long as 12-24 hours. Under a 9 month observation, they discovered that BotNets using Fastflux mostly survive only for about one month, phishing BotNets using Fast Flux technology last even as shot as two weeks. (Alper Caglayan, 2009)。These information provided some design suggestions the BDTS.

4.0 Bad Domain Tracking System

4.1 Overall Design

Bad Domain Tracking System currently uses several online sources, honeypots and sandbox to obtain malicious domain names. These Data will then be updated into a domain list database. The program will retrieve a list of domain names for the current session and start querying name servers its corresponding IP addresses. These data are written to another database and stored. Researchers are able to obtain the query results and visualize the results with Gephi. The architecture is shown in figure 1.

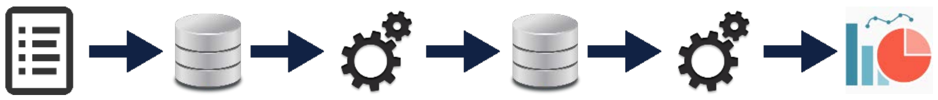
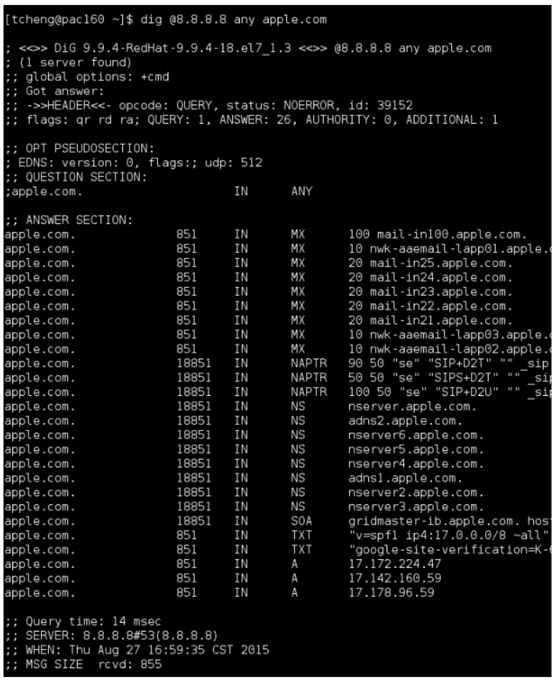


Figure 1



```
[tcheng@pac160 ~]$ dig @8.8.8.8 any apple.com

; <<>> DiG 9.9.4-RedHat-9.9.4-18.el7_1.3 <<>> @8.8.8.8 any apple.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39152
;; flags: qr rd ra; QUERY: 1, ANSWER: 26, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;apple.com.                     IN      ANY

;; ANSWER SECTION:
apple.com.              851     IN      MX      100 mail-in100.apple.com.
apple.com.              851     IN      MX      10 nwk-aaemail-lapp01.apple.
apple.com.              851     IN      MX      20 mail-in25.apple.com.
apple.com.              851     IN      MX      20 mail-in24.apple.com.
apple.com.              851     IN      MX      20 mail-in23.apple.com.
apple.com.              851     IN      MX      20 mail-in22.apple.com.
apple.com.              851     IN      MX      20 mail-in21.apple.com.
apple.com.              851     IN      MX      10 nwk-aaemail-lapp03.apple.
apple.com.              851     IN      MX      10 nwk-aaemail-lapp02.apple.
apple.com.              18851   IN      NAPTR   90 50 "se" "SIP+D2T" "" _sip
apple.com.              18851   IN      NAPTR   50 50 "se" "SIPS+D2T" "" _si
apple.com.              18851   IN      NAPTR   100 50 "se" "SIP+D2U" "" _si
apple.com.              18851   IN      NS      nserver.apple.com.
apple.com.              18851   IN      NS      adns2.apple.com.
apple.com.              18851   IN      NS      nserver6.apple.com.
apple.com.              18851   IN      NS      nserver5.apple.com.
apple.com.              18851   IN      NS      nserver4.apple.com.
apple.com.              18851   IN      NS      adns1.apple.com.
apple.com.              18851   IN      NS      nserver2.apple.com.
apple.com.              18851   IN      NS      nserver3.apple.com.
apple.com.              18851   IN      SOA     gridmaster-ib.apple.com. hos
apple.com.              851     IN      TXT     "v=spf1 ip4:17.0.0.0/8 ~all"
apple.com.              851     IN      TXT     "google-site-verification=K-
apple.com.              851     IN      A       17.172.224.47
apple.com.              851     IN      A       17.142.160.59
apple.com.              851     IN      A       17.178.96.59

;; Query time: 14 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Thu Aug 27 16:59:35 CST 2015
;; MSG SIZE  rcvd: 855
```

Figure 2

4.2 Scraping System

The Scraping System is the core of BDTS. Scraping System made use of DiG9.9.4 RedHat to preform Domain Name queries. To ensure the stability of `dig`, we fix the name server to 8.8.8.8: each query would be sent with a format of: `dig any @8.8.8.8 domain_name`。The results are snapshot as figure 2. These data are categorized into A record and non-A records. A records save only the IPv4 addresses into the database. Other related information related to the domain name such as mail server, IPv6 records and many others are saved to the latter category.

Several versions of the scraper was introduced to the system. BDTS is currently running version 3.1. This version now include multi-threading ability to enhance performance and automated MySQL insert and querying ability. This was result make use from MySQL-connector library and multithreading libraries from Python. Multithreading has significantly increased the performance to up to 5 times faster.

4.3 Database Design

BDTS uses two databases. One database for managing domain names. This database saves the malicious domains and a white list to filter safe sites accidently captured inside honeypots. The view was created to filter white domains from black and return a list to the BDTS scraping system. One important feature of the malicious domain list is it recorded an item called “next query time”. This data is important for the scraping system to identify whether this domain should be listed in the next query session. The other database is for storing query records and its results. This structure can be visualized in Figure 3 and Figure 4.

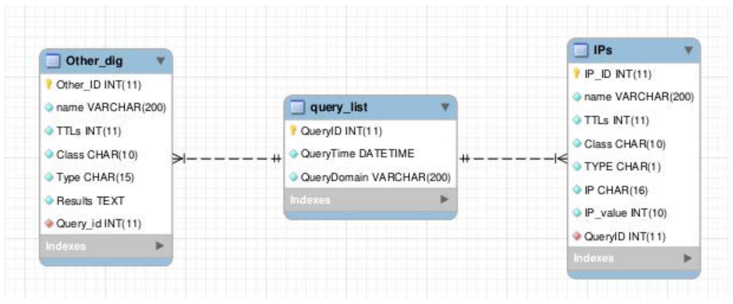


Figure 3

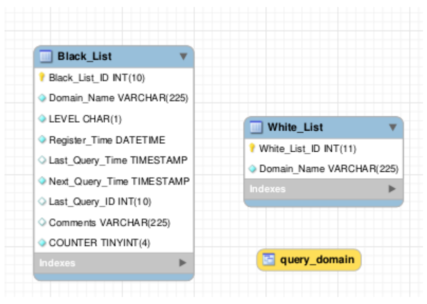
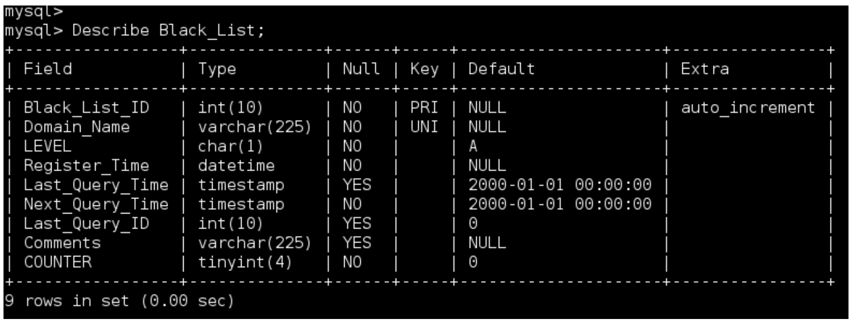


Figure 4

4.4 intelligent exit mechanism

Since BDTS is connected to a growing database of domain names, it is important to assure that there should not be unnecessary query made. Thus, the design of exit mechanism was born. When a new record was inserted inside the database, it has a default query interval of 5 minutes, meaning that the minimal waiting time before the next query of this domain name would be five minutes. If for three consecutive time without a legit return of IP for this particular domain, it will advance to the next level where the time interval for next query is increased. There are total of 20 intervals which the final one was set to query the domain every month. Once a domain was detected a valid corresponding IP, it would return to the initial state of a query at most every 5 minutes. By having this mechanism prevents the BDTS to do extra unnecessary work. This mechanism is combined with the database schema storing malicious domains. The table data is shown as Figure 5.



```
mysql>
mysql> Describe Black_List;
+-----------------+--------------+------+-----+---------------------+----------------+
| Field           | Type         | Null | Key | Default             | Extra          |
+-----------------+--------------+------+-----+---------------------+----------------+
| Black_List_ID   | int(10)      | NO   | PRI | NULL                | auto_increment |
| Domain_Name     | varchar(225) | NO   | UNI | NULL                |                |
| LEVEL           | char(1)      | NO   |     | A                   |                |
| Register_Time   | datetime     | NO   |     | NULL                |                |
| Last_Query_Time | timestamp    | YES  |     | 2000-01-01 00:00:00 |                |
| Next_Query_Time | timestamp    | NO   |     | 2000-01-01 00:00:00 |                |
| Last_Query_ID   | int(10)      | YES  |     | 0                   |                |
| Comments        | varchar(225) | YES  |     | NULL                |                |
| COUNTER         | tinyint(4)   | NO   |     | 0                   |                |
+-----------------+--------------+------+-----+---------------------+----------------+
9 rows in set (0.00 sec)
```

Figure 5

4.5 Visualization

BDTS make use of Gephi to visualize the relationship between Domain and IPs. To clarify the relationship, BDTS provide a preprocess program that filters one to one relationships. The visualization is also able to include MD5 data from Honeynet Projects. Examples are shown in Figure 5, 6, 7 and 8.

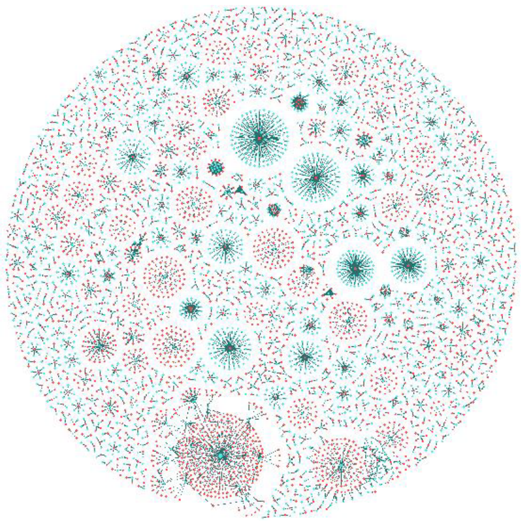


Figure 5

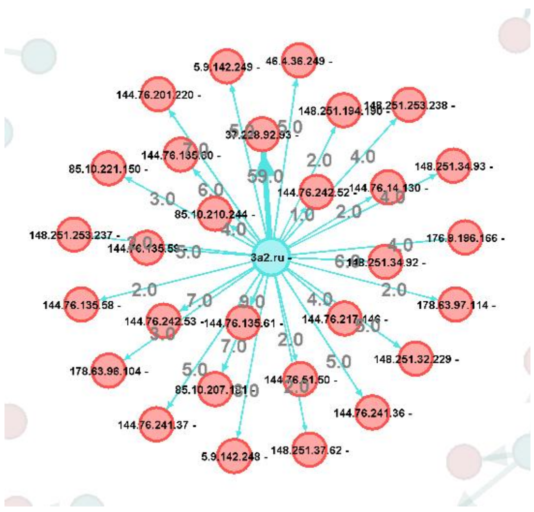


Figure 6

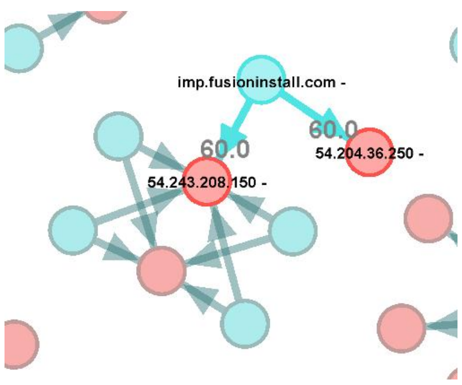


Figure 7

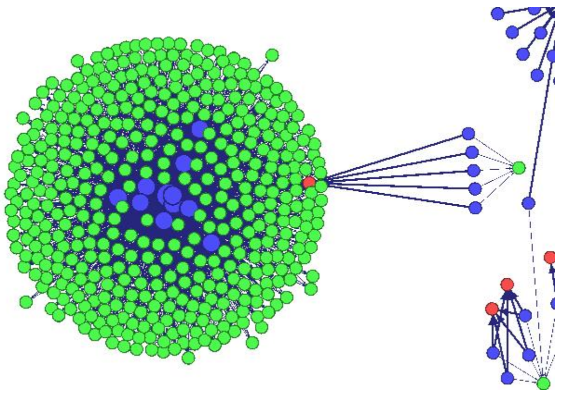


Figure 8

5.0 Further work

With time constraints, some further work can be done to improve the current BDTS running version 3.1. With the use of multi-threading, logging information has to be redesigned and saved for better further use. Second of all, a more automatic visualization tool should be appended with the BDTS. Current visualization still require some coding by hand, mainly because Gephi was written in Java and that it has several restrictions. Finally, it might be possible to rewrite the whois open source project currently in PHP so that there could be some form of querying the IP information.

One of the challenging facing would be the growth of data. Even with the help of intelligent exit mechanism, the data records are growing with a rate of more than 100,000 new records per day. Current MySQL database might not be able to handle billions of data in the far future. This should be an issue to look into in the near future.

6.0 Conclusion and Thanks

Even without deep analysis, the team has identified some interesting relationships of bad domains and its IP just throughout the development of BDTS. We believe there could be potential patterns leading to interesting results with such a long period of mining time. The team truly hopes that Bad Domain Tracking System can come in handy for researchers researching in areas of internet securities and benefit the academic community even with challenges ahead.

Special thanks to the help of National Center for High-performance Computing (NCHC) and the whole Network and Information Security Team for the data, technology and professional support in helping the completion of Bad Domain Tracking System. Finally thanks to team leading research fellow Tsai, Yi-Lang, Associate Research Fellow Prof. Ann, Hsing-Yen and team member Lee, Bo-Yi.

7.0 References

1. Alper Caglayan, Mike Toothaker, Dan Drapaeau, Dustin Burke, and Gerry Eaton. 2009. Behavioral analysis of fast flux service networks. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies* (CSIIRW '09), Frederick Sheldon, Greg Peterson, Axel Krings, Robert Abercrombie, and Ali Mili (Eds.). ACM, New York, NY, USA, , Article 48 , 4 pages. DOI=10.1145/1558607.1558662 http://doi.acm.org.easyaccess1.lib.cuhk.edu.hk/10.1145/1558607.1558662

2. ICANN. GNSO Issues Report on Fast Flux Hosting, March 2008.