# Open DNS Resolver Activity in Campus Network System

Yoshitsugu Matsubara
Graduate School of Science and Technology
Kumamoto University
2-39-1 Kurokami, Central W.,
Kumamoto-City,
JAPAN, 860-855
matubara@cc.saga-u.ac.jp

Yasuo Musashi and Kenichi Sugitani
Center for Management of Information Technologies
Kumamoto University
2-39-1 Kurokami, Central W.,
Kumamoto-City,
JAPAN, 860-855
{musashi,sugitani}@cc.kumamoto-u.ac.jp

Toshiyuki Moriyama
Faculty of Social and Environmental Studies
Fukuoka Institute of Technology
3-10-1 Wajirohigashi, Higashi-ku
Fukkuoka-City,
JAPAN, 842-0295
moriyama@0disaster.net

*Abstract*— **We statistically investigated the total A-resource record (RR) based DNS query request packet traffic from the campus network system to the top domain DNS server in a university during January 1st to December 31st, 2014. The obtained results are: (1) we found significant query keyword based entropy changes in the total DNS query request traffic at February 5th, 2014. (2) In the total A-RR based DNS query request packet traffic, we observed 73-90% of unique random query keywords including eleven source IP addresses like a Kaminsky-like random query (KLRQ) attack. (3) Also, we found that the source IP addresses were assigned to the home/broadband routers in campus laboratories, as open DNS resolvers. (4) Also, we calculated frequency distribution of the Levenshtein distance between the DNS query keywords and the peaks that were observed at 10-15 per day. Therefore, we can conclude that the Levenshtein distance model is useful for developing a detection model of open DNS resolvers.**

***Keywords-Open DNS resolver; Open Resolver; DNS Log Analysis, Kaminsky attack;***

## I. INTRODUCTION

Recently, we observed an interesting entropy increase in the A resource record (RR) based domain name system (DNS) query request packet traffic from the campus network to the top domain DNS (tDNS) server in a university, continuously since February 5th, 2014. The entropy increase means an increase in the DNS query request packet traffic including a lot of unique random query keywords (the DNS unique query request packet access). The similar traffic increase has been also reported in the several Weblog sites [1, 2]. This is probably because the DNS unique query request can perform or induce the DNS amplification distributed denial of service (DDoS) attack or the Kaminsky DNS cache poisoning attack, employing the source IP address spoofing technology [3-6]. Furthermore, the random DNS queries access have a possibility to consume the CPU resources of the DNS servers. From these reasons, it is very important to develop and evaluate the specific detection model in order to prevent and/or mitigate the A RR based DNS unique query request packet access to the DNS servers.

Previously, we reported development and evaluation of the restricted Damerau-Levenshtein [7, 8] distance based detection model for the Kaminsky DNS cache poisoning attack [6, 9], and it can be also available for detecting the DNS unique query request packet access. In this paper, (1) we carried out entrops uniqueness, and restricted Damerau-Levenshtein (edit) distance based analyses of the total A RR based DNS query request packet traffic from the campus network through January 1st to December 31st, 2014, (2) we proposed a detection model of the DNS unique query request packet traffic, hybridizing the edit distance and the uniqueness models, and (3) we assessed the proposed detection model.

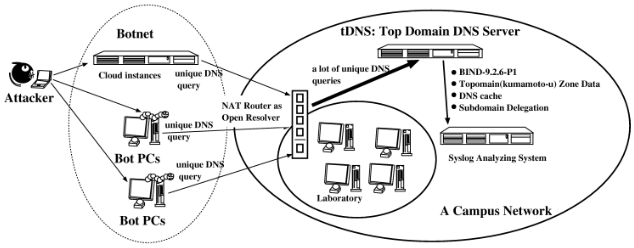


Figure 1. An observed network in the present study.

## II. OBSERVATION

### A. *Network Systems and DNS Query Packet Capturing*

We investigated on the DNS query request packet traffic between the top domain DNS (tDNS) server and the DNS clients. Figure 1 shows an observed network system in the present study, which consists of the tDNS server, the home/routers and the PC clients in laboratories, and the bots like DDoS bots in the campus or cloud instances. The tDNS server is one of the top level domain name (kumamoto-u) system servers and its operating system is Linux OS (CentOS 7.0) in which the kernel-3.10.0 is currently employed with the Intel Xeon E5-2430 v2 2.50 GHz 6 Cores quad node system, 16GB core memory, and Emulex Corporation OneConnect 10GbE NIC (be3) Ethernet Controller. In the tDNS server, the BIND-9.9.4 program package has been employed as a DNS server daemon [10]. The DNS query request packets and their query keywords have been captured and decoded by a query logging option in the BIND program. The log of the DNS query request packet access has been recorded in the syslog files. The line of syslog message consists of the contents in the DNS query request packet like a time, a source IP address of the DNS

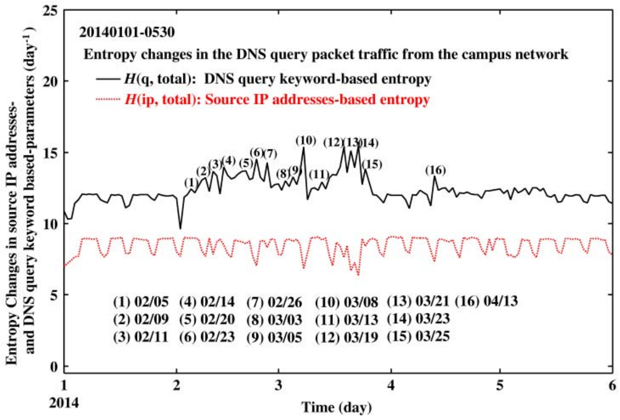


Figure 2. Entropy changes in the total A resource records (RR) based DNS query request packet traffic from the Internet to the top domain DNS (tDNS) server through January 1st to May 30th, 2014. The solid and dotted lines show unique DNS query keywords and the unique source IP addresses based entropies, respectively ($day^{-1}$ unit).

client, a query keyword, a type of resource record (A, PTR, MX, etc).

### *B. Estimation of DNS Query Traffic Entropy*

We employed Shannon's function in order to calculate entropy value H(X), as

$$H(X) = -\sum_{i \in X} P(i)\log_2 P(i) \tag{1}$$

where X is the data set of the frequency freq(j) of a unique IP address or that of a unique DNS query keyword in the DNS query request packet, and the probability P(i) is defined, as

$$P(i) = freq(i)/(\sum_{j} freq(j)) \tag{2}$$

where i and j (i, j ∈ X) represent the unique source IP address or the unique DNS query keyword in the DNS query request packet, and the frequency freq(i) are estimated with the script program, as reported in our previous work [12].

### *C. Entropy Changes in the A RR based DNS Query Traffic*

Firstly, we demonstrate the calculated source IP address- and the query keyword based-entropies for the total A resource record (RR) based DNS query request packet traffic from the campus network to the top DNS (tDNS) server through January 1st to May 30th, 2014, as shown in Figure 2.

In Figure 2, we can observe that the both entropy curves change in a mild manner (a source IP address based entropy value of 8.9 $day^{-1}$ and a query keyword based entropy value of 11.8 $day^{-1}$). However, we can see that the DNS query keyword based entropy value drastically changes (to 12.3 $day^{-1}$) after February 5th, 2014. Recently, Coleman et al. also reported the similar A request resource (RR) based DNS unique query request packet traffic [1, 2], including a lot of unique DNS query keywords, which will be discussed in the next subsection.

### *D. Frequency Distribution of Source IP addresses and Query Keyword Uniqueness*

We also calculated frequency distribution of each source IP address with a uniqueness rate of its query keywords in the A RR based DNS query request packet traffic through February 5th, 2014, and the results are shown Table 1.

Table 1. Frequency distributions of source IP addresses in the total A RR based DNS query request packet traffic and uniqueness rates of their query keywords at February 5th, 2014 ($day^{-1}$).

| No. | IP address | Frequency ($day^{-1}$) | Uniqueness Rate of Queries (%) |
|---|---|---|---|
| 1 | 133.95.a1.a2 | 20,763 | 88 |
| 2 | 133.95.b1.b2 | 17,362 | 73 |
| 3 | 133.95.c1.c2 | 16,812 | 90 |
| 4 | 133.95.c1.c3 | 16,754 | 90 |
| 5 | 133.95.d1.d2 | 13,296 | 80 |
| 6 | 133.95.e1.e2 | 13,198 | 90 |
| 7 | 133.95.c1.c4 | 13,048 | 83 |
| 8 | 133.95.a1.a3 | 12,853 | 77 |
| 9 | 133.95.f1.f2 | 12,602 | 86 |
| 10 | 133.95.b1.b3 | 12,384 | 84 |
| 11 | 133.95.g1.g2 | 11,004 | 86 |

In Table 1, we can observe the top eleven source IP addresses, in which the frequencies take more than 10,000 $day^{-1}$, and their uniqueness rates of DNS query keywords do round 73%-90%. Fortunately, we were able to find out the top eleven IP hosts that were home routers in laboratories in the campus.

Further, we investigated the query keyword change in the A RR based DNS query request packet traffic through February 5th, 2014, and the results are shown in Figure 3.

```
Feb 5 11:46:02 kun named[13758]: client 133.95.a1.a2#7890: query: rtilzib.aa.cp375.com IN A
Feb 5 11:46:03 kun named[13758]: client 133.95.a1.a2#7891: query: lngfzdhulhd.tt.63fy.com IN A
Feb 5 11:46:03 kun named[13758]: client 133.95.a1.a2#7892: query: afoaypibbks.tt.63fy.com IN A
Feb 5 11:46:03 kun named[13758]: client 133.95.a1.a2#7893: query: osinzhfhuxt.tt.63fy.com IN A
Feb 5 11:46:03 kun named[13758]: client 133.95.a1.a2#7894: query: whkhb.aa.cp375.com IN A
Feb 5 11:46:04 kun named[13758]: client 133.95.a1.a2#7895: query: kbfidynucdqjozo.aa.cp375.com IN A
Feb 5 11:46:05 kun named[13758]: client 133.95.a1.a2#7896: query: nbcdrsghiwxlz.aa.cp375.com IN A
Feb 5 11:46:05 kun named[13758]: client 133.95.a1.a2#7897: query: b.aa.cp375.com IN A
Feb 5 11:46:05 kun named[13758]: client 133.95.a1.a2#7898: query: zpqyaruhezv.www.cc176.com IN A
Feb 5 11:46:06 kun named[13758]: client 133.95.a1.a2#7899: query: msuyckakzguswxu.aa.cp375.com IN A
Feb 5 11:46:08 kun named[13758]: client 133.95.a1.a2#7900: query: bzfvcat.aa.cp375.com IN A
Feb 5 11:46:09 kun named[13758]: client 133.95.a1.a2#7901: query: szgyksrbb.aa.cp375.com IN A
Feb 5 11:46:10 kun named[13758]: client 133.95.a1.a2#7902: query: vlfjbplnxid.tt.63fy.com IN A
Feb 5 11:46:11 kun named[13758]: client 133.95.a1.a2#7903: query: bfm.aa.cp375.com IN A
Feb 5 11:46:11 kun named[13758]: client 133.95.a1.a2#7904: query: otfnp.aa.cp375.com IN A
Feb 5 11:46:13 kun named[13758]: client 133.95.a1.a2#7905: query: ltn.aa.cp375.com IN A
```

Figure 3. Changes in the log messages A resource record based DNS request packet from the source IP address of 133.95.a1.a2.

In Figure 3, we can observe a continuously repeated sequence of the unique query keywords and this feature apparently differs from that previously reported [9] i.e. the uniqueness of query keywords becomes more complicated. Usually, these features can be observed in the convensional Kaminsky attack, as well as the DNS server simultaneously receives a lot of fake DNS query reply packets. However, we could not observe the DNS query replies in the DNS queries in February 5th, 2014. Hereafter, let us call it as a Kaminsky-like random query (KLRQ) attack activity.

Therefore, it is required to develop a new detection model for the KLRQ attack.

*E. Detection Model for KL-Random Query Attack*

We define here a detection model of the A RR based DNS unique random query request packet access (KLRQ attack).

— A detection model — it can be mainly carried out by a small network address range of IP hosts in the campus network. Since these IP hosts send a lot of the A RR based DNS query request packets to the tDNS server, the traffic can be detected by calculating the Euclidian distance between the source IP addresses. Then, we suggest hereafter the restricted Damerau-Levenshtein (edit) distance [7, 8] based detection system of the KLRQ attack, since the new attack causes the continuously repeated sequence of the random query keyword (Figure 3).

Here, we should also define thresholds for detecting the new attack activity, as setting to 10 packets $day^{-1}$ for the frequencies of the top unique source IP addresses and for the edit distance, respectively.

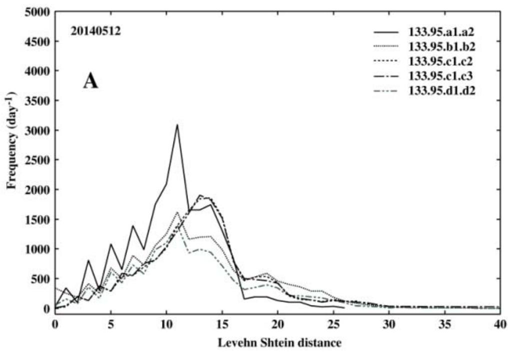

Figure 4. Frequency distributions of the source IP addresses.

*F. Euclidean-Distance of Source IP addresses*

The Euclidean distances, ed($sIP_i$, $sIP_{i-1}$), are calculated, as

$$ed(sIP_i, sIP_{i-1}) = \sqrt{\sum_{j=1}^{4} (x_{i,j} - x_{i-1,j})^2} \qquad (3)$$

where both $IP_i$ and $IP_{i-1}$ are the current source IP address i and the last source IP address i-1 respectively, and where $x_{i,1}$, $x_{i,2}$, $x_{i,3}$, and $x_{i,4}$ correspond to an IPv4 address like A.B.C.D, respectively.

If the KLRQ attack activity model follows a single or distributed source IP address based model i.e. we define the KLRQ activity, the detection is decided by thresholds as ed($sIP_i$, $sIP_{i-1}$)=0 or 1.0≤ed($sIP_i$, $sIP_{i-1}$)≤ 5.0.

*G. Estimation of restricted Damerau-Levehnshtein (Edit) Distance*

The Levenshtein distance, LD (X, Y), is calculated, as

$$LD[x, y] = \min(LD[x-1][y]+1, LD[x][y-1]+1, LD[x-1][y-1]+cost) \qquad (4)$$

where both x and y are lengths of the strings X and Y, and the X and the Y are strings of the current domain name (DN) i and the last DN i-1 of the DNS query keywords, respectively. We show the frequency distribution of Levehnshtein distance in Figure 4.

In Figure 4, we can see major peaks between 10 and 15. Therefore, the detection of the KLRQ attack activity is decided by thresholds as 10≤LD($DN_i$, $DN_{i-1}$)≤15.

```
1 #!/bin/sh
2 TH=10
3 TH2=5000
4 TH3=70
5 # Step 1  Extracting the A RR based DNS Queries
6 cat /var/log/querylog | clgrep -cclients.conf | \
7 grep "IN A +" > tmpfile1
8 # Step 2 Calculating Levenshtein distance and
9 # frequency distribution of source IP address
10 cat tmpfile1 | \
11 sdis 0.0 0.0 1.0 5.0 | \
12 levens -i 10 15 | tr '#' ' ' | \
13 awk '{print $7}' | sort -r | uniq -c | sort -r | \
14 awk '{printf("%s\t%s\n",$2,$1);}' | \
15 qdos $TH >tmpfile2
16 # Step 3 Calculating the rate of unique DNS queries
17 cat tmpfile1 | clgrep -ctmpfile2 >tmpfile3
18 cat tmpfile2 | qdos $TH2 | awk '{print $1}' >tmpfile4
19 UIPLIST='cat tmpfile4 | awk '{print $1}''
20 for ip in $UIPLIST
21 do
22   nq='cat tmpfile3 | clgrep $ip | wc -l'
23   nuq='cat tmpfile3 | clgrep  $ip | awk '{print $9}' | \
24   sort -r | uniq -c | wc -l'
25   urate='echo $nuq" "$nq | \
26   awk '{printf("%d",$1/$2*100+0.5);}''
27   echo "$ip" "$urate  | \
28   awk '{printf("%15s %15s\n",$1,$2);}' >>tmpfile5
29 done
30 # Scoring the detection of Open Reolver
31 cat tmpfile5 | qdos $TH3 >tmpfile6
32 cat tmpfile3 | clgrep -ctmpfile6 | wc -l  >>ORScore.txt
33 exit 0
```

Figure 5. New Kaminsky Attack Detection Algorithm.

*H. Detection Algorithm for KLRQ Activity*

We suggest the following detection algorithm of the new Kaminsky DNS cache poisoning attack activity and we show a prototype program (see Figure 5):

— **Step 1** *Extracting the A RR based DNS Queries* —In this step, the **clgrep** and **grep** commands extract the A RR based DNS query request packet messages from the DNS query log file (*/var/log/querylog*) and write into the *tmpfile1*.

— **Step 2** *Calculating the Levenshtein distance and frequency distribution of source IP address* —In the step, the **sdis** command prints out a syslog message if the Euclidean distance of two source IP addresses is calculated to be zero or to take a range of 1.0-5.0 [11], the **dleven** command prints out the syslog message if the restricted Damerau-Levenshtein distance LD($DN_i$, $DN_{i-1}$) takes a range of 10-15 and the other commands (lines 11 to 15 in Figure 5) compute and check the frequencies of the restricted Damerau-Levenshtein distance LD($DN_i$, $DN_{i-1}$) and if the frequency exceeds a threshold value (TH=10), they write out the candidate IP addresses into a *tmpfile2* as training data.

— **Step 3** *Calculating the rate of unique DNS queries* —In the step, the **clgrep** commands extract the related messages in the total A RR based DNS query log file (*tmpfile1*), using the training data (*tmpfile2*) and they generate only a new

Kaminsky attack activity related DNS query log file (*tmpfile3*,) the next **qdos** command picks up the source IP addresses if the frequency exceeds a threshold value (TH2=5000) and write it to the temporary file (*tmpfile4*), the **awk**, **echo**, and **clgrep** commands calculate the uniqueness rate of the DNS query keywords for each source IP address, with using the source IP addresses in *tmpfile4*, and write the uniqueness rates into the temporary file (*tmpfile5*).

— **Step 4** *Scoring* —In the final step, if the uniqueness rate of the DNS query keywords, the **qdos** command prints out the source IP addresses into the temporary file (*tmpfile6*), the **wc** command calculates the score for the detection of the new Kaminsky attack activity in the file *tmpfile6*, and it writes out the detection score into a score file (*ORScore.txt*) in an appending manner.

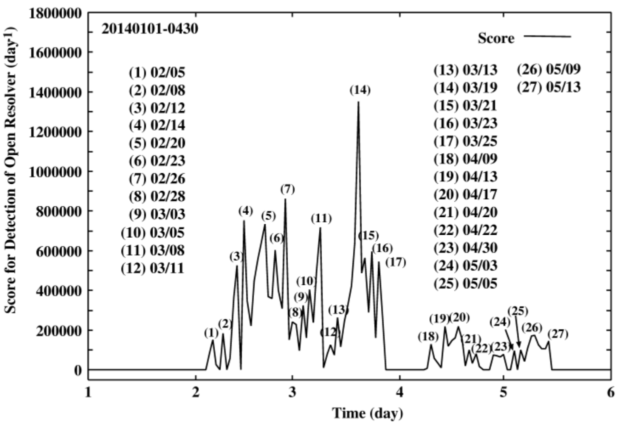


Figure 6. Changes in score of the new Kaminsky attack activity in the total A resource records (RR) based DNS query request packet traffic from the campus network to the top DNS (tDNS) server through January 1st to May 30th, 2014 ($day^{-1}$ unit).

## III. Results and Discussion

### A. Score of New Kaminsky Attack Activity

We illustrate the calculated score of the KLRQ attack activity using restricted Damerau-Lehvenshtein distance based detection model ($10 \le LD(DN_i, DN_{i-1}) \le 15$) between the current domain name $DN_i$ and the last domain name $DN_{i-1}$, as the DNS query keywords in the A RR based DNS query request packet traffic from the campus network to the top DNS (tDNS) server through January 1st, 2014 to April 30th, 2014, as shown in Figure 6.

In Figure 5, we can observe the twenty seven significant peaks (1)-(23), however, we can only sixteen peaks in Figure 2. This feature indicates that the developed detection model can be useful for detecting the KLRQ attack activity

## IV. Conclusions

We developed and evaluated the restricted Damerau-Levenshtein edit distance based detection model of the Kaminsky-like random query (KLRQ) attack activity in the total A RR based DNS request packet traffic from the campus network during January 1st to December 31st, 2014. Interestingly, we observed the twenty three significant peaks in the detection score of the developed detection model for the new KLRQ attack activity in the total A RR based DNS query request packet traffic from the open DNS resolvers in the campus and (2) we also found that the hybridization of edit distance and the uniqueness rate of the DNS query keywords for each source IP address can improve the detection rate of it.


## Acknowledgment

This work was supported by Japan Society for the Promotion of Science KAKENHI (Grant-in-Aid for Challenging Exploratory Research) Grant Number 12013489.


## References


[1] L. Colman, “What Does a DNS Amplification DDoS attack look like?,” SpiceCorps of Metro Detroit, Spiceworks Inc., (2014), http://community.spiceworks.com/topic/441721-what-does-a-dns-amplification-ddos-attack-look-like

[2] Smurfmonitor, “DNS Amplification Attacks Observer,” Blogger, Google Inc. (2014), http://dnsamplificationattacks.blogspot.jp/2014/02/authoritative-name-server-attack.html

[3] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, “A Fair Solution to DNS Amplification Attacks,” Proceedings of the Workshop on Digital Forensics and Incident Analysis 2007 (WDFIA2007), Karlovassi, Samos, Greece, pp.38-47 (2007).

[4] M. Prince, “The DDoS That Knocked Spamhaus Offline (And How We Mitigated It),” ClouFlare (2013), http://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho

[5] J. Nazario: DDoS attack evolution; Computer Security Series, Network Security, Vol.2008, No.4, pp.7-10 (2008).

[6] D. Kaminsky: It’s The End of The Cache As We Know it,” 2008, http://kurser.lobner.dk/dDist/DMK_BO2K8.pdf.

[7] V. L. Levenshtein: Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady, Vol. 10, No. 8, pp.707-710 (1966).

[8] F. J. Damerau: A technique for computer detection and correction of spelling errors, Communications of the ACM, Vol. 7, No. 3, pp.171-176 (1964).

[9] Y. Musashi, M. Kumagai, S. Kubota, and K. Sugitani: Detection of Kaminsky DNS Cache Poisoning Attack, Proceedings of the Fourth International Conference on Intelligent Networks and Intelligent Systems (ICINIS 2011), Kunming, China, pp. 121-124 (2011).

[10] BIND-9.4.4: http://www.isc.org/products/BIND/

[11] N.Shibata, Y. Musashi, D. A. Ludeña Romaña, S. Kubota, and K. Sugitani, “Trends in Host Search Attack in DNS Query Request Packet Traffic,” Proceedings of the Fifth International Conference on Intelligent Networks and Intelligent Systems (ICINIS 2012), Tianjin, China, pp.126-129 (2012).

[12] D. A. Ludeña R., S. Kubota, K. Sugitani, Y. Musashi: DNS-based Spam Bots Detection in a University, *International Journal of Intelligent Engineering and Systems*, Vol. 2, No. 3, 2009, pp.11-18.