Detection of Open Resolver Activity in DNS Query Traffic from Campus Network System

YASUO MUSASHI[†1] YOSHITSUGU MATSUBARA[†2] KENICHI SUGITANI[†1] and TOSHIYUKI MORIYAMA[†3]

We statistically investigated the total A-resource record (RR) based DNS query request packet traffic from the campus network 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 query keywords including eleven source IP addresses (i.e. Kaminsky and/or Kaminsky-like 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.

1. 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 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]. Therefore, it is very important to detect 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 of the Kaminsky DNS cache poisoning attack [6] in the total inbound A RR based DNS query request packet traffic to the campus tDNS server through January 1st to December 31st, 2010 [9], and it can be also useful for detecting the A RR based DNS unique query request packet access. In this paper, (1) we carried out entropy, uniqueness, and restricted Damerau-Levenshtein (edit) distance based analyses of the total A resource record (RR) based DNS query request packet traffic from the campus network through January 1st to December 31st, 2014, (2) we suggested a detection model of the DNS unique query request packet traffic, hybridizing the edit distance and the uniqueness models, and (3) we assessed the detection model.

2. Observation

2.1 Network Systems and DNS Query Packet Capturing

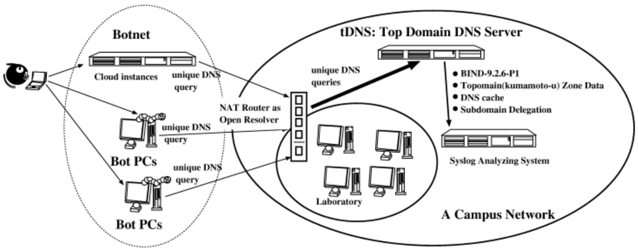
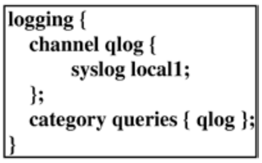


Figure 1 A schematic diagram of an observed network in the present study.



```
logging {
   channel qlog {
         syslog local1;
   };
   category queries { qlog };
}
```

Figure 2 BIND Query logging option in the /etc/named.conf BIND configuration file.

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 and/or broadband 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 it plays an important role of domain name resolution including DNS cache function, and subdomain name delegation services for many PC clients and the subdomain network servers, respectively, and its operating system is Linux OS (CentOS 6.4 Final) in which the kernel-2.6.32 is currently employed with the Intel Xeon X5660 2.8 GHz 6 Cores dual node system, 16GB core memory, and Intel Corporation EthernetPro 82575EB Gigabit Ethernet Controller.

In the tDNS server, the BIND-9.8.2 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 (see Figure 2 and the named.conf manual of the BIND program in more detail). The log of DNS query request packet access has been recorded in the syslog files. All of the syslog files are daily updated by the cron system. 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 client, a query keyword, a type of

†1 Center for Multimedia and Information Technologies, Kumamoto University
†2 Graduate School of Science and Technologies, Kumamoto University
†3 Faculty of Socio-Environmental Studies, Fukuoka Institute of Technology

resource record (A, AAAA, ANY, PTR, MX, or TXT).

2.2 Estimation of DNS Query Traffic Entropy

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

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

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 traffic from the campus network, and the probability P(i) is defined, as

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

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

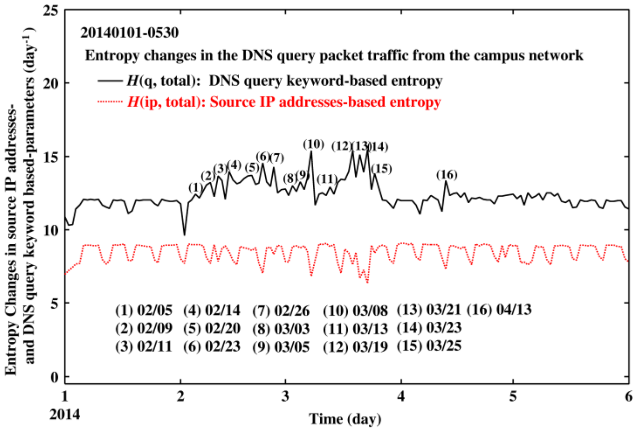


Figure 3 Entropy changes in the total A resource records (RR) based DNS query request packet traffic from the campus network to the top domain DNS (tDNS) server through January 1st to May 30th, 2014. The solid and dotted lines show unique DNS query keywords and the unique source IP addresses based entropies, respectively (day^{-1} unit).

2.3 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, in Figure 3.

In Figure 3, 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. Coleman et al. also reported the similar A RR based DNS unique query request packet traffic [1, 2].

2.4 Frequency Distribution of Source IP addresses and Query Keyword Uniqueness

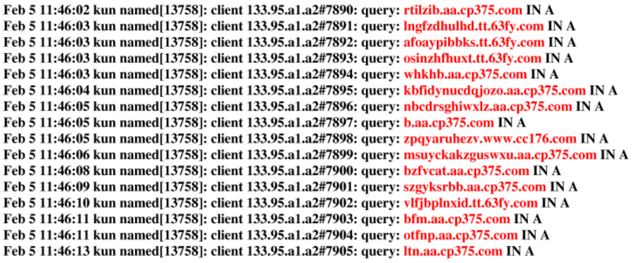
We also calculated frequency distribution of each source IP address with a uniqueness rate of its query keywords in the A RR based DNS query request packet traffic through February 5th, 2014, and the results are shown Table 1.

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 4.



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

In Figure 4, 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 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 new Kaminsky attack or a Kaminsky-like (KL) attack activity.

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

2.5 Detection Model for Kaminsky-Like Attack

We define here a detection model of the Kaminsky-like (KL) 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 Kaminsky-like (KL) attack, since the KL attack causes the continuously repeated sequence of the random query keyword (Figure 4).

Here, we should also define thresholds for detecting the KL 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.

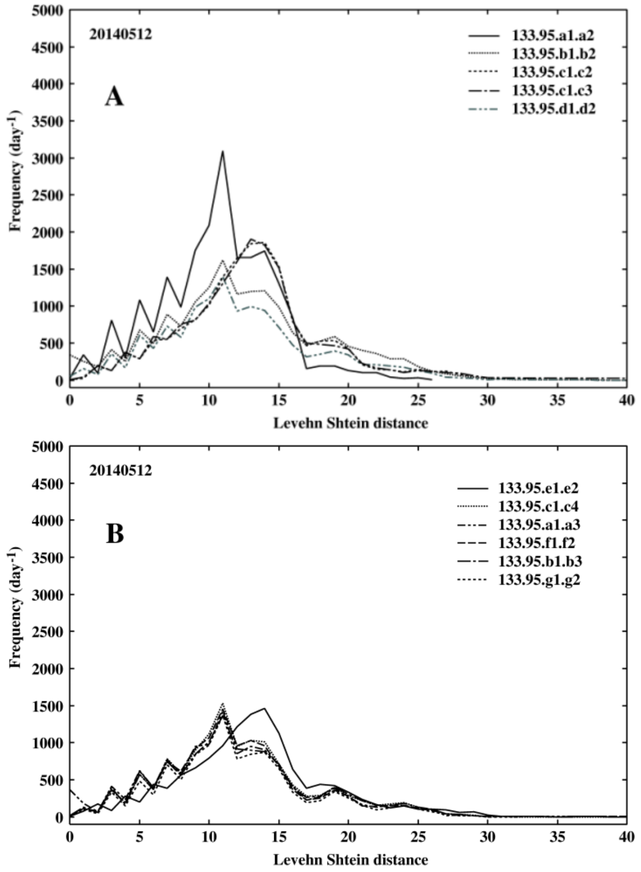


Figure 5 Frequency distributions of the source IP addresses.

2.6 Euclidean-Distance of source IP addresses

The Euclidean distances, **ed(sIP$_i$, sIP$_{i-1}$)**, are calculated, as

$$\mathrm{ed}(\mathrm{sIP}_i, \mathrm{sIP}_{i-1}) = \sqrt{\sum_{j=1}^{4} (x_{i,j} - x_{i-1,j})^2} \qquad (3)$$

where both sIP$_i$ and sIP$_{i-1}$ 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. For instance, if an sIP address is 192.168.1.1, the vector $(x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4})^T$ can be represented as $(192.0, 168.0, 1.0, 1.0)^T$.

If the Kaminsky-like (KL) attack activity model follows a single or distributed source IP address based model i.e. we define the KL attack activity, the detection is decided by thresholds as

$$\mathrm{ed}(\mathrm{sIP}_i, \mathrm{sIP}_{i-1}) = 0.0$$
$$1.0 \le \mathrm{ed}(\mathrm{sIP}_i, \mathrm{sIP}_{i-1}) \le 5.0 \qquad (4)$$

where the thresholds were previously reported in [10].

2.7 Esimation of restricted Damerau-Levehnshtein Distance between DNS Query Keywords

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

$$\mathrm{LD}[x, y] = \min(\mathrm{LD}[x-1][y]+1, \mathrm{LD}[x][y-1]+1,$$
$$\mathrm{LD}[x-1][y-1]+\mathrm{cost}) \qquad (5)$$

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. For instance, if the DNs are X = “a001.example.com” and Y = “a002.example.com”, the Levenshtein distance LD (X,Y) is calculated to be 1, since the Levenshtein distance counts the number of edit operations like “insertion,” “deletion,” and “substitution” [6]. Furthermore, the restricted Damerau-Levenshtein distance takes into consideration the operation “transposition” in order to suppress the overestimation [7].

In Figure 5, we can see major peaks between 10 and 15. Therefore, the detection of the Kaminsky-like attack activity is decided by thresholds as

$$10 \le \mathrm{LD}(\mathrm{DN}_i, \mathrm{DN}_{i-1}) \le 15 \qquad (6)$$

2.8 Detection Algorithm for Kaminsky-like Attack Activity

We suggest the following detection algorithm of the Kaminsky-like (KL) attack activity and we show a prototype program (see Figure 6):

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

Figure 6 New Kaminsky Attack Detection Algorithm.

— **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 6) 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 extracts the related messages in the total A RR based DNS query log file (*tmpfile1*), using the training data (*tmpfile2*) and they generate only a Kaminsky-like (KL) 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 exceeds a threshold value (TH3=70), 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 KL attack activity in the file *tmpfile6*, and it writes out the detection score into a score file (*ORScore.txt*) in an appending manner.

Note that in the above script, we blend the Damerau-Levenshtein distance model and the uniqueness rate of the DNS query keywords for each source IP address, in order to suppress the false positive and to save the calculation time. This is because the DNS queries request unique keywords in the KL attack packets

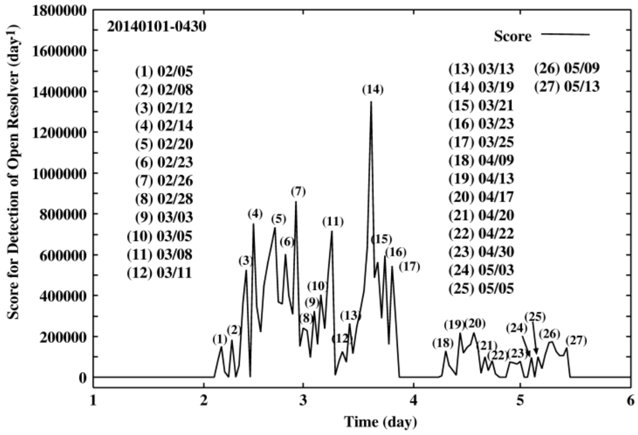


Figure 7 Changes in score of the new Kaminsky attack activity in the total A resource records (RR) based DNS query request packet traffic from the campus network to the top DNS (tDNS) server through January 1st to May 30th, 2014 (day^{-1} unit).

3. Results and Discussion

3.1 Score of Kaminsky-like Attack Activity

We illustrate the calculated score of the Kaminsky-like (KL) attack activity using restricted Damerau-Lehvenshtein distance based detection model ($10 \leq$ LD(DN_i, DN_{i-1}) ≤ 15) between the current domain name DN_i and the last domain name DN_{i-1}, as the DNS query keywords in the A resource record (RR) based DNS query request packet traffic from the Internet to the top DNS (tDNS) server through January 1st to December 31st, 2014, as shown in Figure 7.

In Figure 7, we can observe that the score curve takes a zero value until February 4th, it starts to change drastically after February 5th, 2014, and it terminates in May 13th, 2014. Also, we can observe the twenty seven significant peaks (1)-(27), however, we can only sixteen peaks in Figure 3. This feature indicates that the developed detection model can be useful for detecting the KL attack activity in the A RR based DNS query request packet traffic from the campus network.

3.2 DNS Query Request Traffic to Home Routers

We investigated the DNS query request packet traffic from the Internet to the campus laboratory home routers (133.95.c1.c2, 133.95.b1.b3, and 133.95.f1.f2) through March 26th-27th, 2015. The number of unique source IP addresses is calculated to be 488,435 day^{-1} (488,528 day^{-1}). This feature shows that the Kaminsky-like (KL) attack activity can be carried out with the source IP address spoofing technique [13] and this also means that it is unable to block all the unique source IP addresses in the KL attack activity at the firewall and/or the IPS security appliance. Therefore, we blocked the DNS query request packet traffic from the Internet (ANY:ANY) to the home routers (destination port 53).

4. Conclusions

We developed and evaluated the restricted Damerau-Levenshtein edit distance based detection model of the new Kaminsky-like (KL) attack activity in the total A RR based DNS request packet traffic during January 1st to December 31st, 2014.

The following interesting results are found: (1) we observed the twenty seven significant peaks in the detection score of the developed detection model for the new Kaminsky 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.

Note that the KL attack is currently known as “Water torture” attack as reported on SECURE64 Blog [14].

Acknowledgment All the studies were carried out in Centre for Management and Information Technologies (CMIT) of Kumamoto University. We gratefully thank all the CMIT staffs and all the members of Kumamoto University. This work was supported by Japan Society for the Promotion of Science KAKENHI (Grant-in-Aid for Scientific Research (B)) Grant Number 26282114.

References

1) Colman, L.: 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) *Kambourakis, G., Moschos, T., Geneiatakis, D., and Gritzalis, S.: 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) Prince, M.: Deep Inside a DNS Amplification DDoS Attack, ClouFlare, 2012, http://blog.cloudflare.com/deep-inside-a-dns-amplification-ddos-attack

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

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

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

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

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

10) BIND-9.8.2: http://www.isc.org/products/BIND/

11) Shibata, N., Musashi, Y., Ludeña Romaña, D. A., Kubota, S., and Sugitani, K.: 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) Ludeña Romaña, D. A., Kubota, S., Sugitani, K., and Musashi, Y.: DNS-based Spam Bots Detection in a University, International Journal of Intelligent Engineering and Systems, Vol. 2, No. 3, pp.11-18 (2009).

13) Tanase, M.: IP Spoofing: An Introduction, http://www.symantec.com/connect/articles/ip-spoofing-introduction

14) SECURE64 Blog: Water Torture: A Slow Drip DNS DDoS Attack, https://blog.secure64.com/?p=377