Ports Distribution Management for Privacy Protection Inside Local Domain Name System

Fei Song
SEIE and NGIT
Beijing Jiaotong University,
P. R. China
fsong@bjtu.edu.cn

Wei Quan
SEIE and NGIT
Beijing Jiaotong University,
P. R. China
weiquan@bjtu.edu.cn

Tianming Zhao
SEIE and NGIT
Beijing Jiaotong University,
P. R. China
tmzhao@bjtu.edu.cn

Hongke Zhang
SEIE and NGIT
Beijing Jiaotong University,
P. R. China
hkzhang@bjtu.edu.cn

Ziwei Hu
Global Energy Interconnection
Research Institute,
P. R. China
huziwei@geiri.sgcc.com.cn

Ilsun You
Information Security Engineering
Soonchunhyang University
Republic of Korea
isyou@sch.ac.kr

ABSTRACT
Domain Name System (DNS) had been recognized as an indispensable and fundamental infrastructure of current Internet. However, due to the original design philosophy and easy access principle, one can conveniently wiretap the DNS requests and responses. Such phenomenon is a serious threat for user privacy protection especially when an inside hacking takes place. Motivated by such circumstances, we proposed a ports distribution management solution to relieve the potential information leakage inside local DNS. Users will be able to utilize pre-assigned port numbers instead of default port 53. Selection method of port numbers at the server side and interactive process with corresponding end host are investigated. The necessary implementation steps, including modifications of destination port field, extension option usage, etc., are also discussed. A mathematical model is presented to further evaluate the performance. Both the possible blocking probability and port utilization are illustrated. We expect that this solution will be beneficial not only for the users in security enhancement, but also for the DNS servers in resources optimization.

CCS Concepts
Security and Privacy ~ Network Security

General Terms
Design, Network, Security

Keywords
Domain Name System; Ports Distribution; Resource Management; Privacy Protection

1. INTRODUCTION
In order to keep a balance between human's memory habit and computer's recognition pattern, Domain Name System (DNS) [1]-[3] is introduced as a responsible and straightforward proxy between humans and computers. The Uniform Resource Locators (URLs), which are quite familiar to the users, can be mapped into corresponding IP addresses, which are comprehensible for the machines, before the end hosts want to obtain the online services. Normally, the local DNS servers will store some famous and frequently visited URLs. If there is no matching can be found, by using a hierarchical decentralized structure, these DNS requests will be forwarded to the up level servers. Such iteration should continue until a correct IP address or "Not found" returns. Although this system had been successfully used for decades, there are still many tough issues (such as insider threats [4][5], anomaly behavior [6][7], large scale [8]-[10], distributed denial of service [11]-[13]) needed to be better solved.

From a system perspective, for instance, due to the rapid progress in social network applications, more and more short, irregular and volatile URLs are involved, which definitely increase the burden of DNS during the inquiring process. These huge volume requests generated from smart phones, laptops, pads, etc. further worsen the situation. Since there are only 13 original root servers in current Internet, efficiency, scalability, and reliability of whole DNS should be reconsidered carefully and comprehensively [14]-[16].

From the user perspective, more importantly, privacy is a challenging issue that attracts significant attentions in communities. In order to realize the timeliness during DNS inquiring, the simplification had been regarded as an extremely crucial aspect of the original design philosophy, which also hides some potential threats. One obvious example is that the DNS requests might be wiretapped by other users. It is not complicated for the hackers to execute that because most of DNS is operating on top of User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) with public and fixed port number. If "man-in-the-middle" attack is successfully initiated inside the system, the hackers would be able to spoof the DNS server based on the information they monitored. DNS Private Exchange (dprive) working group [17] had been established in the Internet Engineering Task Force (IETF) [18] just to focus on this emerging area.

Motivated by previous discussions, we aim to design a possible solution for privacy protection during the local DNS inquiring procedures. More specifically, the port assignment and switching management are studied in-depth at the local network range. Port 53 of UDP and TCP is commonly and widely adopted in DNS,

MIST'16, October 28 2016, Vienna, Austria
 ISBN 978-1-4503-4571-2/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2995959.2995965

which bring the convenience and risk simultaneously. In order to achieve such goals, some essential questions need to be carefully answered: How to increase the difficulty of eavesdropping without adding cost? What is the necessary factor during implementation process of new approach? How to analyze the performance of new solution from the mathematical point of view?

The main contributions of this paper could be summarized into two aspects:

(1) A ports distribution management solution is firstly proposed, and the necessary implementation steps together with results are also provided.

(2) A mathematical model is presented and analyzed to quantitatively evaluate the performance.

The organization of this paper is as follow: In Section 2, the frequently-used types of attack are listed and privacy-related issues are emphasized. Then possible location of DNS server is reviewed. The potential influences of information leakage are also analyzed. Section 3 proposed the solution of ports distribution management. After the principle discussion of port number assignments, specific port switching schemes are presented via both end host's and DNS server's perspectives. Then the implementation results are demonstrated for validation purpose. In Section 4, for quantitatively investigating the relationship among different service parameters, one mathematical model is introduced. Then the performance analysis is illustrated and compared. Section 5 summarizes the entire paper at the end.

2. POTENTIAL RISKS IN DNS INQUIRING

There are various types of potential risks when the DNS messages are exchanged between servers and end hosts. In order to comprehensively understand it, we list different types of attack and focus on privacy-related cases. Then possible locations of default DNS server are presented to analyze possible influence of information leakage.

2.1 Possible Types of Attack for DNS

During the DNS request and response process, normally, it is difficult to predict the weak point of a system since there are more than one attack approaches [19]-[22]. For instance, Sniffing attack could grab the data packets during the transmission without changing or interrupting original communication; Injection attack could insert some malicious codes or commands to destroy or affect the ordinary execution; Capture attack could pretend to be the end host to communicate with the remote server; Phishing attack could pretend to be the server and return the fake information back to the end host; Bandwidth occupation attack could generate huge volume of traffic and force them to the server simultaneously. Repetition attack could ask the server to operate the similar task many times. Unfortunately, all these attack forms could happen to DNS. Since we mainly focus on privacy-related issue, the previous three cases (i.e. sniffing, injection and capture) are selected as the primary target.

2.2 Possible Locations of Default DNS Server

Assuming an URL's IP address is a brand new one and had never been buffered in DNS. Then the full resolution steps of these URLs might be as follow: (1) The end host checks its local mapping relationship buffer inside the machine. (2) A new request will be sent from the end host to the default DNS server. (3) Then such request will be forwarded from the default DNS server to the root DNS server. (4) IP address of the first level (or other level) domain name server will be replied to the default DNS server. (5) A new request will be created and sent from the default DNS server to the first level (or other level) domain name server. (6) The iteration of "step (4)" and "step (5)" should be executed until the corresponding IP address of this URL is correctly found. (7) A respond will be returned from the default DNS server back to the end host.

It is clear that the default DNS server is a significant component during the whole process. Based on current practice, the possible locations of it might be: (1) Local Area Network (LAN): the owner of the server might be an institution, a company, or even the user. The Round Trip Time (RTT) between end host and default DNS server is quite small. (2) Internet Service Provider (ISP): multiple servers might be established based on anycast technology. The optimization could be operated to reduce the latency of DNS inquiring. (3) Wide Area Network (WAN): some well known DNS-supported IP address, such as 8.8.8.8, 114.114.114.114, can be easily found online. Normally, they are open to public if their privacy policy is accepted. Passive data collection might be executed for operation, upgrade and research purposes.

The probability of being eavesdropped will be increased if more intermediate network equipments are getting involved between end host and default DNS server. The situations for subsequent inquiries among other DNS servers are also the same. However, this paper only focuses on the first case and attempts to propose effective solution.

2.3 Possible Influence of Information Leakage

When previous privacy-related attacks are enabled between end hosts and default DNS servers, different influences of information leakage might be triggered.

Firstly, as the simplest case, the specific website could be obtained by monitoring the DNS traffic. Some representative tags, such as "sports.163.com", "book.sohu.com", further indicate the users' interest. Moreover, other generic classifications (including the location, native language, personal preference etc.) are also available based on brief analysis.

Secondly, although the information of webpage will not be sent to the DNS server, the exact webpage still might be extracted from the DNS traffic. For example, when a user wants to visit:

example.com/4421184/3847815/5481123,

only the "domain name" part (i.e. "example.com") of the website will be used for DNS inquiring. As soon as the IP address has been received, the user will send the remains of the URL directly to the website. Such process means the monitor on default DNS port can not explicitly capture the information of webpage. However, due to the multiple DNS requests triggered by the inline elements of script language, the "fingerprint of webpage" could be summarized, such as:

www.google-analytics.com,
static.baifendian.com,
api.share.baidu.com.

The purposes of these DNS requests might be statistics of traffic, usage of Application Program Interface (API), etc. They are quite helpful for improving the functionalities of webpage. By identifying the properties (request sequence, time interval, request frequency, etc.) of different "fingerprint", a specific webpage might be located. Then a full URL could be easily obtained.

Thirdly, the privacy also might be challenged even the user did not initiatively visit any websites. When the Internet connection is

ready, the operating systems and applications could automatically generate several DNS requests. For instance, Windows or Linux systems may check "updating" URLs; Firefox browser may check "mozilla" URLs; Applications in smart phones may check "advertisement" URLs. These DNS requests might leak the kernel versions of different software, which will definitely threaten the user privacy.

Fourthly, combination analysis could be utilized in DNS information wiretap. Here we only illustrate two examples:

(1) Identification Recognition: When the visiting pattern of a specific user had been accurately recorded, a usage database can be created based on previous discussions. The identification of this user could be recognized even if he or she had changed original login terminal.

(2) Relationship Mining: Imaging a user had been attempting to visit several typical URLs during a random period, the potential relation among these websites together with user's current state might be achieved. If the URLs contain:

> www.cam.ac.uk, uk.linkedin.com,
> cn.indeed.com, job.alibaba.com,
> www.uber.com/careers, maps.google.com.

In such case, we could infer a reasonable scenario: a student who had finished his or her study in University of Cambridge is looking for a job. The preferred recruit website is "Linkedin at United Kingdom" and "Indeed at China". Two attractive job positions might be provided by "Alibaba" and "Uber". The online map in "Google" is selected for checking the locations or routes.

For reducing the probabilities of privacy leakage during local DNS inquiring procedures, we proposed a ports distribution management solution.

3. PORTS DISTRIBUTION SCHEMES

The preparation work is presented at the beginning of this Section to outline the preliminary of ports distribution. Necessary steps and management solutions are detailed via user's and DNS server's perspective, respectively.

3.1 Preliminary of Port Number Assignments

There are two main strategies to achieve ports distribution: "static mode" and "dynamic mode".

For the first mode, based on the rules of Internet Assigned Numbers Authority (IANA), the traditional DNS service normally open port 53 for receiving TCP and UDP requests, i.e. the user could create two independent connections via TCP 53 port and UDP 53 port, simultaneously. The port number applying can be submitted to IANA. However, it will take a complex process and long time to be approved. The hackers will be also aware of the new assigned port number as soon as they are published. Therefore, such "static mode" is not acceptable for the DNS ports distribution.

For the second mode, port numbers should be temporarily assigned to different users by DNS server when requests are received. These new allocated port numbers must be recycled for subsequent users. According to the regulation of RFC 6335 [23], the port number of transport protocols can be classified into three categories: The system ports (0-1023, assigned by IANA), the user ports (1024-49151, assigned by IANA) and the dynamic ports (49152-65535, never assigned). The system ports and the user ports (0-49151) can be further classified into "Assigned", "Unassigned" and "Reserved". Therefore, it is more convenient to choose "dynamic ports" as the resource pool of port numbers.

3.2 Ports Distribution at the End Host Side

After traditional initialization, end host needs to consider whether a new port number for DNS inquiring is necessary. If so, it should check whether the port number had been assigned already. The sender may also need to verify the expiration period of this port number in some cases. Then a DNS request with new destination port number could be sent if all previous questions are "Yes".

If there is no existing port number, the end host should initiate a "Port Distribution" request. When DNS server has successfully opened the new port for the user, a confirmation will be triggered and sent. If the DNS server does not support such new features or new ports have not been opened yet, the sender could consider other DNS servers or retry again. The final DNS request with new destination port number (with original destination port number) will be generated if port applying is successful (denied).

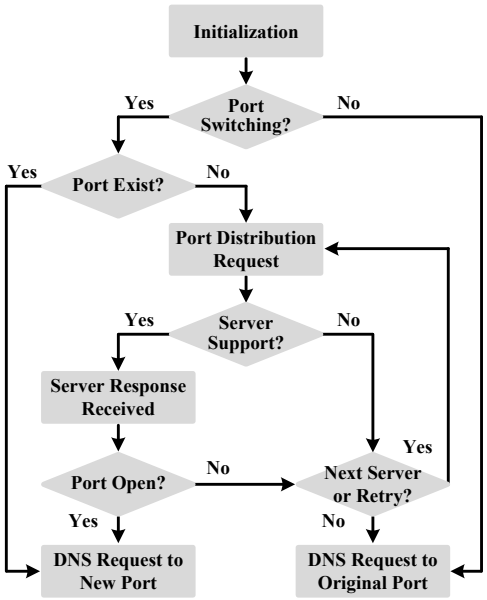


Figure 1 Process flow at the end host side

The whole process flow via end host's perspective can be found in Figure 1.

3.3 Ports Distribution at the DNS Server Side

The procedures at the DNS server side is more complicated. Here we only present new-feature-enabled situation. Port 53 for both TCP and UDP requests should be monitored at all the time. If a "Port Switching" request has been received, the opened port number for this specific user should be checked carefully. If it is still valid, a success message should be returned back to the user. Then necessary notification for firewall must be made at the DNS server's side to guarantee subsequent requests can be received successfully.

If the new port had not been opened yet or expired already, a decision of "Open New" port number should be made by the DNS server. If the answer is "Yes", one or more ports from 49152 to 65535 could be selected based on any customized algorithms. When all the usable port numbers are occupied, the DNS server could also compulsively recycle some "old" or "low priority" port numbers. Then the monitoring threads will be triggered to take care of these new port numbers. Relevant steps for notifying fire-

wall and user should be executed in sequence. If no port number could be retrieved or DNS server does not want to open a new port number, a "N/A" message should be returned back to the user.

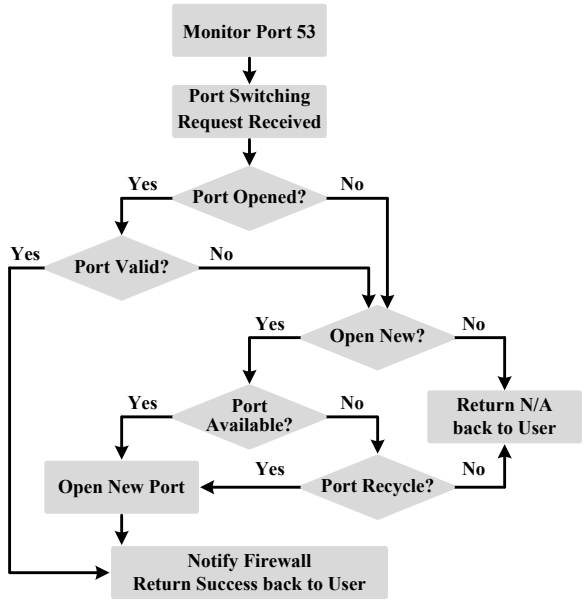


Figure 2 Process flow at the DNS server side

The whole process flow via DNS server's perspective is demonstrated in Figure 2.

We emphasize that these previous procedures could be integrated to Domain Name System Security Extensions (DNSSEC) [24][25] or other encryption protocols. Some representative considerations and updates can be found in [26][27].

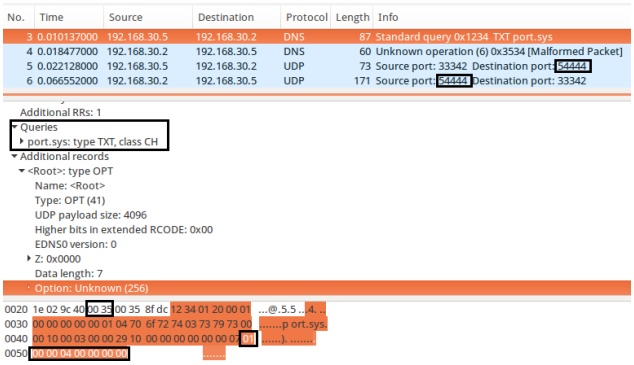


Figure 3 DNS packets exchanging for applying port switching

3.4 Implementation and Verification

There are several handy tools (Unbound, BIND, PowerDNS, etc.) can be used to establish DNS server. We selected the Unbound as the original protocol stack model during the modification. Wireshark is used to capture and analyze the packet exchanging. C and Python can be adopted to achieve our functionality.

Firstly, we generate ordinary data packet and selectively change part of it during the DNS inquiring process. A demons program is attempting to modify destination port number whenever a DNS request is captured. Such function will not only enable the equipment to achieve the port adjustment, but also provide a mechanism for content replacement. The other fields inside DNS request will be untouched.

Secondly, the extension for port switching is implemented based on the packet format proposed in RFC6891 [28] The motivation of Extension Mechanisms for DNS (EDNS) is to release the original limitation in DNS packet modifications. By adding the specific information into additional field, the sender could negotiate with DNS server. Figure 3 illustrate the procedures of port switching. The IP address of end host and DNS server are respectively assigned to 192.168.30.5 and 192.168.30.2. Packet No. 3 (the line highlighted with orange color) shows that the original DNS request is still sent to default port. "00 35" is the hexadecimal of 53. The rest part of this packet will attempt to apply for a new port number. Packet No.4 is the response sent by DNS server. Since there is no Wireshark plug-in to support our new packet, "Malformed Packet" is marked to identify this data format which can not be correctly recognized. However, it will not affect the inquiring at the end host. Packet No.5 is the new DNS request whose source port and destination port are filled with 33342 and 54444. Finally, the corresponding response, i.e. packet No.6, is returned back to the end host with the inquiring result. "Protocol" field of packet No.5 and No.6 are marked with "UDP" instead "DNS". The reason might be Wireshark identifies the type of protocol based on source and destination port number. Since both 33342 and 54444 are not recorded as DNS, only UDP will be shown at "Protocol" field.

4. MATHEMATICAL ANALYSIS

In previous section, basic process at both end host side and DNS server side had been presented, discussed and implemented. However, there are still some inevitable problems which directly affect the performance of this new scheme, such as "When is the suitable timing to recycle these occupied ports?", "How long should an end host possess one port", "What is the reasonable quantity when setting the port number pool?". We believe that the answers are not unified and should be adjusted based on the actual situation. In order to better understand potential relationship among these parameters and reduce the privacy risks during the DNS inquiring process, we present a mathematical model to analyze the system performance.

4.1 Modeling of Port Switching Process

For each end host i, assuming the individual DNS request arriving rate r_{ai} is following Possion distribution. Based on the superposition theorem, the arriving rate of total DNS request

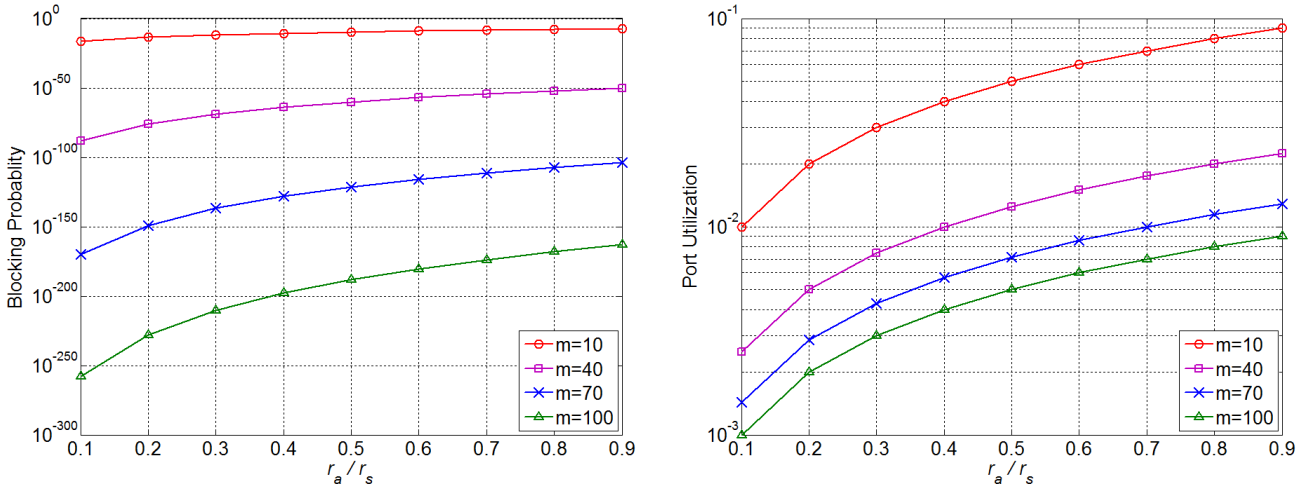
$$r_a = \sum_{i \in U} r_{ai} \tag{1}$$

also conform to Possion distribution, where U is the set of all end hosts. Assuming the service period follows negative exponential distribution and service rate is r_s. The volume of available port number is expressed with N_a, then the value of equivalent port number will be

$$m = \alpha \cdot k \cdot N_a, \tag{2}$$

where α and k are representing the multiplex ratio and the usable ratio, respectively. For instant, if the N_a is 16384 (i.e. 65535-49152+1). The multiplex ratio and usable ratio are 2 and 0.1%, respectively. Then the value of m is roughly equal to 32, which means 32 port number could be assigned to end hosts.

If the DNS server employs "never buffer requests" strategy, port switching requests will be directly denied or blocked when there is no available port number. By utilizing the Markov theory, the number of port switching request (NOT the DNS request) can be regarded as the system state. Based on transition pattern, we have:



(a) **(b)**

Figure 4 Port switching request rates is smaller than one time DNS inquiring rate

$$r_a p_{n-1} = n r_s p_n, \qquad n = 1, 2, \cdots \tag{3}$$

where p_n is the steady-state probabilities. Therefore,

$$p_n = p_0 \left(\frac{r_a}{r_s} \right)^n \frac{1}{n!}, \qquad n = 1, 2, \cdots \tag{4}$$

We could solve the probabilities of steady state 0

$$p_0 = \left[\sum_{n=0}^{m} \left(\frac{r_a}{r_s} \right)^n \frac{1}{n!} \right]^{-1} \tag{5}$$

by using

$$\sum_{n=0}^{m} p_n = 1. \tag{6}$$

Then the blocking probability (all m ports are occupied) will be

$$p_m = \left(\frac{r_a}{r_s} \right)^m \frac{1}{m!} \Big/ \sum_{n=0}^{m} \left(\frac{r_a}{r_s} \right)^n \frac{1}{n!} \tag{7}$$

The port utilization can be further expressed as

$$\beta_e = \left[1 - \left(\frac{r_a}{r_s} \right)^m \frac{1}{m!} \Big/ \sum_{n=0}^{m} \left(\frac{r_a}{r_s} \right)^n \frac{1}{n!} \right] \cdot \frac{r_a}{m r_s} \tag{8}$$

Equation (7) illustrate that the value of p_m only related with r_a, r_s and m. If we want improve the Quality of Service (QoS), i.e. reduce the blocking probability, compressing port switching request rates, shortening average period of one time DNS inquiring and increase the volume of equivalent port number will be quite reasonable and helpful.

Equation (8) represents the busy level of each equivalent port number. Since multiplexed is allowable and recommended, one physical transport layer port in DNS server might be visited by multiple end hosts, simultaneously. Therefore, the practical utilization of one physical transport layer port (expressed with β) might be beyond 100%. For instance, when busy level β_e is 95% and multiplex ratio α is 5, the value of β can reach to 475%. However, due to the complexity of available resource scheduling, it is highly suggested that multiplex ratio setting should consider r_a and r_s.

4.2 Performance Discussions

The modeling process provides an approach to quantitatively understand the blocking probability p_m and the port utilization β_e. The visible perspectives for these two parameters will be more convenient for further investigations. Since it is impossible to enumerate all the cases of port switching request rates r_a and one time DNS inquiring rate r_s, the ratio of them is selected as the independent variable. Figure 4 is generated according to the relation of numerator and denominator. Four representative values of equivalent port number m are considered to illustrate the tendency and variations of function value. For better displaying the contrast, logarithmic coordinate in Y direction is adopted.

In Figure 4(a), the ratio of r_a and r_s is changing from 0.1 to 0.9. For each 0.1 increment, the value of function is calculated. When setting the equivalent port number equal to 10, the curve of blocking probability (presented by red line with circle) is approximate a line. When the ratio value is set to 0.1, p_m is 2.49E-17. The differences to this value are only 1.63E-10 (0.5 in X axis) and 3.91E-08 (0.9 in X axis). If the value of m is enlarged to 40, the overall values of curve (presented by purple line with square) are getting lower. The nonlinearity is slightly displayed. The values in Y axis are 1.11E-88, 6.76E-61 and 7.37E-51 when 0.1, 0.5 and 0.9 are set in X axis. If the equivalent port number is increased to 70, the blocking probability (presented by blue line with star) will be further declined. The values in Y axis are 7.55E-171, 4.29E-122 and 2.13E-104 when 0.1, 0.5 and 0.9 are set in X axis. If the value of m is set to 100, the curve (presented by green line with triangle) illustrate obviously non linear characteristic. The minimum value of blocking probability is 9.70E-259 when the ratio of r_a and r_s is set to 0.1.

In Figure 4(b), the setting of r_a and r_s ratio is unchanged. From bottom to top, the lowest curve is representing the situation that equivalent port number is equal to 100. The utilizations for all cases are not high since the sufficient port number resource is supplied. The minimum value on this curve is only 1.00E-03. When shrinking the value of m to 70, the entire values of curve are getting larger. With the increasing of r_a and r_s ratio, the port utilization is enhanced. Some typical values are 1.43E-03, 7.14E-03 and 1.29E-02 when X axis is set to 0.1, 0.5 and 0.9. If we further decrease the equivalent port number to 40, the whole busy level for all port number will be increased as well. The differences

value is 2.00E-02 when the ratio value is set to 0.1 and 0.9. When minimizing the value of m to 10, the maximum in Figure 4(b) will be reached (the β_e is 9.00E-02 when the value of X axis is 0.9). And the differences value is 8.00E-02 when the ratio value is set to 0.1 and 0.9. Although the overall gaps among four curves in Figure 4(b) seem similar, the actual differences are quite huge due to the influence triggered by logarithmic coordinate.

5. CONCLUSIONS

The huge volume of requests generated from all kinds of electronic devices had brought tremendous pressures and serious privacy issues in current DNS. More research work pointed inside hackers may steal more crucial information of users, which had attracted great attentions from both academic and industrial communities. In this paper, we proposed a ports distribution management solution to enhance the user privacy inside local DNS. Firstly, the potential risks during DNS inquiring are discussed via attack types, DNS server location and information leakage influences. Secondly, the ports distribution schemes are designed from different perspectives and implementation the feasibility is also taken into consideration. Thirdly, a mathematical model for describing the port switching process is presented and a common scenario is established to evaluate the performance. There are some interesting findings in the results, such as the maximum utilization of equivalent port number is smaller than 10%. We hope this solution will be useful for both security enhancement and resources optimization in DNS.

6. ACKNOWLEDGMENTS

We would like to thank all the reviewers and editors for their invaluable comments and efforts on this article. This work was supported in part by the Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (B0190-16-2032, Development of Operating System Security Core Technology for the Smart Lightweight IoT Devices), in part by the Natural Science Foundation of China under Grant No. 61301081, in part by the Fundamental Research Funds for the Central Universities under Grant No. 2015JBM009, in part by the Project of State Grid Corporation of China under Grant No. SGRIXTJSFW [2016] 377.

7. REFERENCES

[1] Gao, H. Y., Yegneswaran, V., Chen, Y., Porras, P., Ghosh, S., Jian, J., and Duan, H. X. 2013. An empirical reexamination of global DNS behavior. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM* (SIGCOMM '13). ACM, New York, NY, USA, 267-278. DOI: http://dx.doi.org/10.1145/2486001.2486018

[2] Callahan, T., Allman, M., and Rabinovich, M. 2013. On modern DNS behavior and properties. *SIGCOMM Comput. Commun. Rev.* 43, 3 (July 2013), 7-15. DOI=http://doi.acm.org/10.1145/2500098.2500100

[3] Otto, J. S., Sánchez, M. A., Rula, J. P., and Bustamante, F. E. 2012. Content delivery and the natural evolution of DNS: remote DNS trends, performance issues and alternative solutions. In *Proceedings of the 2012 ACM conference on Internet measurement conference* (IMC '12). ACM, New York, NY, USA, 523-536. DOI=http://dx.doi.org/10.1145/2398776.2398831

[4] Gavai, G., Sricharan, K., Gunning, D., Hanley, J., Singhal, M., and Rolleston, R. 2015. Supervised and unsupervised methods to detect insider threat from enterprise social and online activity data. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*. 6, 4 (December 2015), 47-63.

[5] Claycomb, W. R. 2015. Detecting insider threats: who is winning the game?. In *Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats* (MIST '15). ACM, New York, NY, USA, 51-51. DOI=http://dx.doi.org/10.1145/2808783.2808794

[6] Satam, P., Alipour, H., Al-Nashif, Y., and Hariri, S. 2015. Anomaly behavior analysis of DNS protocol. *Journal of Internet Services and Information Security*. 5, 4 (November 2015), 85-97.

[7] Hao, S., Feamster, N., and Pandrangi, R. 2011. Monitoring the initial DNS behavior of malicious domains. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference* (IMC '11). ACM, New York, NY, USA, 269-278. DOI=http://dx.doi.org/10.1145/2068816.2068842

[8] Ferguson, A. D., Place, J., and Fonseca, R. 2013. Growth analysis of a large ISP. In *Proceedings of the 2013 conference on Internet measurement conference* (IMC '13). ACM, New York, NY, USA, 347-352. DOI=http://dx.doi.org/10.1145/2504730.2504769

[9] Calder, M., Fan, X., Hu, Z., Katz-Bassett, E., Heidemann, J., and Govindan, R. 2013. Mapping the expansion of Google's serving infrastructure. In *Proceedings of the 2013 conference on Internet measurement conference* (IMC '13). ACM, New York, NY, USA, 313-326. DOI=http://dx.doi.org/10.1145/2504730.2504754

[10] Giang, N. K., Im, J., Kim, D., Jung, M., and Wolfgang, K. 2015. Integrating the EPCIS and building automation system into the Internet of Things: a lightweight and interoperable approach. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*. 6, 1 (March 2015), 56-73.

[11] Van rijswijk-deij, R., Sperotto, A., and Pras, A. 2014. DNSSEC and its potential for DDoS attacks: a comprehensive measurement study. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (IMC '14). ACM, New York, NY, USA, 449-460. DOI=http://dx.doi.org/10.1145/2663716.2663731

[12] Ballani, H. and Francis, P. 2008. Mitigating DNS DoS attacks. In *Proceedings of the 15th ACM conference on Computer and communications security* (CCS '08). ACM, New York, NY, USA, 189-198. DOI=http://dx.doi.org/10.1145/1455770.1455796

[13] Booth, T. and Andersson, K. 2015. Network security of Internet services: eliminate DDoS reflection amplification attacks. *Journal of Internet Services and Information Security*. 5, 3 (August 2015), 58-79.

[14] Chitpranee, R. and Fukuda, K. 2013. Towards passive DNS software fingerprinting. In *Proceedings of the 9th Asian Internet Engineering Conference* (AINTEC '13). ACM, New York, NY, USA, 9-16. DOI=http://dx.doi.org/10.1145/2534142.2534144

[15] Khalil, I., Yu, T., and Guan, B. 2016. Discovering malicious domains through passive DNS data graph analysis. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security* (ASIA CCS '16). ACM, New York, NY, USA, 663-674. DOI=http://dx.doi.org/10.1145/2897845.2897877

[16] Shulman, H. and Ezra, S. 2014. POSTER: On the resilience of DNS infrastructure. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (CCS '14). ACM, New York, NY, USA, 1499-1501. DOI=http://dx.doi.org/10.1145/2660267.2662376

[17] DNS PRIVate Exchange (dprive) Working Group (WG) https://datatracker.ietf.org/wg/dprive/documents/

[18] Internet Engineering Task Force (IETF) https://www.ietf.org/

[19] Hands, N. M., Yang, B., and Hansen, R. A. 2015. A study on notnets utilizing DNS. *In Proceedings of the 4th Annual ACM Conference on Research in Information Technology* (RIIT '15). ACM, New York, NY, USA, 23-28. DOI=http://dx.doi.org/10.1145/2808062.2808070

[20] Yadav, S., Reddy, A. K. K., Reddy, A. N., and Ranjan, S. 2012. Detecting algorithmically generated domain-flux attacks with DNS traffic analysis. *IEEE/ACM Trans. Netw.* 20, 5 (October 2012), 1663-1677. DOI=http://dx.doi.org/10.1109/TNET.2012.2184552

[21] Hao, S., Thomas, M., Paxson, V., Feamster, N., Kreibich, C., Grier, C., and Hollenbeck, S. 2013. Understanding the domain registration behavior of spammers. In *Proceedings of the 2013 conference on Internet measurement conference* (IMC '13). ACM, New York, NY, USA, 63-76. DOI=http://dx.doi.org/10.1145/2504730.2504753

[22] Herzberg, A. and Shulman, H. 2014. DNS authentication as a service: preventing amplification attacks. In *Proceedings of the 30th Annual Computer Security Applications Conference* (ACSAC '14). ACM, New York, NY, USA, 356-365. DOI=http://dx.doi.org/10.1145/2664243.2664281

[23] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and Cheshire, S. 2011. Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry. RFC 6335.

[24] Eastlake, D. and Kaufman, C. 1997. Domain Name System Security Extensions. RFC 2065.

[25] Eastlake, D. 1999. Domain Name System Security Extensions. RFC 2535.

[26] Arends, R., Austein, R., Larson, M., Massey, D., and Rose, S. 2005. Protocol Modifications for the DNS Security Extensions. RFC 4035.

[27] Weiler, S. and Blacka, D. 2013. Clarifications and Implementation Notes for DNS Security (DNSSEC). RFC 6840.

[28] Damas, J., Graff, M., and Vixie, P. 2013. Extension mechanisms for DNS (EDNS(0)). RFC 6891.