JSERNET UDP53 Port Traffic Analysis

Yanjun Su, Wei Ding, Shi Dong

Department of Computer Sciences and Engineering, Southeast University, Nanjing, 211100, China
E-mail:{yjsu,wding,shdong}@njnet.edu.cn

Abstract: As the well known port for DNS service, the 53 port won't be shielded by most of firewalls or network services. This defect can be used to send packets through firewalls without being detected. For comprehensive understanding of this phenomenon, a program was designed to inspect UDP flows going through the port 53. With the help of this program, we conducted traffic analysis on the 53 port of JSERNET and obtained the proportions of DNS flow and non-DNS flow, which were used to describe the real components of such flows.

keywords:Traffic analysis; Domain Name System; Identification

I. INTRODUCTION

Public ports are tightly bound to some special services in the previous network, such as HTTP traffic uses port 80, DNS traffic uses 53 port. Therefore, when designing security mechanisms, firewalls usually block the traffic whose port number cannot be identified. Traffic from public ports is regarded as normal flow at the same time. With the constant expansion of Internet and the rise of various network applications, this situation has begun to change. Some applications or attacks may use these public ports to camouflage themselves because this disguise will help them pass through the firewall or router. Therefore, the traditional port-based method for determining the network protocol and designing security policy have been hampered in today's network.

Domain Name System (DNS) is one of the Internet's core services, which enable Internet users to have a more humane domain name to identify the network nodes, without remembering a huge number of IP addresses, thereby providing for easy access to network [1]. 53 port is the port used by DNS service, the client sends a query to the 53 port of a DNS server, the server will parse the query and return the response to the client. However, along with normal DNS packets, plenty of through traffic also uses 53 port in current network. Therefore, in order to find out the real state of DNS packets in the network, we take a traffic analysis for JSERNET UDP53 port and obtain the ratio of DNS traffic flow and the proportion of non-DNS flow.

In this study, the main advantage is IP Trace, which is captured by the network center of northeast regional CERNET in the border of Jiangsu net. Our study designs a inspection procedure for UDP53 port according to the format of IP Trace, and makes use of the standard answer to determine the procedure's precision. The observation of IP Trace in 2005 shows that the partition of DNS packets in UDP53 port packets is more than 99%, while after 2009 the partition is only about 10% or less. This phenomenon fully indicates that non-normal use of 53 port becomes more and more frequent.

II. BACKGROUND

A. IP Trace

The research is based on IP Trace that captured by the network center of northeast regional CERNET with the purpose of supporting collation and analysis of network data. The collection point is in the border of Jiangsu net. The data used in the experiment, is stored in a format as Figure 1, and they are finally organized into a binary file with the size of 200M. Considering limited storage and privacy of other network users, the acquisition system uses limited length capture mode rather than the whole packet capture mode. In this study, the main difficulty is how to use these incomplete packets to filter out the real DNS packet.

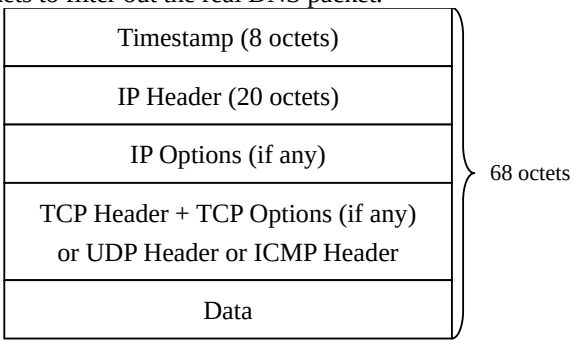

Figure 1. The specific format of each packet

B. Related Work

As the growing number of emerging network services select UDP as their underlying transport protocol. So UDP traffic analysis also has been the concern of scholars in recent years. Some of researches are the port-distribution analysis of UDP. Y.B. Zhang listed his analysis of a domestic backbone router in the article [2], 53 port had the second highest usage frequency among all ports. We also can see that 15.8% of UDP flows ran on 53 port in another figure of the article. However, a report of CAIDA in 2009[3] showed that only 3%-4% of UDP flows ran on 53 port. Differences between the two data make us to study the reason. But present study for UDP53 port is around DNS attacks. DNS is the basis for most part of network, but due to design flaws in the agreement itself that did not provide information on protection and authentication mechanism. The DNS server becomes vulnerable [4]. So scholars have been exploring the DNS security issues. Our article is only about the traffic statistics of UDP53 port caused by the front doubt. We restore the real situation of traffic on UDP53 port to seek the reason.

On that basis, we organize the overall working steps. The rest of the paper is organized as follows: In section Ⅲ, we get characteristics that can be used to design our algorithms

through analyzing the format of a standard DNS packet. Our core algorithms are established in section Ⅳ. The accuracy of inspection procedure is evaluated based on the standard data in section Ⅴ. At last, we list the proportions which are classified by our identification algorithm.

III. DNS PACKET INSPECTION

A. *DNS Packet Format*

DNS defines a format for query message and response message. Table Ⅰ shows the general format of a DNS packet. The message consists of a 12 bytes long header and four variable-length fields [5]. The identification field is set by the client and returned by the server, Client use it to determine whether the response matches the query packet. The header is followed by 16-bit flag field and the remaining four parts are the amount of questions, answers, authority resource records, and additional resource records. The last four variable-length fields correspond to the specific content of questions, resource records and so on.

TABLE Ⅰ. FORMAT OF DNS PACKET

0 15	16 31
Identity ID	Mark
Number of questions	Resource records
Record number of authorized resources	Record number of additional resources
Query question(variable length)	
Answer(variable length)	
Authorization information(variable length)	
Additional Information(variable length)	

16-bit flag field has been divided into sub-fields in Figure 2. QR (1 bit): query / response flag, 1stands for response and 0 stands for query. Opcode (4 bits): defines the type of query or response. AA (1 bit): the flag for authorization. This bit is only valid in the response message. A name server is a privilege server if the flag is set to 1. TC (1 bit): truncate flag, 1 shows that the response is more than 512 bytes and has been truncated. RD (1 bit): This bit is 1 if the client asks for a recursive answer. RA (1 bit): only in the response packet is set to 1, indicates that the response can be recursive. Zero (3 bits): Reserved field should be zero. Rcode (4 bits): return code indicates the error state of response. Figure 3 shows a normal DNS response packets which analyzed by wireshark, Its identification is 0x8180, question is 1, answer number is 3, authority resource record number is 2, additional resource record number is 3 .

Figure 2. 16-bit flag in DNS packet header

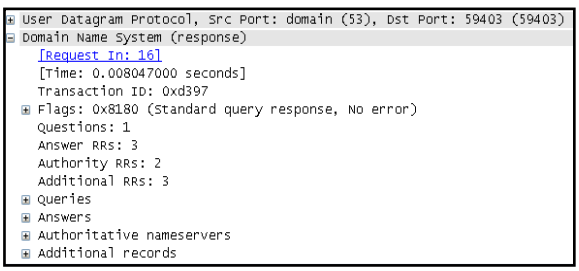

Figure 3. DNS packet

B. *DNS Packet Detection Principle*

The storage size of a packet in IP Trace is 68 bytes and a DNS packet generally attaches with a UDP packet. We must remove 8-bytes timestamp, 20-bytes IP header (no options), and 8-bytes UDP header, then the remaining 32-bytes belong to the part of DNS packet. The principle of packet detection is on the basis of conditions that the 12-bytes DNS packet header must satisfy. The following three conditions are designed in our article.

(1) The highest bit of opcode and rcode must be zero. Figure 2.2 shows that opcode and rcode is a four-digit mark. Opcode is the query type; such as 0 is a standard query, 1 is a reverse lookup. Rcode is the return code set by server, for example, 0 is no error, 3 is an error of name. According to the current implementation of DNS packet protocol, 0 to 5 of them have been used, and 6 to 15 are reserved temporarily. So the first condition is designed for this feature.

(2) The number of question must be 0 or 1. For query packet, the number of questions must be 1, and for response packet the number can be either 0 or 1.

(3) The number of records is reasonable. According to the definition of DNS packet, the number of answers, authority resource records and additional resource records represents the reported number of records in the three fields. Meanwhile, UDP header can be used to calculate the bytes which correspond to these three records. The third Condition discusses the relationship between the bytes and the number of records.

Though the last three fields of DNS packet are variable length fields, they all use a same format which called RR (Resource Record), Table Ⅱ shows the format. The minimum size of NAME is 2 bytes if it uses compression mode (16-bit pointers). The total size of response type, response category, survival time and data length is 10 bytes because they are all fixed-length. Due to resource data's size is indeterminate, we can't get the minimum size. So we choose to omit this part. In summary, the minimum size of a DNS resource record is determined as 12 bytes here.

TABLE Ⅱ. FORMAT OF RESOURCE RECORD

0 15	16 31
NAME (variable length)	
Response Type	Response category
TTL	
Data length	Resource Data (variable length)

Rss_len is set as the size of three records. The UDP packet size that can be got from the UDP header minus 8 bytes (UDP header), 12 bytes (DNS header) and the size of question is rss_len. The length of the question is non-fixed, the format is shown in Table Ⅲ. The query name is uncertain and irregular, so it is ignored. The total size of the query type and query class is 4 bytes. Therefore rss_len discussed here is the largest rss_len.

TABLE Ⅲ. FORMAT OF QUESTION

0 15	16 31
Query name (variable length)	
Query type	Query class

If we use the largest rss_len to be divided by minimum record size, the result is considered as the upper limit of records number. If the sum of records number calculated according to DNS header is over the maximum number of records. The

phenomenal will show unreasonable, then we can determine the packet is a non-DNS message.

C. Example Packet

Figure 4 shows a normal DNS query packets. The opcode of this packet is 0, rcode is 0, satisfy the first conditions. The number of question is 1, meeting the second condition. The size of UDP packet is 39 (not shown), so the corresponding rss_len is 15, and maximum number of records is 1, while the sum of records in figure is 0, satisfying the third conditions. To the end, the message will be judged as a real DNS packet which is consistent with the actual situation.

Figure 5 shows a through packet. Opcode is 11 that does not meet the first condition. So it is determined to be a through packet.

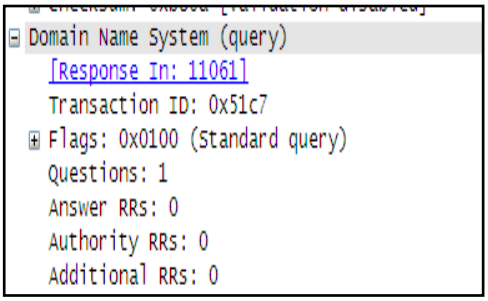

Figure 4. A normal DNS packet

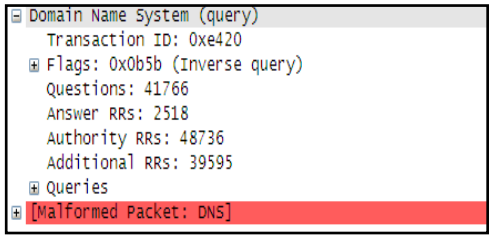

Figure 5. A through packet

IV. VERIFICATION

In order to verify the accuracy of test procedure, our experiment collected 25 minutes packets in the border of Jiangsu net. These data are different from IP Trace which is organized in 68-bytes. The data for verification are saved as integrate packet. Then we can check the whole packet to determine whether the packet is a DNS packet and the result can be used as a standard answer. The main method for checking packet is a function called ns_initparse which is provided by the library of libbind. The function can be used to resolve a packet according to the format of DNS packet described in section Ⅲ. If a packet is a real DNS packet, the function will accomplish the analysis successfully. On the other hand, if a packet is not a DNS packet, the function will not accomplish the analysis. At the same time, we will affirm the packet is not a DNS packet. So from the view of DNS packet's definition, the determination method is absolutely right. The verify process is designed as follows:

step1: get 25 minutes data from Jiangsu net boundary and these data are saved as integrate packet.

step2: call ns_initparse function to resolve the packet. If the return value is 1, it shows that the packet is a DNS packet. If the return value is 0, it shows that the packet is a non-DNS packet.

step3: the answer is stored in a byte. 0 is no, and 1 is yes. Then we intercept every packet with the format of IP Trace except timestamp. The answer will be stored in the front of the "IP Trace", so the size of a packet is 61 bytes.

step4: organize all 61-bytes packets as a binary file.

step5: apply inspection procedure designed in section Ⅲ to the file.

setp6: compare the answer get by the inspection procedures with the standard answer.

We divide 25 minutes to five 5 minutes with the purpose of testing inspection procedure's stability. The result is shown in Table Ⅳ, all of the precision rates are more than 99.4% and the recall rates are nearly 100%. Therefore, according to the table, we know that the determination method presented in section Ⅲ has a high precision rate and recall rate. The method is precise enough to be applied in real data (IP Trace).

TABLE Ⅳ. VERIFY RESULTS

No.	53 port packets	DNS packets	Precision	Recall
1	3245133	1617837	99.50%	100.00%
2	3435804	1615204	99.5%	100.00%
3	3398826	1612866	99.53%	100.00%
4	3420075	1623985	99.45%	99.99%
5	3393680	1621648	99.45%	99.99%

V. EXPERIMENT RESULTS AND ANALYSIS

A. Datasets and Essential Information

The data for experiment is IP Trace from 2005 to 2010. Specific measurement data and its essential information is in Table Ⅴ. Table Ⅴ lists the collecting line, capture time, duration and the corresponding number of packets. The collecting line is decided by source and destination IP. The collector separates packets into four lines. As we can see from Table Ⅴ, the proportion of UDP packets in total packets is 6.99% in 2005 but it rises to 50% in 2008, indicating that variety of UDP-based network applications have begun to popular in recent years. Of course, many researchers also mention it. Mena found that 60% of 80% of the audio data steam is carried over UDP traffic [6]. What's more, Sripanidkulchaiso presented a statistic of a well-know content delivery network which showed UDP has a absolute advantage in audio/video field [7]. So the proportion of UDP in our research shows a very significant improvement. The table also shows that the value of UDP53/UDP in 2005-2008 decreases with the increase of UDP/IP. We suppose that the number of non-53 port UDP packets have a large increase, but 53 port packets only have a very small increase. Because according to these new applications, they generally use a large port and not use some known port (for example, 53). Thus the phenomenon leads to the decrease of UDP53/UDP. But the value of UDP53/UDP has a very substantial growth in 2009 and 2010, relative to previous years. However, DNS packets can't have such a big increase according to the actual situation of network. So we guess that most of these "DNS packets" may be pseudo-DNS packets which just use port 53. Our proposed detection method just can restore the real situation of the traffic.

B. Results

Table Ⅵ lists the result , consisting of the number of UDP 53-port packets, the number of DNS packets and the ratio between the two. The value of DNS/UDP53 is more than 99.5% in 2005-2006 and decreases slightly in 2007-2008. However, the value falls to 3.553% and 11.087% in 2009 and 2010, indicating that there are a large number of packets in the use of non-DNS 53 ports from 2009.

VI. CONCLUSION

In this paper, we take a traffic analysis for JSERNET UDP53 port with DNS packet inspection procedures whose precision rate is over 99.4 %. Datasets are distributed in 2005-2010, from the measurement results that we can see, through packets which use 53 port increases in 2009. It makes the value of DNS/UDP53 in 2009 and 2010 drop to 3.553% and 7.178%. The result will make a contribution to the deployment of firewall of Jiangsu net in the future. What's more, the three conditions can be applied to the firewall if it is possible.

TABLE Ⅴ. DATASETS

Collecting line	Capture time	Time duration	All packets/10^6	UDP packets/10^6	UDP53 packets/10^3	UDP/All	UDP53/UDP
all	2005-11-10	14:00-15:00	2342	164	5929	6.99%	3.606%
all	2006-12-31	14:00-15:00	1662	373	5315	22.44%	1.423%
1	2007-10-30	14:00-14:40	170	38	510	22.53%	1.332%
1	2008-12-20	14:00-15:00	724	340	1993	49.96%	0.585%
1	2009-12-18	14:00-16:00	1896	923	132416	48.68%	3.585%
1	2010-07-18	14:00-16:00	1141	435	31277	38.12%	7.178%

TABLE Ⅵ. TEST RESULT

Collecting line	Capture time	UDP53 packets	DNS packets	DNS/UDP53
all	2005-11-10	5929217	5900105	99.509%
all	2006-12-31	5315218	5312967	99.970%
1	2007-10-30	510556	499081	97.752%
1	2008-12-20	1993835	1933694	96.983%
1	2009-12-18	132416594	4705893	3.553%
1	2010-07-18	31277250	3468021	11.087%

ACKNOWLEDGMENTS

This paper is supported by National 973 Plan Projects (2009CB320505) and National Science and Technology Plan Projects (2008BAH37B04).

REFERENCES

[1] J. Tian, D.W. Gu, and H.N. Lu, "A Solution for Packet Validity Check Against DNS Cache Poisoning," Communication Technology, vol. 43, pp. 146-146, 2010.

[2] Y.B. Zhang, Z.B. Zhang, Y. Zhao, and L. Guo, "Comparat ive analys is on TCP and UDP network traffic,"Application Research of Computers, vol. 27, pp. 2195-2196, 2010.

[3] http://www.caida.org/research/traffic-analysis/tcpudpratio/

[4] B.R. Yan, B.X. Fang, B. Li, and Y. Wang, "Detection and Defence of DNS Spoofing Attack," Computer Engineering, vol. 32, pp. 130-130, 2006.

[5] W.R. Stevens, and G.R. Wright, TCP/IP Illustrated: the protocols, Addison-wesley, 1994.

[6] A. Mena, and J. Heidemann, "An empirical study of real audio traffic," INFOCOM 2000. Proc of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE Press, 2000, pp. 101-110.

[7] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An Analysis of Live Streaming Workloads on the Internet," Proc of the 4th ACM SIGCOMM conference on Internet measurement, ACM Press, 2004, pp. 41-54.