3.8 Ruo Ando, Yuuki Takano, Satoshi Uda/ Unraveling large scale geographical distribution of vulnerable DNS servers using asynchronous I/O mechanism

3.8.1 Ruo Ando

Ruo Ando has received Ph.D. from Keio University in Japan.He is now senior security researcher of NationalInstitute of Information and Communication Technology in Japan. Also, he has been working as TechnicalOfficial of Ministry of Internal Affairs and Communications since 2006. His research interests are Cloudcomputing technologies and its security.He has been working in Driverware "Immune" project supported by USAir Force Office of Scientific Research withGrant Number AOARD 03-4049 in 2005-2006. He receivedOutstanding Leadership Award in the8th IEEE International Conference on Dependable, Autonomic and SecureComputing (DASC-09) at China in 2009. He is the member of Trusted Computing Group JRF (Japan Regional Forum). He has presented in many security conferences such as SysCan Singapore 2009 and PacSec Tokyo2011. His research products such as inforamtion gathering system, DHT crawler and vulnerability analysissystem are now deployed on the large scale Test bed of National Institute of Information and CommunicationTechnology. He recently presented secure Cloud computing technologies in Singapore(2009) and Taiwan(2010).He served as reviewer of Willey Journal of Security and Communications Networks and IEEE transactions ofInformation Forensics and Security.

- `http://sites.google.com/site/andoruo`
- @And_Or_R

3.8.2 Yuuki Takano

`http://www.informatik.uni-trier.de/~ley/pers/hd/t/Takano:Yuuki`

3.8.3 Satoshi Uda

- `http://www.uda-lab.imr.tohoku.ac.jp/member/uda_e.html`
- `http://www.researchgate.net/profile/Satoshi_Uda/`

3.8.4 Unraveling large scale geographical distribution of vulnerable DNS servers using asynchronous I/O mechanism

The Domain Name System (DNS) has become one of the most important infrastructures of Internet. Despite of its importance, we have not obtained the comprehensive view of DNS servers deployed in real-world to evaluate the security level with the fine-grained information. This paper we present some results of analyzing DNS servers in some security concerns such as software version and geographical distribution. In experiment, we have succeeded to obtain information of 10,334,293 DNS servers in 24 hours. For rapid crawling, we adopt Libevent which provides asynchronous I/O mechanisms and MongoDB which is fast and document based NoSQL cluster. By analyzing the result of 24 hours monitoring, we have found some important facts for security assessment of DNS deployment in Internet. For example, more than 1000 servers still uses the oldest version of BIND 4.x. Besides, we show in-depth study of geographical distribution of vulnerable DNS servers with time series analysis. It is shown that even advanced IT countries achieving high security level has "weakest link" which means these countries actually has vulnerable DNS servers. Also, it is turned out that the large scale information gathering of vulnerable DNS servers could be easily achieved in only several hours.

- Talk and paper can be downloaded from `http://grehack.org`

Unraveling large scale geographical distribution of vulnerable DNS servers using asynchronous I/O mechanism

Ruo Ando †, Yuuki Takano †, Satoshi Uda ††

†Network Security Institute, National Institute of Information and Communication Technology,
4-2-1 Nukui-Kitamachi, Koganei,
Tokyo 184-8795 Japan
††Research Center For Advanced Computing Infrastructure,
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292 Japan

Abstract. The Domain Name System (DNS) has become one of the most important infrastructures of Internet. Despite of its importance, we are not able to comprehensive view of the situation of deployment of DNS servers in real-world and therefore assess the security level according to the monitoring results. In this paper we present some results of analyzing DNS server deployment over Internet fine-grained information of software versions, geographical locations. In experiment, we have succeeded to gather information of 10,334,293 DNS servers over the world in 24 by leveraging asynchronous event notification library. For high-speed active monitoring, we adopt Libevent which provides asynchronous I/O mechanisms and MongoDB which is fast and scalable NoSQL cluster software. From monitoring results, we show some findings which should be considered for security assessment for DNS deployment over Internet. Surprisingly, more than 1000 DNS servers employ the oldest version of BIND 4.x which could be easily compromised by malicious hosts. Besides, we show in-depth study of geographical distribution of vulnerable DNS servers which can provides comprehensive view of security level of each country. It is turned out that even advanced IT countries achieving high-security level has " weakest link" exposed by fine-grained active monitoring, which means these countries actually have serious vulnerabilities concerning IT infrastructure. In addition, it is shown that large scale active monitoring of DNS servers have been easily achieved within only several hours.

Keywords: DNS, vulnerability assessment, geographical distribution, asynchronous I/O, rapid and scalable monitoring

1 Introduction

Nowadays, The Domain Name System (DNS) which was originally designed for one-to-one mapping s between two kinds of logical address space (a domain

name and an IP address) has a mission-critical infrastructure of Internet. However, despite its importance, unfortunately, there have been many attacks on DNS servers using exploitation such as DNS cache poisoning and Open Resolver based DDoS amplification. To make things worse, only poor and coarse-grained information was provided about limited region about the situation of DNS deployment over Internet. So far, we have not taken a comprehensive view for security assessment of DNS server deployment all over the world. In this paper we present the fine-grained active monitoring results of DNS server deployments. We show the statistical results of DNS software versions and geographical distribution of DNS servers. We have inspected the current situation of software versions such as BIND, DNSMASQ, Nominum and so on.

One of the main important findings is that more than 20,000 servers is still using obsolete (and therefore vulnerable) DNS software such as BIND 4.x, 8.x versions. Also, it has been revealed that even advanced IT countries has a weak link which is built by vulnerable DNS servers remaining unmonitored inside their countries. In fact, vulnerable servers has been found in the countries where IT infrastructure is widely pervasive rather than relatively less advanced countries. For the better security assessment, we have implemented asynchronous I/O based rapid and massive crawling system which can gathering information all over the Internet within 24 hours. It is implied that attacks against DNS servers of old software version could be easily discovered and compromised in a short time.

The remainder of this paper is structured as follows: we start with introducing methodology for crawling techniques and its measurement scope in Section 3. In Section 4, we present the detailed measurement and analysis active monitoring of DNS servers. Some statistics and ranking about software versions, geographical location and time series of DNS query response are presented. After related work of section 5, section 6 presents the evaluation of the monitoring result and some points which has not been discussed. In Section 7, we conclude and present some further works.

2 Proposed system

DNS is one of the most scalable systems in Internet. Currently, it is not measured or estimated accurately how many DNS servers are running in Internet. However, the number of DNS servers deployed all over the world is huge, which means monitoring result is big data which cannot be handled by conventional detection systems. For coping with DNS servers, we have constructed the scalable, rapid crawling system using Libevent [14] and MongoDB [15]. Figure 1 show the proposed system which consists of two physical servers.

On frontend server, crawler which employs Libevent (event notification library) is running concurrently. Crawlers issues queries based on asynchronous I/O method which is suitable for handling DNS commands of which response time cannot be estimated exactly. For coping this problem, we employ MongoDB which is document-based data store on backend cluster. MongoDB is surely scal-

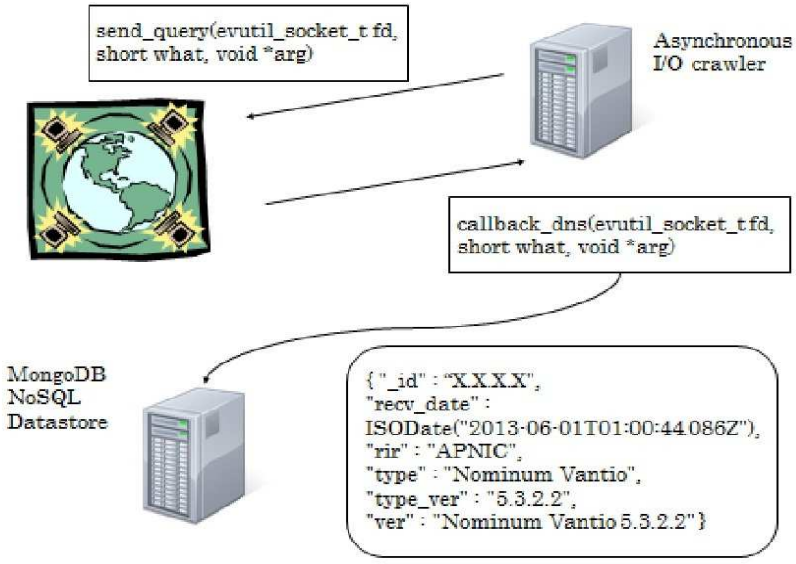

Fig. 1. Proposed system using Libevent and MongoDB. The handler of crawler on frontend (callback_dns) sends query to store key-value of response from DNS server to MongoDB on backend.

able and besides has several useful functions for data processing such as aggregating, indexing and sorting using key-value. For example, by leveraging Map-Reduce, we can easily extract information such as geographical distributions and statistics of software versions of DNS.

3 Methodology

In this section we present our methodology of active monitoring of DNS servers.

First, some utilities we leverage for rapid and massive monitoring are introduced: 1) Libevent which provides asynchronous I/O mechanism and 2) MongoDB which is document based data store for scalable cluster by advanced functions such as sharding. Second, before presenting the results of our measurement in Section 3, we discuss the measurement scope of active monitoring for fetching DNS versions and geological distribution. Then, the kinds of queries our system issue and the detected version of DNS software is listed.

3.1 Callbacks for asynchronous event notification

Although there has not been exact measurement or estimation, it is sure that the number of DNS servers is huge. Actually, conventional monitoring method of

blocking send/recv and SQL database cannot handle this kind of "big data". For achieving rapid, massive and scalable active monitoring of DNS, we implement asynchronous key-value based tracking system.

Our system design is based two concepts. First, we should consider that the response time of queried DNS servers cannot be estimated. So we employ Libevent (event notification library) for providing asynchronous I/O mechanism between sending and receiving messages. Second, we should cope with massive query response from numerous DNS servers. Besides, one of the important purposes of our system is to generate some statistics of monitoring. For this challenge, we leverage MongoDB which is document-based scalable data store.

As a result, to efficiently gather information of massive DNS servers, we have succeeded to implement simple querying system in C++ which is able to handle more than 10 million of DNS queries and answer messages (responses) by concurrently performing with MongoDB using Libevent.

In deployment, crawlers on frontend uses Libevent and backend analyzer used MongoDB. The Libevent based crawler which leverages asynchronous I/O on NoSQL cluster achieves high-speed active monitoring. We have succeeded to connect 10,334,293 DNS servers in 24 hours.

```
--- LIST1: send_query in Figure 1 ---
ev_dns = event_new(ev_base, sockfd, EV_READ | EV_PERSIST,
callback_dns, NULL)
event_add(ev_dns, NULL)

timeval tv = {0, QUERY_CYCLE * 1000};
ev_send = event_new(ev_base, -1, EV_TIMEOUT | EV_PERSIST,
send_query, NULL);
event_add(ev_send, &tv);
```

LIST 1 shows code snippets of two callbacks. There are two terminate conditions for each session to query DNS. Send_query issues query about software versions with timeout (tv). Callback_dns process the response from DNS server which is invoked when file descriptor is available for reading (EV_PERSIST). For each session for querying DNS server, these two callbacks are assigned and processed in asynchronous I/O mechanism. In the case that the query has response (DNS server respond to our crawler), parsed reponse is stored in MongoDB.

```
--- LIST2: callback DNS in Figure 2 ---
auto_ptr<mongo::DBClientCursor> cur;
mongo::BSONObjBuilder b;
mongo::BSONObj doc;
mongo::Date_t recv_date;
b.append("_id", addr);
b.append("recv_date", recv_date);
-- snip --
p_txt = DNS_RDATA_TO_TXT(it->m_rdata);
```

```
b.append("ver", p_txt->m_txt);
-- snip --
doc = b.obj();
mongo_conn.insert("DNSCrawl.servers", doc);
```

LIST 2 shows code snippets of processing the answer from DNS servers. Our system parses the response for getting items such as IP address, recv data and software versions. Then our system stores these items into list by method chain. At the last line, our system stores key-value pairs into db of "DNSCrawl" and collection of "servers".

After gathering information of 10,334,293 servers, we utilize the routines for aggregation and sorting which is called as Map Reduce. Map Reduce is one of the useful utilities which MongoDB provides for aggregating and sorting optimized for large scale data processing. Also, we use MongoDB perl driver to make a simple script for pre-processing.

There are some reason why we choose MongoDB. For example, MongoDB could be easily scaled out by sharding. By leveraging these utilities, we have handled massive query responses from 10,334,293 DNS servers in 24 hours.

3.2 Scalable geolocation lookups

For translating huge output from DNS crawler into geographical information, we deployed geolocation querier on another physical machine.

Figure 2 shows how crawlers which sends fetched IP (logical) addresses and our analyzer (extractor) which extracts geographical (physical) address work in parallel. Crawler and extractor connect MongoDB server in parallel on the left side of Figure 2. Proposed method is divided into four steps.

```
[1] Crawler running on frontend stores information about IP
    address of each DNS servers into MongoDB.
[2] At the same time, extractor on the right side of figure
    is issuing a query.
[3] Then, extractor invokes GeoIP lookup routine to retrieve
    geographical information.
[4] Finally, extractor store the results of GeoIP into MongoDB.
```

In our system, crawlers and extractors are running on two different physical machines, which enables crawlers and extractors to be scaled out. Importantly, any modification of data format and processing on one side does not take effects on the other side.

3.3 Measurement Scope

We have crawled 10,334,293 servers in 24 hours using two machines. In measurement, we have detected old versions of BIND 4.x and 8.x Nomium, PowerDNS and so on. More than 40% of all connected servers did show the banner. Surprisingly, many DNS servers with the obsolete version of BIND such as 8.x and

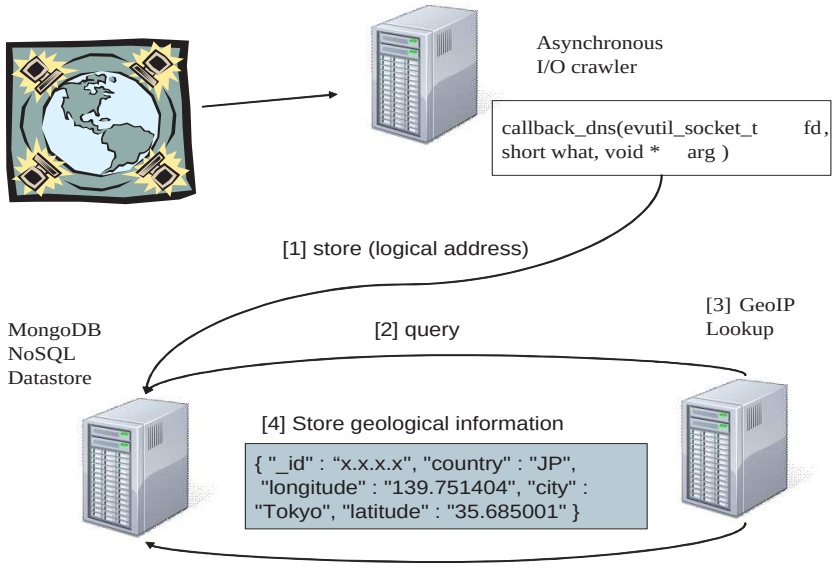

Fig. 2. Extractor of Geographical information using GeoIP lookup is deployed on another physical machine. Crawling and GeoIP lookup modules can be run in parallel. Besides, querier of geolocation is able to be scaled out by using flag indicating completed lookups.

4.x has been detected. Also, we have monitored approximately 94% of all servers which is registered to APNIC, RIPE, ARIN, LACNIC and AFRNIC.

We have tracked 10,334,293 servers and fetched information corresponding to items as follows.

- IP address
- recv datetime
- software version
- country, city, latitude and longitude

Among 10,334,293 servers, we have clearly identified about 4 million servers by the banner in response message. Also, we have extracted geographical location using GeoIP and measures recv datetime. IP address, version, receiving time is obtained in real-time. Then geographical information is obtained in off-line on backend cluster.

Table 1 shows list of software name and its regular expressions. In this paper, we specify BIND into three category (4.x, 8.x and 9.x). As shown in Table 2, more than 70% of servers of which the software version was detected are BIND, DNSMASQ, and Nominum. Also, Table 2 shows types of DNS servers with RIR (Regional Internet Registry) code.

Table 1. Regexs for DNS Type Detection

Type of DNS	Regex
BIND 9.x	^9(\.[0-9])+
BIND 8.x	^8(\.[0-9])+
BIND 4.x	^4(\.[0-9])+
DNSMASQ	^dnsmasq
Nominum Vantio	^Nominum Vantio
Nominum ANS	^Nominum ANS
PowerDNS	^PowerDNS
Unbound	^unbound
NSD	^NSD
Windows series	.*Windows

4 Measurement and Analysis

In this section, we present the result from our crawling 10,334,293 servers using asynchronous I/O mechanism which is constructed by event notification library. First, one of the important our findings that many servers do employ obsolete version of BIND 4.x, 8.x. Second, we have generated statistics about countries where each server is running. We have discovered many obsolete DNS versions and its geographical locations with off-line analysis. Statistics of monitoring results of software version of BIND 4.x, 8.x and countries are shown in Table 5.

4.1 DNS Versions

Currently, it has been found that Bind 4.x, Bind 8.x and some versions DNSMASQ have many serious vulnerabilities and therefore it is strongly recommended to update these version to the latest. Surprisingly we have discovered 1935 servers of BIND 4.x, 15771 servers of BIND 8.x and 946294 servers of DNSMASQ. Table 3 shows top 15 most frequently appeared version of BIND 4.x, 8.x and DNSMASQ. For example, multiple vulnerability has been found for DNSMASQ version 2.5 and earlier. Surely, BIND 4.x and 8.x are already obsolete and has lots of implementation flaw. Besides, unfortunately, the BIND-8.4.7 which is relatively new version among this list does have vulnerability according to www.cvedetails.com [16]. Also, BIND-4.9.11 or earlier has many serious potential and already-exploited vulnerabilities.

4.2 Geographical location lookups

We have extracted geographical location information after the active monitoring is done. In our system GeoIP is used for each IP address of DNS servers. GeoIP is a unix command which is called as geoip. As we described earlier, we

Table 2. Types of DNS Servers with RIR code

Type of DNS	Total #	APNIC #	RIPE #	ARIN #	LACNIC #	AFRINIC #	other #
BIND 9.x	2,369,863	336,263	769,182	860,335	96,703	10,953	296,427
BIND 8.x	15,771	3,265	7,065	3,828	355	15	1,243
BIND 4.x	1,935	99	1,362	349	28	-	97
Dnsmasq	946,294	495,205	158,282	59,145	159,969	25,993	47,700
Nominum Vantio	450,079	209,051	198,019	18,808	14,500	7,465	2,236
Nominum ANS	502	15	23	67	25	-	372
PowerDNS	94,299	4,946	57,115	28,138	1,013	35	3,052
Unbound	30,588	5,461	17,926	5,447	1,030	206	518
NSD	25,837	1,296	7,955	13,835	257	13	2,481
Windows series	5,324	1,296	386	400	3,217	-	25
can't detect	3,067,979	1,943,992	620,895	291,737	113,120	9,706	88,529
no version info	3,325,822	739,726	1,307,181	710,867	327,504	29,272	211,272
Total	10,334,293	3,740,615	3,145,391	1,992,956	717,721	83,658	653,952

have connected approximately ten million servers in 24 hours. By using GeoIP utilities. High rate of addresses of these monitored servers has been translated into country code.

Table 4 lists the top 15 countries where old versions of DNS are deployed. Servers of which version is BIND 4.x and 8.x are widely deployed in countries whereas the number of servers of other DNS versions is relatively large. There are clear different of deployment situation between BIND 4.x and 8.x It is turned out that advanced IT countries do not always used the latest or relative secure versions of DNS software, which means that these countries do have a potential weakest links.

Besides, we have generated two time series of receiving time of query response in 24 hours. Figure 2 and 3 show the number of DNS servers of which response is "BIND 4.x" and "BIND 8.x" detected in every one hour. Although the monitoring results much depends on the order of IP address we have crawled, we have obtained more than 70% of answers from DNS servers in first 5 hours. In the view of security assessment, five hours is long enough for attackers to achieve large scale exploitation and too short to take some countermeasure over boarder control on defensive side.

5 Related Work

The studies in early phase have been presented by Danzig et al. [1] and Jung et al. [2]. These two studies analyze lookup behavior from a single local resolver at the vantage point. Recently there has been many research efforts for measuring and analyzing DNS resource records. Previous researches can be classified into three

BIND 4.x		BIND 8.x		DNSMASQ	
BIND-4.9.4	432	BIND-8.4.7-REL	3240	DNSMASQ-2.5.2	379825
BIND-4.9.11	170	BIND-8.3.7-REL	2046	DNSMASQ-2.4.0	323875
BIND-4.9.7	112	BIND-8.3.4-REL	1741	DNSMASQ-2.5.1	199834
BIND-4.9.6-REL	74	BIND-8.2.3-REL	1449	DNSMASQ-2.4.8	103819
BIND-4.9.8-REL	60	BIND-8.2.4-REL	878	DNSMASQ-2.5.5	97604
BIND-4.9.11-REL	55	BIND-8.4.4	815	DNSMASQ-2.1.5	78197
BIND-4.2.7331	53	BIND-8.4.6-REL	615	DNSMASQ-2.1.5	24958
BIND-4.9.7-REL	42	BIND-8.2.2-P7	462	DNSMASQ-2.3.8	17861
BIND-4.9.1	32	BIND-8.4.7	462	DNSMASQ-2.3.6	15411
BIND-4.8.1	32	BIND-8.3.6-REL	462	DNSMASQ-2.6.1	15239
BIND-4.9.3-P1	23	BIND-8.2	275	DNSMASQ-2.2.3	14860
BIND-4.9.4-P1	23	BIND-8.2.7-REL	430	DNSMASQ-2.6.1	15239
BIND-4.9.3	21	BIND-8.4.x	344	DNSMASQ-2.2.3	14860
BIND-4.8.3	10	BIND-8.1.2	82	DNSMASQ-2.4.7	13260
BIND-4.0.1	10	BIND-8.3.1-REL	229	DNSMASQ-2.5.9	8219

Table 3. TOP 15 software versions of BIND 4.x, 8.x and DNSMASQ

#1 China	1055973	#6 Japan	167571	#11 Canada	50553
#2 Taiwan	551156	#7 Thailand	98184	#12 Hong Kong	47005
#3 United States	426306	#8 India	82341	#13 Russia	46560
#4 South Korea	363363	#9 Australia	57174	#14 Germany	37971
#5 Brazil	186218	#10 Romania	50872	#15 England	90083

Table 4. TOP 15 countries which has DNS serevers

BIND 4.x		BIND 8.x	
England	434	United States	3515
United States	224	Russia	3235
Germany	89	Japan	1780
Sweden	48	Germany	1397
Denmark	43	India	563
Itary	39	Puerto Rico	447
Japan	38	France	388
Russia	32	China	347
Canada	18	Taiwan	341
Hungry	15	Canada	318
China	11	England	275
Brazil	10	Poland	263
Poland	10	Itary	225
Urkrine	6	Ukraine	219
Mexico	6	Brazil	185

Table 5. TOP 15 countries where BIND 4.x, 8.x are deployed.

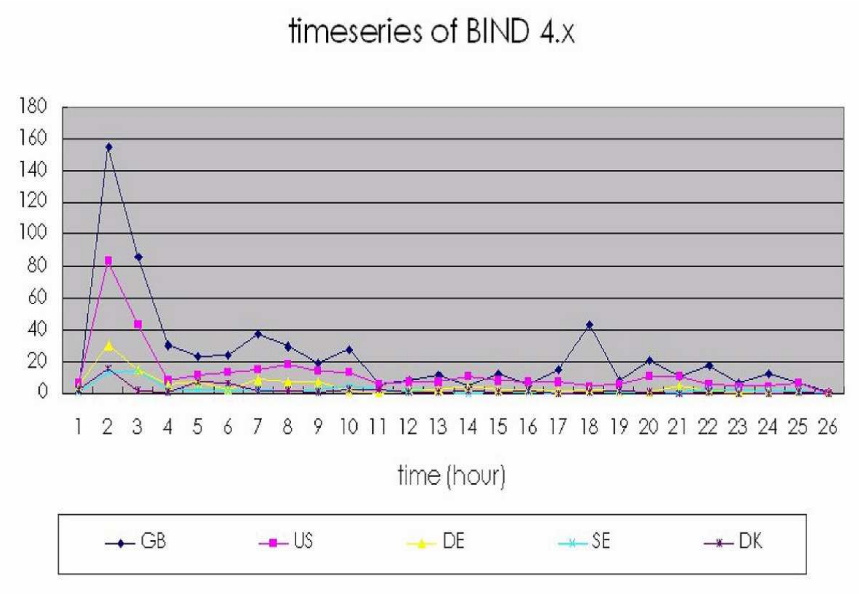

Fig. 3. the number of BIND 4.x servers detected every one hour

categories: domain registration inference, DNS lookup behavior and monitoring on zones' resource records.

Sprint et al. [2] analyze the delay between the response of the first successfully resolved traffic and one from a malicious domain. They present some patterns by correlating data from registries for several top-level domains and a large scale passive DNS data source. Felegyhazi et al [3] examine the potential of leveraging properties of domain registrations and the response in DNS Zone files in order to detecting the malicious use of domains proactively.

Notos[4] and EXPOSURE[5] was designed for building domain's reputation by looking up behavior within a local domain under the DNS resolvers. M. Antonakakis[6] et al. proposed a dynamic reputation system of DNS based on the assumption that malicious and agile use of DNS has unique characteristics. They examine the DNS traffic of 1.4 million users in a large ISP network. EXPOSURE is a large scale passive DNS data analysis system for detecting malicious domains which is resolved by botnets. In [6], they employ 15 features which are extracted from the DNS traffic. They analyze real-world data set of 100 billion DNS requests for identifying unknown malicious domains.

For queries and responses, DNS resource records (RRs) is used for retrieving the characteristics of a zone. monitoring and analysis on zones' resource records previous studies have used the mechanism of querying the DNS servers to check the zones' resource records. DNS resource record is employed for retrieving the zones' feature by measuring and alayzing queries and responses. Holz et al. [8] reveal the technique of employing DNS to establish a proxy network on compro-

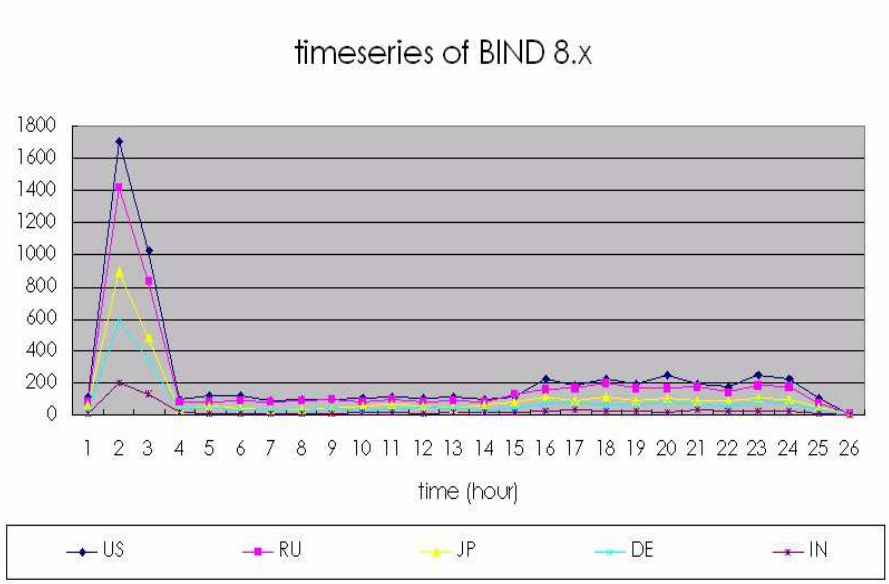

Fig. 4. the number of BIND 8.x servers detected every one hour

mised machines for exploiting illegal online services. This exploitation is called as FFSNs (fast-flux service networks). M. Antonakakis[9] et al propose Anax which scans the recursive servers to detect the anomaly cache records for detect poisoning attacks. This is based on the fact that fast poisoning attacks against DNS servers allows attackers to change records in open recursive DNS servers. They have collected cache changes in a geographically-diverse set of 300,000 open recursive DNS servers.

Related topics of crawling and revealing networks are SSL and DHT. M. Antonakakis et al. performs the large scale network survey of TLS and SSH servers [9]. They adopt Libevent for large scale monitoring of vulnerable keys. C. Zhang [10] et al. implemented asynchronous I/O based crawler for revealing ecosystem of BitTorrent network which consists of tracker and mainline DHT. They present TOP 20 tracker organization ranked by the tracker peers about each country.

6 Discussion

Recently, vulnerable DNS servers have been frequently exploited for establishing malicious domains. Also, flaw of DNS configurations allow attackers to operate scams and spam campaigns. From the result of monitoring of 10,334,293 servers, there have been more than 20,000 servers which could be easily compromised by attackers. In current situation we have revealed, malicious domain registration

and a proxy networks can be easily done by exploiting these servers of which version is obsolete. Also, we have examined time series of query response for BIND 4.x, 8.x and DNSMASQ. It is turned out that more than 60% of servers of old versions have been found in 6 hours. It is partly implied from the result that it is not easy to take countermeasures if attacker begin to exploit these vulnerable servers in the early phase. Because the early detection over border control of several countries might take more long time, at least more than 6 hours. Furthermore, there has not been consensus yet about whether monitoring and warning the servers abroad as mitigation is always recommended or not. For example, although many research effort has been presented about detecting malicious domains registered in DNS, they have no consensus about the mitigation and active (and sometimes positive) monitoring DNS servers without any conditions.

7 Conclusion

In this paper we have presented large scale vulnerability assessment of DNS servers running all over the world. In spite of its importance, there have not been few research efforts on providing the comprehensive view and the detailed report about deployment of DNS servers in real-world networks.

For coping with this situation, we have implemented the asynchronous I/O based crawling system working with MongoDB which is document-based scalable NoSQL. Employing Libevent and MongoDB have enabled us to obtain successfully information of 10,334,293 DNS servers in 24 hours.

Our contributions are classified into three points.

First, our findings reveal that more than 20,000 servers is still using obsolete (and therefore vulnerable) DNS software versions. Particularly, more than 1000 servers still uses the oldest version of BIND 4.x. According to this fact, we can conclude that current situation DNS deployments in Internet do has high risk Second, we have presented some statistics and ranking concerning software versions and its geographical distribution. Third, we show in-depth study of geographical distribution of vulnerable DNS servers with time series analysis.

Lessons from this experiment and monitoring results are classified into two points. First, even if the country achieving high security level do has a potential security risk which could be weakest link.

Second, large scale information gathering and exploitation of vulnerable DNS servers could be easily achieved with relatively low cost implementation and more importantly, in short time. Actually, in current situation, several hours could be not enough for taking countermeasures beyond border controls.

For further work, our system can be applied for more detailed security analysis and evaluation of some DNS exploitation such as malicious domain registration and fast-flux network service. Proposed method can leverage previous research effort for offering the potential for discovery of malicious domains on initial DNS behavior on the early phase. Also, DNS cache poisoning such as

Kaminsky attack can be evaluated corresponding the global distribution of particular BIND versions.

References

1. [1] P. Danzig, K. Obraczka, and A. Kumar. An Analysis of Wide-Area Name Server Traffic: A Study of the Internet Domain Name System. ACM SIGCOMM Computer Communication Review, 22(4):292, Oct. 1992.
[2] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and the Effectiveness of Caching. In Proc. ACM SIGCOMM Internet Measurement Workshop, San Fransisco, CA, Nov. 2001.
[3] J. M. Spring, L. B. Metcalf, and E. Stoner. Correlating Domain Registrations and DNS First Activity in General and for Malware. In Proc. Securing and Trusting Internet Names (SATIN), Teddington, United Kingdom, Apr. 2011.
[4] M. Felegyhazi, C. Kreibich, and V. Paxson. On the Potential of Proactive Domain Blacklisting. In Proc. 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), San Jose, CA, Apr. 2010.
[5] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a Dynamic Reputation System for DNS. In Proc. 19th USENIX Security Symposium, Washington, DC, Aug. 2010.
[6] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In Proc. 18th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, Feb. 2011.
[7] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling. Measuring and Detecting Fast-Flux Service Networks. In Proc. 16th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, Feb. 2008.
[8] M. Antonakakis, D. Dagon, X. Luo, R. Perdisci, W. Lee, and J. Bellmor. A Centralized Monitoring Infrastructure for Improving DNS Security. In Proc. 13th International Symposium on Recent Advances in Intrusion Detection (RAID), Ottawa, Ontario, Canada, Sept. 2010.
[9] N. Heninger, Z. Durumeric, E. Wusraw, J. A. Halderman, Mining your Ps and Qs: detection of widespread weak keys in network devices, In Proc. the 21st USENIX conference on Security symposium 2012
[10] C. Zhang, P. Dhungel, D. Wu Sun Yat-Sen, K. W. Ross, Unraveling the BitTorrent Ecosystem, IEEE Transactions on Parallel and Distributed Systems 2011.
[11] B. Ager, W. Muhlbauer ETH Zurich, G. Smaragdakis, S. Uhlig, Comparing DNS resolvers in the wild, IMC '10 Proceedings of the 10th ACM SIGCOMM conference on Internet measurement.
[12] P. Vixie. DNS Complexity. ACM Queue, 5(3):24?29,2007.
[13] P. Vixie. What DNS is Not. Commun. ACM,52(12):43?47, 2009.
[14] Libevent: a event notification library
http://libevent.org/
[15] MongoDB: open-source document and leading NoSQL database
http://www.mongodb.org/
[16] BIND vulnerability statistics: cvedetails
http://www.cvedetails.com/product/144/ISC-Bind.html?vendor_id=64