IPSJ Journal Vol. 50 No. 9 2163–2172 (Sep. 2009)

Regular Paper

IP Traceback Using DNS Logs against Bots

Keisuke Takemori,[†1] Masahiko Fujinaga,[†1]
Toshiya Sayama[†1] and Masakatsu Nishigaki[†2]

Recently, source IP spoofing attacks are critical issues for the Internet. These attacks are considered to be sent from bot infected hosts. There has been active research on IP traceback technologies. However, the traceback from an end victim host to an end spoofing host has not yet been achieved, due to the lack of traceback probes installed on each routing path. Alternative probes should be employed in order to reduce the installation cost. In this research, we propose an IP traceback scheme against bots using DNS logs of existing servers. Many types of bots retrieve IP addresses of victim hosts from fully qualified domain names (FQDNs) at the beginning of an attack. The proposed scheme checks from the destination to the source DNS logs, in order to extract the actual IP addresses of bot infected hosts. Also, we propose a scheme to ascertain the reliability of traceback results, and a method to distinguish spoofing from non-spoofing attacks. We collect bot communication patterns to confirm that the DNS log can be used for reasonable probes and for achieving a high traceback success rate.

1. Introduction

Recently, source IP spoofing attacks are critical issues for the Internet. These attacks are considered to be sent from bot infected hosts that are controlled via command-and-control (C&C) servers. There has been active research on IP traceback systems. For example, a method involving ICMP traceback messages[1] that sends information concerning a spoofed packet, a packet marking method[2],[3] that fills router identification in a packet header, and a digest method[4] that records and retrieves the hash values of packets on each router have been proposed. However, traceback from an end victim host to an end spoofing host has not yet been achieved, due to the lack of traceback probes installed on each routing path. In a serial traceback scheme, the end-to-end traceback success rate may decrease with the cumulative power order of the failure rate on each routing hop[5]. Many probes must be installed on each domain in order to achieve a high end-to-end traceback success rate. Therefore, it is necessary to employ them with alternative probes in order to reduce the installation cost.

In this research, we propose an IP traceback scheme against bots using logs from existing DNS servers. Because many types of bots retrieve IP addresses of victim hosts from their FQDNs at the beginning of an attack, we can track bots from the DNS query logs. The proposed scheme checks from the destination DNS to the source DNS (generally called a resolver) logs, in order to extract the IP addresses of the bot infected hosts. Also, we consider how to obtain reliable traceback results, involving the matching of a few DNS logs and the extraction of common IP addresses of the bot infected hosts. Furthermore, we propose methods to distinguish spoofing from non-spoofing attacks, which can retain the privacy of a communication pair from non-participant domains. We collect bot communication patterns to confirm that the DNS log can be used for reasonable probes and for achieving a high traceback success rate.

The remainder of this paper is organized as follows. Section 2 presents a survey of previous IP traceback works. Section 3 investigates bot communication patterns, especially DNS queries. Section 4 proposes an IP traceback schemes against IP spoofing attacks from bots using DNS query logs. Section 5 evaluates the end-to-end traceback success rate. Section 6 explains further studies. And finally, Section 7 concludes the paper.

2. Previous Works

In this section, we survey conventional IP traceback schemes, namely the ICMP traceback, the packet marking method, and the hash-based traceback.

In the case of the ICMP traceback[1], the probe samples packets and calculates a packet digest that is sent to the destination host with the probe identification using an ICMP packet. The destination host receives the ICMP packets and depicts the routing path according to the digests and the probe identifications. The disadvantages of the ICMP traceback scheme are as follows. Many probes need to be installed on the Internet, many digests and probe identifications must

[†1] KDDI R&D Laboratories
[†2] Shizuoka University, Graduate School of Science and Technology

be collected to depict the routing path, and the ICMP traceback packets load to the network bandwidth.

In the case of the packet marking method[2),3)], the probe identification is inserted in the packet header area that is not used for the Internet routing. The destination host receives the packets and extracts the probe identification to depict the routing path. The disadvantages of the packet marking scheme are as follows. A large number of probes need to be installed on the Internet, many probe identities need to be collected, and many insertion conflicts occur with other probes due to the limitation of the unused header area.

In the case of the hash-based traceback[4)], the probe records the digest of the packet. When the destination host requests a traceback, the probes on the routing path retrieve the digests. If the same digest is retrieved, the packet is exchanged on the probe. The disadvantages of the hash-based scheme are as follows: a lot of probes must be installed on the Internet and the capacity of a probe HDD needs to be considerable in order to record the packet digests for as long as possible.

3. Investigation of Bot Communication Patterns

In this section, we investigate actual bot communication patterns, especially DNS queries.

Figure 1 shows models of a botnet topology and its communication pattern. The bots receive control packets from C&C servers to attack the victim host. Next, each bot sends a DNS query to a DNS server to retrieve the IP address of the victim host using its FQDN. Then, the attack packets are sent to the victim host from the bot.

We have collected 44 kinds of bot code using the honeypot[6)] and infected a virtual machine[7)]. Thirty-seven kinds of bot communicated with other hosts, while 7 kinds of bot were not active on the virtual machine. The details of the attack communication patterns are as follows; 37 bot codes communicated with the C&C servers, 15 bot codes sent spam mails, 14 bot codes scanned IP addresses, 13 bot codes sent DoS packets to port 135–139, and 2 bot codes tried SSH dictionary attacks. Because some of the bot codes communicated with multiple attack patterns, the total number of attack patterns is 81. These activities were monitored at an outbound firewall that allows the DNS and C&C packets and denies mail, ICMP, DoS, and dictionary attack packets from the bot. The outbound firewall achieves a safety LAN that blocks the leakage of attack packets to the Internet.

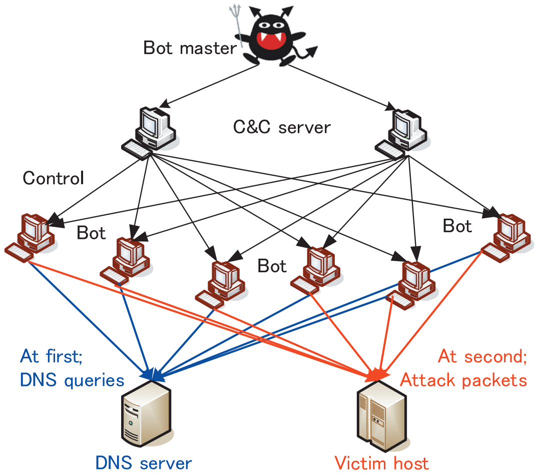
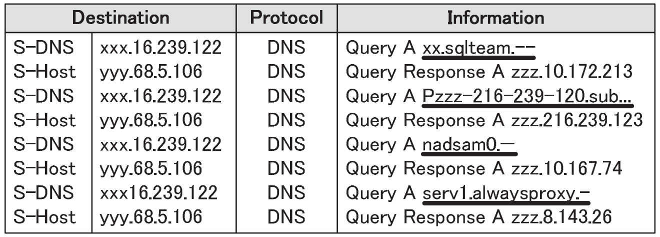


Fig. 1 Model of the botnet and communication pattern.

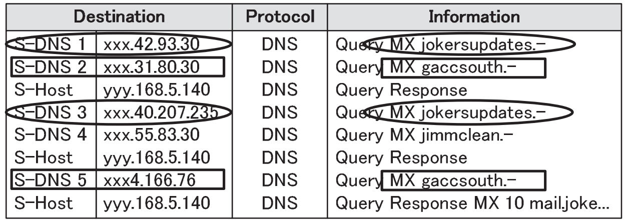
Figure 2 shows an example of a bot communication pattern with a primary DNS server, where DNS queries between the bot and the primary DNS server are extracted[8)]. The bot sent recursive DNS queries that retrieved 4 kinds of FQDN, which included both the victim hosts and the C&C servers.

Figure 3 shows an example of a DNS query pattern from a spam-mail bot. The spam-mail bot turned into a DNS resolver and sent DNS queries to retrieve the MX records of each domain directly. **Figure 4** shows a screenshot of the communication pattern visualizer that depicts the communication pattern between the bot and the DNS servers shown in Fig. 3. Figure 4 shows communication patterns consisting of 10 axes that represent 1 axis of the bot ports, 4 axes of octet sub-IP addresses of the bot, 4 axes of octet sub-IP addresses of the DNS server, and 1 axis of the DNS server ports. The spam-mail bot accessed many domain DNS servers in order to retrieve the MX records. The left side is the bot turned into the DNS resolver that sends DNS queries to many outside DNS servers depicted right side.



Destination		Protocol	Information
S−DNS	xxx.16.239.122	DNS	Query A xx.sqlteam.−−
S−Host	yyy.68.5.106	DNS	Query Response A zzz.10.172.213
S−DNS	xxx.16.239.122	DNS	Query A Pzzz−216−239−120.sub...
S−Host	yyy.68.5.106	DNS	Query Response A zzz.216.239.123
S−DNS	xxx.16.239.122	DNS	Query A nadsam0.−
S−Host	yyy.68.5.106	DNS	Query Response A zzz.10.167.74
S−DNS	xxx16.239.122	DNS	Query A serv1.alwaysproxy.−
S−Host	yyy.68.5.106	DNS	Query Response A zzz.8.143.26

Fig. 2 DNS queries between an attacker and a primary DNS server.

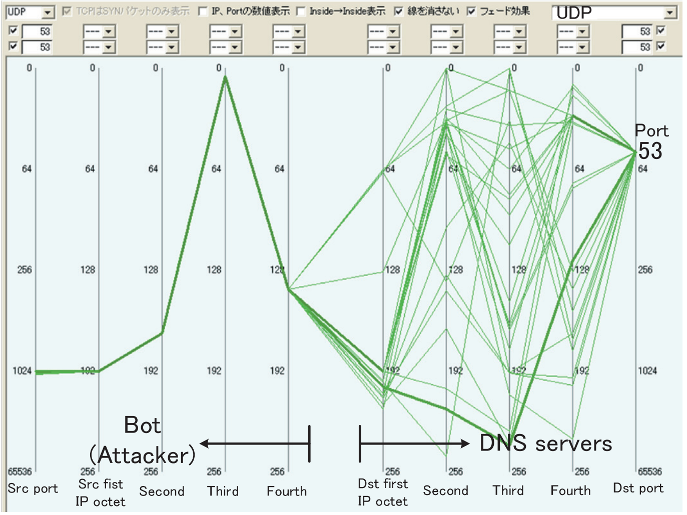


Destination		Protocol	Information
S−DNS 1	xxx.42.93.30	DNS	Query MX jokersupdates.−
S−DNS 2	xxx.31.80.30	DNS	Query MX gaccsouth.−
S−Host	yyy.168.5.140	DNS	Query Response
S−DNS 3	xxx.40.207.235	DNS	Query MX jokersupdates.−
S−DNS 4	xxx.55.83.30	DNS	Query MX jimmclean.−
S−Host	yyy.168.5.140	DNS	Query Response
S−DNS 5	xxx4.166.76	DNS	Query MX gaccsouth.−
S−Host	yyy.168.5.140	DNS	Query Response MX 10 mail.joke...

Fig. 3 DNS queries between an attacker and DNS servers.

Table 1 DNS query ratio of retrieving the victim hosts and the C&C servers.

	Communications	IP address resolution
Active bot (All)	37 bots	-
Packet to victim hosts	29 bots	55% (16/29)
Packet to C&C servers	37 bots	100% (37/37)



UDP ☑ TCPはSYNパケットのみ表示 □ IP、Portの数値表示 □ Inside→Inside表示 ☑ 線を消さない ☑ フェード効果 UDP

☑ 53 ☑ 53 | 53 ☑ 53 ☑

Fig. 4 DNS query pattern between the bot and the DNS servers (shown in Fig. 3).

Table 1 shows the DNS query ratio of retrieving the IP addresses of the victim hosts and the C&C servers. Twenty-nine kinds of bot attacked victim hosts, and 16 kinds of the 29 bots sent DNS queries in order to resolve the IP addresses of the victim hosts, while all 37 kinds of bot sent DNS queries to resolve the IP addresses of the C&C servers. IP addresses of the DNS query packets are never spoofed in order to get the DNS query response. After the IP address resolution, the bot communicates with the victim hosts and the C&C servers.

4. IP Traceback Using DNS Logs against Bots

In this section, we propose an IP traceback scheme against bots, including source IP spoofing attacks, by using DNS logs. Our model tracks the bot by collaborating with a source DNS server that is not limited to a primary or a secondary DNS server of the bot. One assumption of our scheme is that the attacker retrieves the IP address from the DNS server before sending spoofed packets.

4.1 Review of the DNS Query Model

We review the typical DNS query model shown in **Fig. 5**. The source host sends a recursive query packet to a source DNS server in order to retrieve the IP address of the FQDN. The source DNS server will be a resolver and resolves the FQDN by traversing a DNS tree.

I A source host sends a recursive query *www.example.com* to a source DNS server that will be a resolver.

II The source DNS server sends iterative queries *.com* to the root DNS, *example.com* to the region DNS, and *www.example.com* to the destination DNS servers.

III The destination DNS server replies the IP address *xxx.xxx.xxx.xxx* of the destination FQDN *www.example.com* to the source DNS server.

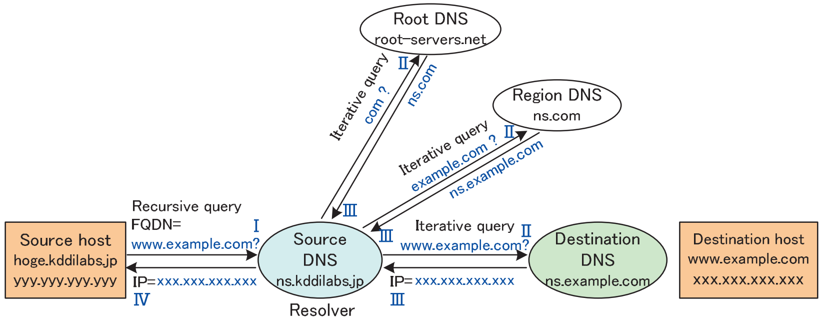
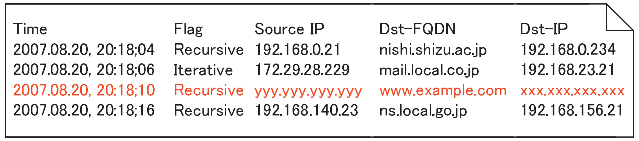


Fig. 5 Typical DNS query model.



```
Time                  Flag       Source IP        Dst-FQDN           Dst-IP
2007.08.20, 20:18;04  Recursive  192.168.0.21     nishi.shizu.ac.jp  192.168.0.234
2007.08.20, 20:18;06  Iterative  172.29.28.229    mail.local.co.jp   192.168.23.21
2007.08.20, 20:18;10  Recursive  yyy.yyy.yyy.yyy  www.example.com    xxx.xxx.xxx.xxx
2007.08.20, 20:18;16  Recursive  192.168.140.23   ns.local.go.jp     192.168.156.21
```

Fig. 6 Example of a DNS log.

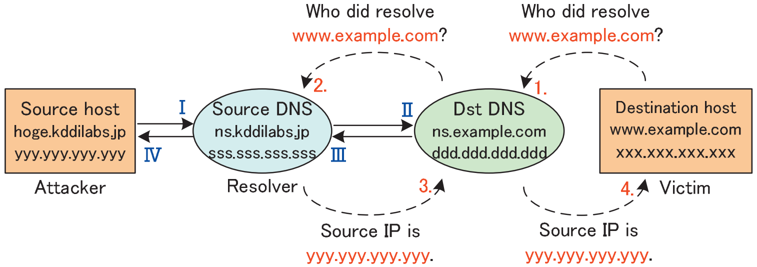
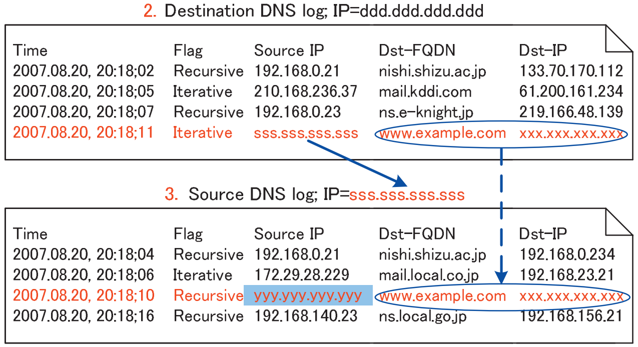


Fig. 7 IP traceback scheme using source and destination DNS logs.



2. Destination DNS log; IP=ddd.ddd.ddd.ddd

```
Time                  Flag       Source IP        Dst-FQDN           Dst-IP
2007.08.20, 20:18;02  Recursive  192.168.0.21     nishi.shizu.ac.jp  133.70.170.112
2007.08.20, 20:18;05  Iterative  210.168.236.37   mail.kddi.com      61.200.161.234
2007.08.20, 20:18;07  Recursive  192.168.0.23     ns.e-knight.jp     219.166.48.139
2007.08.20, 20:18;11  Iterative  sss.sss.sss.sss  www.example.com    xxx.xxx.xxx.xxx
```

3. Source DNS log; IP=sss.sss.sss.sss

```
Time                  Flag       Source IP        Dst-FQDN           Dst-IP
2007.08.20, 20:18;04  Recursive  192.168.0.21     nishi.shizu.ac.jp  192.168.0.234
2007.08.20, 20:18;06  Iterative  172.29.28.229    mail.local.co.jp   192.168.23.21
2007.08.20, 20:18;10  Recursive  yyy.yyy.yyy.yyy  www.example.com    xxx.xxx.xxx.xxx
2007.08.20, 20:18;16  Recursive  192.168.140.23   ns.local.go.jp     192.168.156.21
```

Fig. 8 Matching between destination and source DNS events.

IV The source DNS server responds with the resolved IP address *xxx.xxx.xxx.xxx* to the source host.

Figure 6 shows an example of the source DNS log. The log records the IP address of the source host linked to the destination FQDN.

4.2 IP Traceback for Basic Recursive DNS Queries

We propose an IP traceback scheme whereby the source DNS server cooperates with the destination DNS server as shown in **Fig. 7** and **Fig. 8**. Here, the root and region DNS servers are abbreviated. Tags "I, II, III, and IV" represent the same procedures as shown in Fig. 5, while the proposed IP traceback procedures are as follows:

1. When the destination host receives the source IP spoofed packets, an IP traceback request including the destination FQDN *www.example.com* and attack time information are sent from the destination host to the destination DNS server.
2. The destination DNS server inspects its DNS events in order to extract the IP address of the source DNS server who resolves the *www.example.com*. Moreover, the IP traceback request is relayed from the destination to the source DNS server.
3. The source DNS server inspects its recursive DNS events to extract the source host who resolves the FQDN *www.example.com*. If the IP address of the source host *yyy.yyy.yyy.yyy* is extracted, the source DNS server replies it to the destination DNS server.
4. The destination DNS server relays the IP address of the source host *yyy.yyy.yyy.yyy* to the destination host.

4.3 IP Traceback for Forwarding DNS Queries

Several source DNS servers are configured for DNS forwarding[9]. A forwarder DNS server for the source DNS server will be a DNS resolver. In this case,

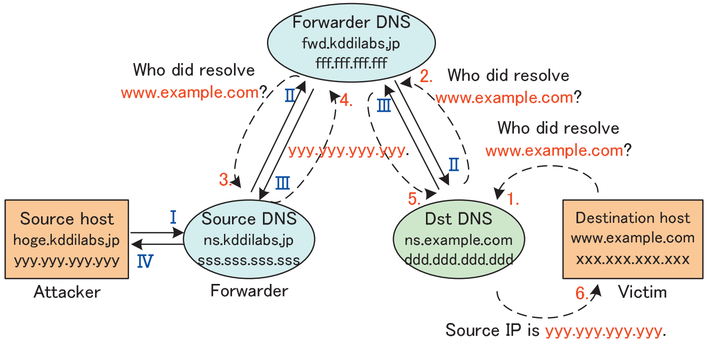


Fig. 9 Multi-hop DNS query traceback under DNS forwarding mode.

the DNS query log of the destination DNS server records the IP address of the forwarder DNS server instead of the source DNS server. Thus, it is necessary to track additional hops to the source DNS server. **Figure 9** shows the traceback model using three DNS server logs.

1. The same procedures as Step 1 in Fig. 7.
2. The same procedures as Step 2 in Fig. 7.
   The IP traceback request is relayed to the forwarder DNS server.
3. The forwarder DNS server inspects its DNS events to extract the IP address of the source DNS server and relays the IP traceback request.
4. The same procedures as Step 3 in Fig. 7.
   The IP traceback result is replied to the forwarder DNS server.
5. The forwarder DNS server relays the IP traceback result to the destination DNS server.
6. The same procedures as Step 4 in Fig. 7.

Following this section, we do not consider the DNS forwarding model to focus on the basic schemes.

4.4 Reduction of False Positives

When many source hosts retrieve the FQDN of the destination host simultaneously, it is difficult to distinguish between legitimate hosts and attackers. With this in mind, we consider how to reduce false positives of source IP traceback. Approaches to false positive reduction are applied at both the source and the destination DNS servers.

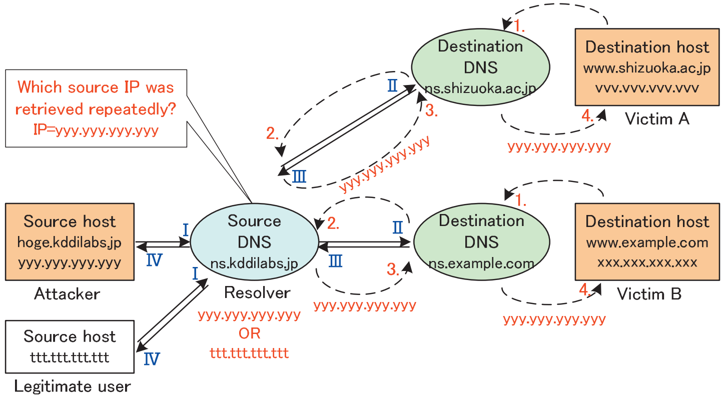


Fig. 10 Extracting the attacker IP address at the source DNS server.

4.4.1 Extracting the Attacker at the Source DNS Server

It is often observed that a bot retrieves the FQDNs of many victims as shown in Fig. 2. When these kinds of attacks are sent, the source DNS server receives a lot of IP traceback requests regarding a certain IP from many destination DNS servers. In other words, if the source DNS log includes a common source host, the IP address is considered to be the attacker. **Figure 10** shows the false reduction model at the source DNS server.

1. When certain destination hosts receive source IP spoofed packets, IP traceback requests are sent from the destination hosts to the destination DNS servers.
2. The destination DNS servers inspect their DNS events in order to extract the source DNS servers who resolve the victim FQDNs. IP traceback requests are relayed to the source DNS servers.
3. The source DNS server receives certain IP traceback requests from many destination DNS servers; extracts the common IP address of the source host *yyy.yyy.yyy.yyy*, and replies it, as the presumed attacker, to the destination DNS servers.
4. The destination DNS servers relay the IP address of the source host *yyy.yyy.yyy.yyy* to the destination hosts.

4.4.2 Extracting the Attacker at the Destination DNS Server

It is often observed that a bot retrieves the IP address of a victim with many DNS queries as shown in Fig. 3. When these kinds of attack are sent, the destination DNS server receives common DNS queries from many source DNS servers. In other words, if the IP traceback results replied to the destination DNS server include the common source host, the source IP address is considered to be the attacker. **Figure 11** shows the false reduction model at the destination DNS server.

1. When the destination host receives source IP spoofed packets, an IP traceback request is sent from the destination host to the destination DNS server.
2. The destination DNS server inspects its DNS events in order to extract the source DNS servers who resolve the victim FQDN. Moreover, IP traceback requests are relayed from the destination DNS server to some source DNS servers.
3. The source DNS servers inspect their recursive DNS event in order to extract the source host *yyy.yyy.yyy.yyy*, and reply it to the destination DNS server.
4. The destination DNS server receives the IP traceback results, extracts the common IP address of the source host *yyy.yyy.yyy.yyy*, and relays it, as the presumed attacker, to the destination host.

4.5 Spoofing/Non-spoofing Confirmation while Keeping Communication Privacy

The traceback scheme can be executed only when we have an evidence of a spoofing attack, because the source host violates the IP communication protocols of the Internet. As a spoofing attack damages other legitimate communication services, a network operator should filter spoofed packets as soon as possible. The spoofing/non-spoofing confirmation is applied before the traceback schemes shown in above subsections.

We propose a confirmation procedure capable of distinguishing a spoofing from a non-spoofed packet while preserving communication privacy from non-participant domains. Here, communication privacy means the IP addresses of the communication pair. With this in mind, we use the hash value to conceal the IP address: to resist a brute-force attack, the hash value is calculated from not only the source IP address but also the destination FQDN.

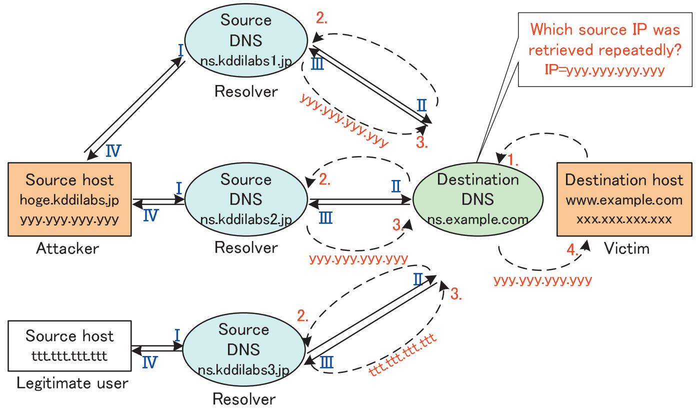


Fig. 11 Extracting the attacker IP address at the destination DNS server.

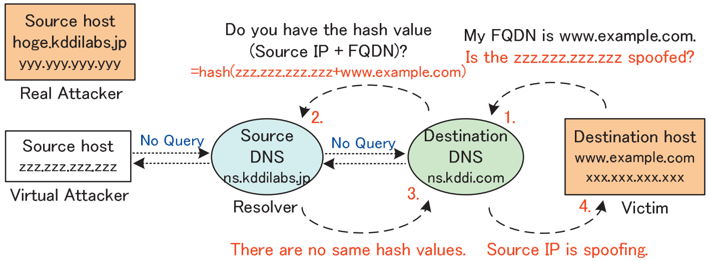


Fig. 12 IP spoofing confirmation while retaining the privacy of the communication pair.

The destination host calculates the hash value using the source IP address of the attack packet and the destination FQDN. A source domain DNS server calculates the hash value using the source IP address and the destination FQDN in the DNS log. In the case of a non-spoofing attack, the source DNS server receiving the DNS query from the attacker can calculate the same hash value using the source IP address of the attacker and the destination FQDN. On the other hand, in the case of a spoofing attack, the source DNS server is unable to calculate the same hash value using the source IP address of the DNS query and the destination FQDN. **Figure 12** shows a procedure of spoofing/non-spoofing confirmation, while preserving the communication privacy.

1. When the destination host receives the attack packets, the hash value calculated from the source IP address of the attack packet *zzz.zzz.zzz.zzz* and destination FQDN *www.example.com* is sent to the destination DNS server as a confirmation request.
2. The destination DNS server inspects its DNS events to extract the source DNS server, which resolves *www.example.com*. Subsequently, the hash value calculated from *zzz.zzz.zzz.zzz* and *www.example.com*, is sent to the source DNS servers as a confirmation request.
3. The source DNS server calculates the hash value using its recursive DNS events in the DNS log.
   If the same hash value can be calculated, the source IP address of the attack packet is considered to be non-spoofing. If the same hash value cannot be calculated, the source IP address of the attack packet is considered to be spoofed. The confirmation result is replied to the destination DNS server.
4. The destination DNS server relays the confirmation result to the destination host.

5. Evaluation

In this section, we evaluate the end-to-end traceback success rate and the recording load of the DNS server log.

5.1 End-to-end Traceback Success Rate

First, we investigated end-to-end routing path hops on the Internet [10),11)] and estimated the average end-to-end routing path hops to be 15 as of May 2008.

In the case of a conventional IP traceback scheme, the end-to-end traceback success rate is calculated as the power of the success rate per hop [5)]. Reference 5) shows the theoretical approach depending on the success rate per hop. Here, the success rate per hop means the probe installation rate on each router. In the case of the proposed IP traceback scheme, the end-to-end traceback success rate is calculated as the square of the success rate per hop times the DNS query rate of the bot. **Figure 13** shows the end-to-end traceback success rate versus the traceback hop length. In this figure, the success rate per hop is $p = 0.9$. The end-to-end traceback success rate of the conventional scheme decreases quickly, because the rate is followed by the power of the success rate per hop. On the other hand, the end-to-end traceback success rates of the proposed scheme are constant values at more than 2 hops. At 15 hops, the end-to-end traceback success rates of the conventional scheme, the proposed scheme with the DNS query rate = 0.55, and the proposed scheme with the DNS query rate = 1.00 are about 0.20, 0.45, and 0.81, respectively. The proposed schemes can achieve a higher end-to-end success rate than the conventional scheme at more than 3 or 8 routing hops.

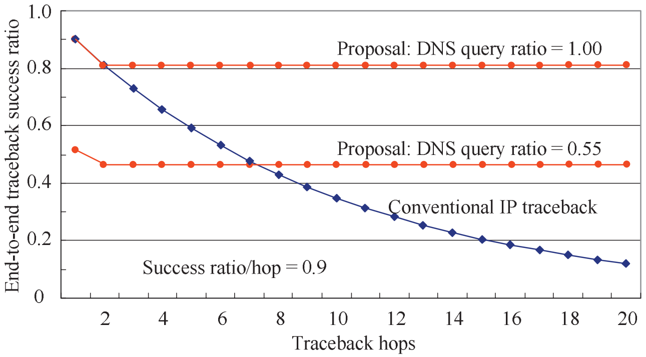


Fig. 13 End-to-end traceback success rate versus number of traceback hops.

5.2 Output Load of the DNS Log

The proposed scheme can track bots when the DNS servers output the DNS logs. Thus, we evaluated the CPU load of the DNS server with and without the DNS log output.

Figure 14 shows the CPU load of the DNS server, with the server specifications as follows: Intel Xeon 3.6-GHz dual-CPU, 4.0-Gbyte memory, Linux 2.4.21-27.0.1.ELsmp, and a BIND 9.2.4-5 EL3 DNS server. We use a traffic generator capable of emulating DNS queries. All the queries from the traffic generator are matched with a DNS cache on the server, while the average CPU load is monitored by the *vmstat* command for 30 seconds. The CPU loads with the outputting DNS log are about 1% and 2% larger than the loads without the outputting DNS log at 100 and 1,000 queries/sec, respectively. This means that the CPU load of the source and destination DNS servers will not be a critical problem when the DNS query log is output. It is easy to apply the IP traceback scheme using a DNS log to the Internet.

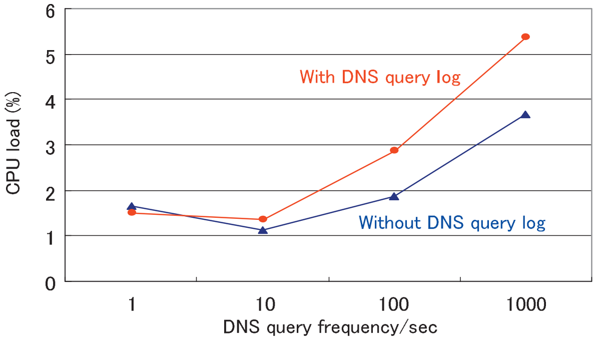


Fig. 14 CPU load of DNS server versus DNS query frequency.

6. Considerations and Further Studies

6.1 False Positives and False Negatives

The proposed DNS log traceback includes false positives and false negatives.

The false positives occur when both a bot and a non-infected host send DNS queries and access the victim site simultaneously. For example, a DNS server that is operated by Communication Laboratory locating famous FTP servers receives about 200,000 DNS queries/day from the Internet. If the tracking time range is set as 5 seconds, at most 12 false positives may occur in this case.

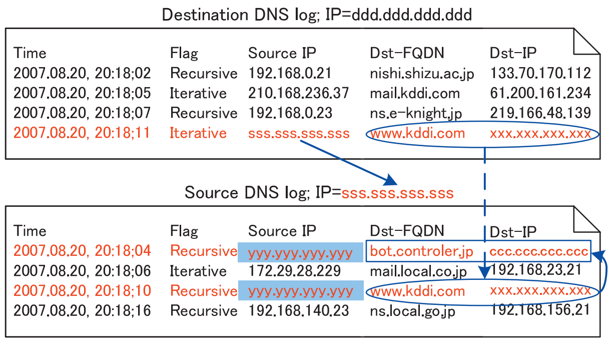
The false negatives occur when a bot accesses a victim host without DNS queries, and when the DNS cache is matched with the source DNS server. The access ratio without DNS queries is presented in Table 1 as 45%. In general, the DNS cache is kept for about 7,200 seconds. When non-infected hosts on the same domain retrieve the victim host, the DNS queries from the bot match with the cache of the source DNS server.

6.2 Discrimination Accuracy between Spoofed and Non-spoofed Attacks

The proposed spoofed/non-spoofed confirmation scheme includes false discrimination of spoofed/non-spoofed attack.

The false discrimination of spoofed attack occurs when the bot accesses the victim host without DNS queries. In this case, as the source DNS server holds no queries from the bot, the false negatives of tracking occur. So the victim host notices the false negatives of tracking instead of the false discrimination of a spoofed attack.

The false discrimination of a non-spoofed attack occurs when non-infected hosts on the spoofed source domain send the DNS queries of the victim host. In this case, as the source DNS server holds legitimate DNS queries from the non-infected hosts, the false positives of tracking occur. So the victim host misjudges that the source IP address is not spoofed and indicates the attacker.



Destination DNS log; IP=ddd.ddd.ddd.ddd

Time	Flag	Source IP	Dst−FQDN	Dst−IP
2007.08.20, 20:18;02	Recursive	192.168.0.21	nishi.shizu.ac.jp	133.70.170.112
2007.08.20, 20:18;05	Iterative	210.168.236.37	mail.kddi.com	61.200.161.234
2007.08.20, 20:18;07	Recursive	192.168.0.23	ns.e−knight.jp	219.166.48.139
2007.08.20, 20:18;11	Iterative	sss.sss.sss.sss	www.kddi.com	xxx.xxx.xxx.xxx

Source DNS log; IP=sss.sss.sss.sss

Time	Flag	Source IP	Dst−FQDN	Dst−IP
2007.08.20, 20:18;04	Recursive	yyy.yyy.yyy.yyy	bot.controler.jp	ccc.ccc.ccc.ccc
2007.08.20, 20:18;06	Iterative	172.29.28.229	mail.local.co.jp	192.168.23.21
2007.08.20, 20:18;10	Recursive	yyy.yyy.yyy.yyy	www.kddi.com	xxx.xxx.xxx.xxx
2007.08.20, 20:18;16	Recursive	192.168.140.23	ns.local.go.jp	192.168.156.21

Fig. 15 IP traceback scheme against the C&C server.

6.3 IP Traceback Using DNS Logs against C&C Servers

In our further study, we consider how to track C&C servers using DNS logs. Table 1 shows that most bots retrieved the IP addresses of the C&C servers, hence the recursive DNS log of the source DNS server records the DNS query used to retrieve the C&C server from the bots.

Figure 15 shows the IP traceback scheme against the C&C servers using the source and destination DNS logs. The procedures involved in the IP traceback request are almost identical to those of the bot traceback schemes, while the inspection procedures of the recursive DNS events at the source DNS server are as follows:

- If the IP address of the bot *yyy.yyy.yyy.yyy* is detected in the source DNS log, the previous DNS query to the outside from the bot IP *yyy.yyy.yyy.yyy* is extracted.
- The destination IP address of the recursive records *ccc.ccc.ccc.ccc* indicates

the C&C server. The source DNS server passes on the previous recursive records to the destination DNS server.

However, if the bot infected host is controlled by the user, various DNS query events are recorded in the source DNS server logs. It is difficult to distinguish between legitimate user and bot queries and a further study is required to ensure a reliable extraction procedure.

6.4 Investigation of the DNS Query Ratio

When a destination FQDN is matched with a cache list of the source DNS server, the DNS query is not exchanged between the source and the destination DNS servers. Our scheme will be unable to track a source host if the DNS query from the source DNS server is not recorded in the destination DNS log. We will investigate the DNS query ratio from the source to the destination DNS servers.

6.5 Communication Protocol between the Victim Host and DNS Servers

The victim host communicates with the destination DNS server, and the destination DNS server also communicates with the source DNS server. The requirements of the communication links are as follows:

- Encrypted connection
- Client authentication
- Server authentication

It is expected that the SSH service is applied to both connections. It is important to consider client authentication strictly, because the DNS log is considered to include communication privacy. A further study of the communication protocol will address who is permitted to send a traceback request.

7. Conclusions

In this research, we proposed an IP traceback scheme against bots using DNS logs. Also, we considered how to distinguish spoofing from non-spoofing attacks, and how to ensure reliable traceback results. Our scheme is easy to apply on the Internet because the DNS server log can be substituted for conventional IP traceback probes. The end-to-end traceback success rate of the proposed scheme is higher than the conventional scheme when the routing path length is considerable. The CPU load of the DNS server that outputs the DNS events is relatively light. Our scheme is able to track IP spoofed packets from bots. In addition, it is expected that the C&C server may also be tracked by extracting the events on the source DNS log.

Acknowledgments This research was supported by the National Institute of Information and Communication Technology (NICT) of Japan. The purpose of the research theme is R&D for traceback systems on the Internet.

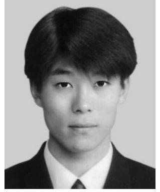
References

1) Bellovin, S., Leech, M. and Taylor, T.: ICMP Traceback Messages, IETF, Internet Draft, draft-ietf-itrace-04.txt (Aug. 2003).
2) Song, D. and Perrig, A.: Advanced and Authenticated Marking Schemes for IP Traceback, *Proc. IEEE Infocom* (Apr. 2001).
3) Yaar, A., Perrig, A. and Song, D.: FIT: Fast Internet Traceback, *Proc. IEEE Infocom* (Apr. 2005).
4) Snoeren, A.C., Partridge, C., Sanchez, L.A., Jones, C.E., Tchakountio, F., Kent, S.T. and Strayer, W.T.: Hash-Based IP Traceback, *Proc. ACM SIGCOM 2001*, San Diego, USA (Oct. 2001).
5) Takemori K. and Endo S.: Performance Analysis of IP Traceback Systems with Serial and Parallel Control Schemes, *IEEE, Proc. PacRim'07*, pp.227–231 (Aug. 2007.)
6) Nepenthes. http://nepenthes.mwcollect.org/
7) VMware. http://www.vmware.com/
8) WireSHARK. http://www.wireshark.org/
9) DNS Forwarding. http://tldp.org/HOWTO/DNS-HOWTO-4.html
10) BGP Routing Table Analysis Report. http://bgp.potaroo.net/as4637/index-bgp.html
11) The CAIDA Web Site. http://www.caida.org/home/

(Received November 28, 2008)

(Accepted June 4, 2009)

(Original version of this article can be found in the Journal of Information Processing Vol.17, pp.232–241.)



Keisuke Takemori received his B.E. and M.E. degree in Electrical Engineering, and Ph.D. in Information and Computer Science from Keio University, Japan in 1994, 1996, and 2004. He joined KDD Corporation in 1996. He is now a research engineer at KDDI R&D Laboratories. His current research interests are in network security and communication network. He is members of IEEE and IEICE.

Masahiko Fujinaga received his B.S. and M.E. in Electrical Engineering from the University of Tokyo in 1982 and 1984. He is now a senior research eingineer of KDDI R&D Laboratories. He has been working in the areas of unix communications, G4 facsimile, distributed processing, routing in the internet and traceback systems. He is a member of IEEE.

Toshiya Sayama received his B.E. in Electrical Engineering and Electronics from Aoyama Gakuin University, Japan in 1981. He joined KDD Corporation in 1981. He is now a senior manager at KDDI R&D Laboratories. His current development interests are in network and mobile phone security.

Masakatsu Nishigaki recieved his Ph.D. in Engineering from Shizuoka University, Japan. He served as a Postdoctoral Research Fellow of the Japan Society for the Promotion of Science in 1995. Since 1996 he has been engaged in research at the Faculty of Infomatics, Shizuoka University. He is now an Associate Professor at the Graduate School of Science and Technology of Shizuoka University. His research interests are in information security, neural network, circuit simulation, etc.