DNS と TCP コネクションに関する挙動の解析

力武 健次 [*][★]　野川 裕記 [★]　菅谷 史昭 [‡]　中尾 康二 [‡]　下條 真司 [★]

DNS（ドメイン名システム）はクライアントがドメイン名解決を行うほとんどの場合参照されるため，その参照のパターンはインターネット上の通信活動に密接に関連している．本稿では TCP コネクションと DNS 参照の間の挙動の解析を行い，解析に必要なデータを収集するためのシステムと，解析結果がどのようにインターネットに接続されたシステムを防衛するために使えるかについて提案する．

Behavioral Analysis of DNS and TCP Connections

Kenji Rikitake[*][‡], Hiroki Nogawa[★], Fumiaki Sugaya[‡], Koji Nakao[‡] and Shinji Shimojo[★]

The patterns of DNS (Domain Name System) references are closely related to the Internet communication activities, since DNS is referred when a client tries to resolve a domain name in most cases. In this paper we analyze the behavioral relationship between DNS references and TCP connections. We also propose a system to collect necessary data for the analysis, and how the analyzed results can be used for the defense of Internet-connected systems.

1 Introduction

Activities on the Internet is highly dependent on the DNS (Domain Name System) [1, 2], since DNS provides a mapping service between the domain names and various resources, such as IP addresses, MX (Mail eXchanger) host names, and various text-based entries.

Internet application services must use DNS to resolve domain names. Looking up the DNS is mandatory for a Web browser to find out the IP address of the server in the given URL (Uniform Resource Locator). Mail transfer agents find out the destination agent by looking up the DNS for the MX RRs (Resource Records) of the destination domain name in the given mail addresses.

When a host provides DNS service, the TCP [3] and UDP [4] port number 53 assigned for DNS must be left wide-open for the public access. This requirement is also applicable to the gateway systems which provides the packet- and content-filtering functionality between the private and public networks, so in the gateway systems a DNS server or a cache must be running to provide the service. This suggests that a DNS system can also be used for monitoring or collecting activity data between the private and public networks.

The correlation between DNS references and activities of the application software suggests that behavioral patterns of DNS references can be used as a clue of finding out the traffic characteristics. Monitoring the DNS traffic patterns helps analyzing and predicting the events on the network, such as a DoS (Denial-of-Service) attack generated inside or coming outside the network firewall.

One of the real-world example of the relationship between the DNS traffic and an activity which we have learned is the fact that the traffic to the DNS server of a domain increased immediately after an e-mail message was sent to a mailing list from the domain. In this case we concluded

[*] 大阪大学大学院情報科学研究科 / Graduate School of Information Science and Technology, Osaka University
rikitake@ist.osaka-u.ac.jp
[‡]（株）KDDI 研究所コンピュータセキュリティグループ / Computer Security Laboratory, KDDI R&D Laboratories, Inc.
{kenji,fsugaya,nakao}@kddilabs.jp
[★] 大阪大学サイバーメディアセンター / Cybermedia Center, Osaka University
{nogawa,shimojo}@cmc.osaka-u.ac.jp

that the traffic was generated by the mail transfer agents which looked up the domain name of the sender to verify whether the domain name did exist and the domain was reachable over Internet.

Another example is found in the papers of Musashi, Sugitani and Matsuba [5, 6], which shows that the number of DNS queries sent from an e-mail server has a strong correlation with the abnormal activities of an e-mail client suspected to be compromised by a computer virus, by comparing the activity logs of the DNS and e-mail servers.

An approach similar to the previous examples for the e-mail traffic can be applied to the lower-level traffic monitoring, such as on the transport- or link-layer levels by using the DNS activity data and another source such as the alerts from an external IDS (Intrusion Detection System). If a strong correlation of behavioral relationship between DNS and the other application activities were found, an incident could be detected solely from an anomaly in DNS activities.

In this paper, the authors first explain how a DNS server can be integrated in an IDS in Section 2, and the packet-data retrieval system and strategy for investigating the DNS-related traffic trends in Section 3. We what kind of relationship can be expected between the traffic of DNS and other TCP applications. We analyze an example taken from the real-world traffic in Section 4 to show some of the characteristics of the correlation between the DNS and other applications, and in Section 5 we conclude this paper.

2 DNS Server as an IDS Component

Proctor [7] defines that a network-based IDS depends on a packet-monitoring program which detects a pre-defined pattern of misuse on a network, commonly called *signatures*, while a host-based IDS processes the event logs for each host system. Snort [8] is a popular network-based IDS available as an open-source software, which has also been widely used as a research platform for a high-speed IDS.

One of the problems of the network-based system is that the signature is only applicable to each individual packet or a TCP stream and does not provide a tool for analyzing relationship of multiple events or streams. Another problem is that attempting to monitor *all* traffic packets will eventually lead to dropping packets because of the system performance limit, while a hardware assistance may increase the monitoring performance [9].

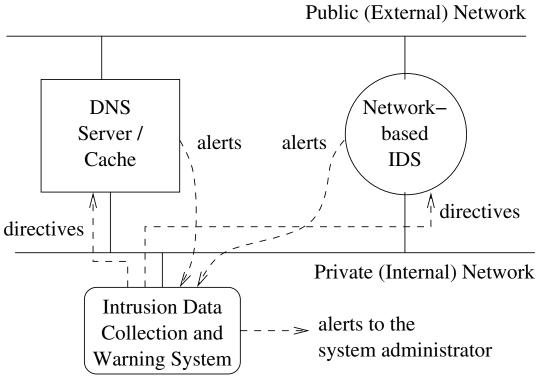
Public (External) Network
DNS Server / Cache
alerts
alerts
Network–based IDS
directives
directives
Private (Internal) Network
Intrusion Data Collection and Warning System
alerts to the system administrator

Fig. 1 A Model of IDS using a DNS Server as a Component

Figure 1 shows a diagram of our proposed model for an IDS. The network-based IDS and DNS server system, which may also include the cache functionality, are connected to both public and private networks, while they monitor the public network activities and interacts with the data collection and warning system inside the private network. The IDS and DNS systems send an alert or equivalent piece of logging information to the data collection system, and receive the directives from the warning system.

The reason why DNS server/cache is effective as an IDS component is as follows:

- Access from the public network to the DNS server/cache through TCP and UDP Port number 53 *must* be open. This means the DNS traffic is guided to the DNS server and must be processed there.
- DNS server software such as BIND [10] and djbdns [11] can be used to verify the contents and anomalies of the DNS traffic. For example, `tinydns`, the UDP non-recursive server of djbdns, logs per-packet details for every incoming query.
- The alert generated by the network-based IDS can also be used to control the behavior of the DNS server. If a DoS traffic to the DNS is detected by the IDS, the IDS can tell the DNS server to ignore or restrict processing the data on the suspected traffic.

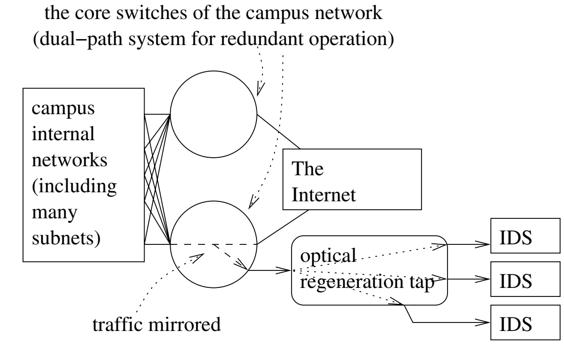
the core switches of the campus network
(dual−path system for redundant operation)
campus internal networks (including many subnets)
The Internet
optical regeneration tap
IDS
IDS
IDS
traffic mirrored

Fig. 2 Distribution System of Mirrored Packets for Multiple IDSes

- In many network system configurations, DNS access from the clients to the servers are relayed by a small number of the cache servers, to reduce the number of external traffics by reusing the cached DNS data. The cache servers forward both the queries and the replies, so they can be used as DNS traffic monitor devices.

Behavioral relationship between DNS traffic and other protocols, however, should be investigated to find out what kind of monitoring is necessary for a DNS server to act as an IDS component. While real-time response is preferred for an effective defense against attacks, retrieving and analyzing the past data is also effective to gather the long-term trends.

3 Retrieving DNS and Related Traffic Data

We collected a set of raw packet traffic data at a Internet-connected campus network. The network has a centralized architecture to separate the internal and the external networks by a set of core switches, as shown in Fig. 2.

We connected a FreeBSD host using Celeron [*1] CPU with the 1.4GHz clock speed and 512Mbytes of RAM running snort [8] with a 1000BASE-SX optical link to the regeneration tap shown in Fig. 2. We confirmed that the machine performance is sufficient to log the incoming mirrored traffic, of ≈12000 packets/sec.

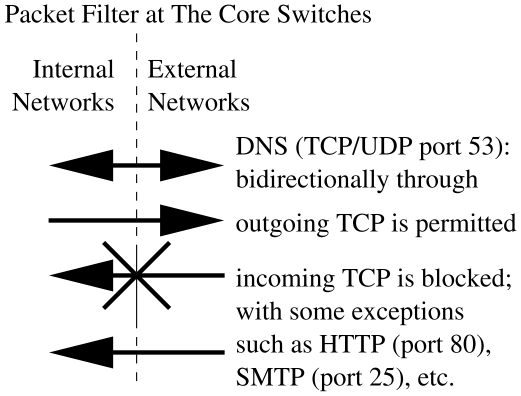
Packet Filter at The Core Switches
Internal Networks
External Networks
DNS (TCP/UDP port 53): bidirectionally through
outgoing TCP is permitted
incoming TCP is blocked; with some exceptions such as HTTP (port 80), SMTP (port 25), etc.

Fig. 3 Packet-filtering Policy of The Campus Network Core Switches

Since the campus network has a packet-filtering policy for the core switches shown in Fig. 3, and to minimize collecting the traffic contains private data, we decided to retrieve the following packets:

- all packets regarding the public DNS, of TCP and UDP including the port number 53 on either the source or the destination port;
- all TCP packets of initiating the connection, with the SYN bit in the TCP header is set;
- all TCP packets of acknowledging the connection, with the SYN and ACK bits in the TCP header are set.

The filtering function for snort and a packet analysis tool tcpdump [12] of the above rule is:

```
(udp port 53) || (tcp port 53) ||
   (tcp[13] & 0x2 != 0)
```

We ran snort solely to retrieve the raw packets, and later analyzed the packet dump file by tcpdump. Tcpdump can parse and display the DNS query types and other attributes in the DNS packets.

We collected two sets of continuous traffic flows, each continued for about an hour. About 6∼12% of the mirrored packets matched the filter and collected.

[*1] *Celeron* is a registered trademark of Intel Corporation.

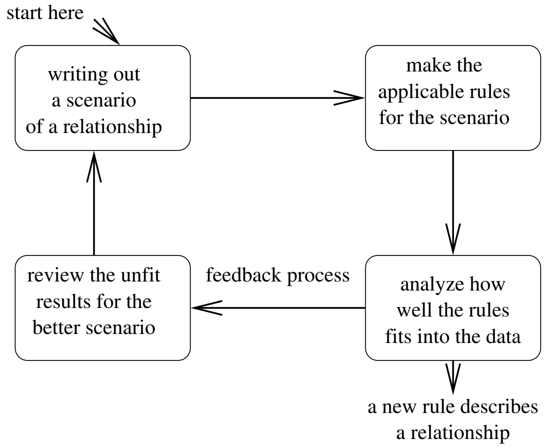
start here
writing out a scenario of a relationship
make the applicable rules for the scenario
analyze how well the rules fits into the data
review the unfit results for the better scenario
feedback process
a new rule describes a relationship

Fig. 4 Development Cycle of IDS Rules

Total number of DNS query answers	1437
DNS query answers including at least one A RR	650
Total number of valid A RRs in the answers above	1177

Table 1 Characteristics of retrieved DNS packets

Total number of TCP SYN packets	24652
TCP SYN packets included the destination address of A RRs in the DNS query answer within one second before the SYN packet sent	374

Table 2 Characteristics of retrieved SYN packets

4 An Example of Behavioral Analysis between DNS and TCP SYN Packets

In this section, we show an example of finding out relationship between DNS and TCP SYN packets.

Behavioral analysis between two or more independent phenomena is not a trivial task, and is rather heuristic. For example, the signature rules in snort and other IDS systems are written from the observation of traffic packets adapted to known vulnerabilities. This is rather a try-and-error process than a deterministic one as shown in Fig. 4.

Through the observation of the retrieved traffic, we reached to the following scenario, a hypothetical relationship:

> A TCP connection from the internal network to the external one is likely to occur immediately after resolving a domain name to an IP address, by receiving a DNS answer including an A RR (Resource Record) of the IP address [*2] from the external network to the internal one.

We picked up a 1-minute-length set of retrieved data from 11:49:45 JST, August 19, 2003 to verify how well the scenario fits into the real-world traffic. We developed the following rules:

- Extracting DNS query answers including A RRs;
- Searching the TCP SYN packets whose destination address is specified in one of the A RRs, during the time span of one second from the delivery of the query answers;
- Eliminate the possible duplication by only choosing the TCP SYN packet which occurs the earliest after the arrival of the DNS query answer;
- If the matching TCP SYN packet is found, we conclude that the pair of the DNS and TCP SYN packet fits into our scenario.

Table 1 and Table 2 show the results of the analysis. While the number of matching TCP SYN packets is only 1.5% of the total, it is 57.5% of the DNS replies contained at least one A RR, so it is highly likely that a DNS reply with A RR leads to successive TCP connection to the corresponding IP address.

Table 3 and Table 4 show the list of the destination port numbers in the TCP SYN packets, which suggests the service corresponding to the packets. Table 3 shows that the Microsoft NetBIOS Location Service is at the top of the destination port ranking, which corresponds to the W32/Blaster worm [13] spreading during the sampling period. This suggests that the Blaster worm did not resolve the host names by the DNS while in their infection

[*2] In this paper we only consider the IP version 4 addresses and packets, but the same rule is also applicable to an IP version 6 address and the AAAA or A6 RRs.

packets	port number	service name
11431	135	MS Loc-Srv
11086	80	HTTP
327	443	HTTPS
221	25	SMTP
143	110	POP3
49	10000	(unknown)
41	6667	IRC
41	53	DNS
40	7743	(unknown)
25	113	IDENT

Table 3 Top-ten Destination Ports for TCP SYN Packets

packets	port number	service name
347	80	HTTP
11	25	SMTP
8	443	HTTPS
3	113	IDENT
3	110	POP3
1	1755	(unknown)
1	10000	(unknown)

Table 4 Destination ports for TCP SYN Packets after Receiving DNS A RRs

activities. On the other hand, about 93% of the TCP SYN packets after receiving DNS A RRs are for HTTP [14] connections, which suggests that HTTP connections are followed after resolving the domain name part of the URL by DNS.

We also measured the amount of time passed between the DNS replies and corresponding TCP SYN packets. Fig. 5 shows that the distribution of the amount of time for each event is on an exponential curve [*3].

5 Conclusion and Further Works

In this paper, we proposed an IDS model using a DNS server as a component and how the DNS traffic gathered by the server can be used for an IDS. We also showed an example of behavioral analysis under a limited data of DNS and outgoing TCP SYN traffic, and found out that DNS query answers of the A RRs were highly likely to be followed by the TCP connection attempts to a given address among the A RRs. The relationship means that a mass flow of DNS query answers including A RRs to a DNS cache suggests some sort of DoS attack is ongoing from one of the clients connected to the DNS cache.

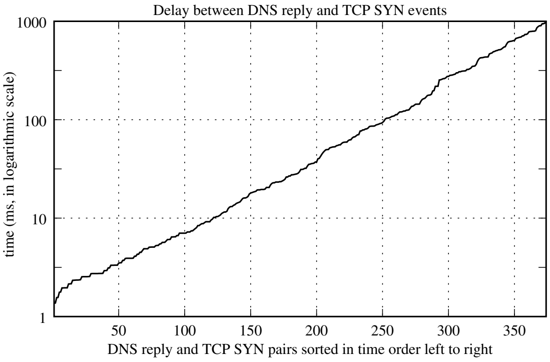

Fig. 5 Delay Time between DNS Replies and Corresponding TCP SYN Packets

We need to solve the following issues for making our proposed model more feasible for the production-level systems:

- Determining temporal relationship with two or more independent phenomena in a real-time manners requires to store the characteristics of each received packet to a FIFO (First-In-First-Out) buffer. Even for a post-mortem or data-forensic analysis, the events have to be indexed by the recorded time for a faster search.
- We need to develop a faster way to find out the statistical irregularities among the collected data. While heuristic methods by human beings is effective since they are based on the real-world experience, an automated methods such as an applied data mining should also be developed.
- In-depth analysis of DNS packets may be of privacy concerns for the network users, since it reveals the relationship between a client host and the targets which the client connects.

The implementation and detailed evaluation are currently ongoing as of September 2003.

Acknowledgments

Our thanks go to Mr. Tohru Asami, the president and CEO of KDDI R&D Laboratories, Inc., for supporting our research activities.

[*3] The curve is displayed linear in the Fig. 5 because the vertical axis is logarithmically scaled.

References

[1] Mockapetris, P. V.: Domain names – concepts and facilities (1987). RFC1034 (also STD13).

[2] Mockapetris, P. V.: Domain names – implementation and specification (1987). RFC1035 (also STD13).

[3] Postel, J.: Transmission Control Protocol (1981). RFC793 (also STD7).

[4] Postel, J.: User Datagram Protocol (1980). RFC768 (also STD6).

[5] Musashi, Y., Sugitani, K. and Matsuba, R.: Traffic Analysis on Mass Mailing Worm and DNS/SMTP, *IPSJ SIG Notes 2002-CSEC-19*, Vol. 2002, No. 122, pp. 19–24 (2002).

[6] Musashi, Y., Sugitani, K. and Matsuba, R.: Statistical Analysis in Logs of DNS Traffic and E-mail Server, *IPSJ SIG Notes 2002-CSEC-20*, Vol. 2003, No. 18, pp. 185–190 (2003).

[7] Proctor, P. E.: *The Practical Intrusion Detection Handbook*, Prectice-Hall (2001).

[8] M. Roesch et al.: Snort. `http://www.snort.org/`.

[9] Hayashi, T., Yokoyama, M., Takahara, A. and Iwahashi, M.: Evaluation of Intrusion Prevention System (IPS) with Snort Using a High-speed Hardware/Software Detection Architecture, *IPSJ SIG Technical Reports 2003-CSEC-21*, Vol. 2003, No. 45, pp. 59–64 (2003).

[10] Internet Software Consortium: BIND. `http://www.isc.org/bind/`.

[11] Bernstein, D. J.: `djbdns`. `http://cr.yp.to/djbdns.html`.

[12] TCPDUMP Public Repository: tcpdump. `http://www.tcpdump.org/`.

[13] CERT/CC: W32/Blaster Worm. CERT Advisory CA-2003-20, `http://www.cert.org/advisories/CA-2003-20.html`.

[14] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T.: Hypertext Transfer Protocol – HTTP/1.1 (1999). RFC2616.