Global Information Assurance Certification Paper

DNS Security Considerations and the Alternatives to BIND

Seng Chor, Lim
Version 1.0
October 2nd, 2001

Introduction

Abstract

This paper is going to discuss about the important considerations of the DNS Security. Due to the continuous break-ins to BIND 8 (one of the most popular choice of DNS server) in the past, this paper proposes either (a) securing your BIND 8 by running as an unprivileged user with chrooting into jail, (b) upgrading to BIND 9 and securing it running as an unprivileged user with chrooting into jail or (c) switch to using other alternatives. By the end of this paper, the reader will have some ideas on a more secure implementation of the DNS server.

Acknowledgement

The author would like to thank Michael Ian Hartley, Shannon Chong and Lee San Phang for their help of proofreading this paper.

Why DNS Security is so important?

The entire Internet depends on Domain Name System (DNS). Without DNS, the Internet users cannot access the Internet without resolving all the hostname into IP address while at the same time other external networks also cannot access your public servers. If the web server becomes inaccessible, people from external still can send email to the site. But, if the DNS servers are compromised, then the web servers and the SMTP servers (and any other Internet services) will all become inaccessible to the Internet. Some firewalls or proxy systems use hostname instead of IP address to build the access control lists. But if the DNS servers are compromised, the line of the defense will collapse if it depends on the DNS servers.

Nobody would want to see that happen. This paper has just shown the importance of the DNS security. Welcome to the DNS security world.

Some DNS Security Considerations

Do not place all the DNS Servers in the same subnet with the single choke point or router

"To combat denial of service attacks and prevent accidental service outages, eliminate single points of failure in your DNS infrastructure" ~Cricket Liu [7]

I am not too sure about you, but I had a bad experience like this: I had one big bunch of all my keys, including my bedroom key, main door key, backdoor key, gate key, storeroom key and other keys, in one key chain. One day I was so careless that I locked all the keys in my bedroom. I had no key to unlock my bedroom and I also had no other keys to open any doors. Can you believe that? I had just locked myself in my own house! After this incident, I have learned not to put all my keys in one single key chain.

The improper design of the network infrastructure is paradise to the Denial of Service (DoS) or Distributed Denial of Service (DDoS) attack [12a, 12b, 13]. The administrators or the network managers are strongly recommended not to place all the DNS servers in the same subnet with the only single choke point. Imagine if someone tries to flood the router with large volume of bogus network packets in order to bring the router down, the DNS servers will become inaccessible to the Internet. A good practice is to always distribute the DNS servers in different networks in different routing paths.

With the fact that the Microsoft Network was experiencing DoS and DDoS attack on 25-26 January 2001 [8, 9], Microsoft has decided to contract with Akamai to have backup DNS servers. [11] This idea would generally help preventing the downtime of the DNS service if one of the subnet where the DNS servers located is being attacked.

Running the DNS servers on different platforms

OS platform of your secondary DNS server should be chosen carefully. You want to minimize the threats for the known OS vulnerabilities issues to your DNS servers [14]. You don't want the same trick which killed your first DNS server to work on second or the rest of your DNS servers. If you are running identical Operating Systems for all your DNS, you will get yourself into trouble when someone uses a newly discovered OS bug(s) to attack. Sadly, all your DNS servers will be compromised. So, try to consider using different platforms for your DNS servers, e.g. Solaris and OpenBSD or AIX and FreeBSD. Of course, the complexity of managing the DNS servers would then be higher.

Split Horizon DNS

This is a very useful technique to hide your internal DNS structure from the hackers. Sometimes your DNS server is giving too much information to people and creating a roadmap for the hackers to attack your system or network. So, try to split your DNS servers into public (an external DNS which contains public servers information only) and private (an internal DNS which contains internal servers or workstation information). Then, place your public DNS servers in your perimeter network and place your private DNS servers in your internal protected network. You may consider to customize the behaviour of the name servers in your perimeter network and internal protected network.

Run your DNS server on a dedicated machine

You will never know whether other Internet services running on your DNS machine are vulnerable or not. But you know running only one Internet service (BIND) on your dedicated machine greatly creates a higher level of difficulty to the hacker who is watching you. Cricket Liu suggested that to filter out all traffic except traffic from the Internet to TCP and UDP port 53 if the host that running your name server is a dedicated machine [7].

If you want to remotely manage your DNS server, SSH is your best friend (BIND 9 has a very good remote administration tool called *rndc* though). Do not manage your DNS server in clear text through the insecure network [15].

Restricting Zone Transfers

Again, Cricket Liu [7] and Paul Albitz [16] suggested restricting zone transfers to prevent (a) *others from taxing your name server's resources*, (b) *hackers from listing the contents of your zones*. Zone transfer request from an unauthorized source is same as a stranger checking your financial records with unknown purpose. So, only allow the zone transfer requests from your authorized name servers.

Harden your DNS server and always review your configuration file

Your job is not just to ignore the DNS server after the installation. If you think your firewall can protect your DNS servers, you are probably wrong. You will be surprised when you see the attacks coming to the open TCP or UDP port (port 53).

So, you are not wasting your time to take a few simple steps to harden the DNS server, as they are very important. Later this paper is going to discuss how to harden your BIND 9, which is one of the favourite Internet DNS servers.

A good DNS server configuration file is always your first and minimal defense. Always review your configuration file(s). Employ some security software like Tripwire or other

similar software to verify the integrity of your DNS server binaries, configuration file(s), zone data and other important files like log file. Schedule the integrity check to run every day.

Stay current with the latest release or patch update

Always stay current with the latest release or patch update. You can join the mailing list that announces the latest release or patch update to keep you informed. These mailing list addresses can always be found at your vendor's website [7].

Do not run your DNS as root

Never run your DNS server as root user. In general, do not allow your Internet services running as root. If your DNS server is compromised, at least the hacker or intruder won't take over your whole machine by having root access [7]. A good approach would be running your DNS server in a chroot jail as an unprivileged user. A further discussion on how to implement chrooting your DNS server (BIND) into jail environment and running it as a non-root user will be discussed later.

Split-Service Name Server

Basically, name servers can be classified into two groups, Advertising Name Servers (Authoritative for Zone, the SOA), and Resolving Name Servers (resolver, caching, or to answer recursion DNS queries) [7]. In the presentation "*Securing an Internet Name Server*" by Cricket Liu (http://www.verisign-grs.com/dns/securing_an_internet_name_server.pdf), he has demonstrated the split-service name server configuration by way of example. Daniel J. Bernstein implements the idea and put into his djbdns DNS server by having a few individual programs to obtain a better security [1, 6]. Splitting the name server makes it easy to customize the behaviour and restrictions of each name servers. You may combine this technique with Split Horizon DNS.

Choose other alternatives

This may not be a DNS security consideration; I rather call it a trick. We learned that vulnerabilities are most often discovered on the most popular server. Some DNS servers are designed to achieve a better security. They attempt to replace the popular DNS server that has a lot of known vulnerabilities in its past history. A mixture of your DNS server choice among your secondary DNS servers would create difficulties for hackers trying to compromise all of your DNS servers compromised at the same time. Later, this paper will discuss the alternative available.

BIND

BIND (Berkeley Internet Name Daemon) is the most popular and famous implementation of the DNS protocols. Currently, there are nearly 80% of the sites over the Internet running BIND as their DNS servers. BIND is currently maintained at ISC (http://www.isc.org/bind/) [10]. Also, BIND development is supported by some large commercial organizations [17b].

The major components of the BIND includes [17a]:

- a Domain Name System server (named)
- a Domain Name System resolver Library
- tools for verifying the proper operation of the DNS server

BIND has its long bloody history over the years. As being the most popular DNS server on the Internet, it is always the favourite target of the attackers. Just like Sendmail, discussion of the BIND version is always a hot topic among the system administrators or IT professionals.

Several vulnerabilities have been identified by CERT/CC [18]. Please visit http://www.cert.org/advisories/CA-2001-02.html for more detailed information. For more information about the BIND security issues, please visit http://www.isc.org/products/BIND/bind-security.html.

But now, BIND 9 is totally rewritten and shares no code with BIND 8. Several of the security features have been described by ISC software [17b]. ISC software recommends BIND 4 and BIND 8 users upgrade their BIND DNS server to BIND 9. According to the announcement to bind-announce@isc.org made by Paul Vixie, chairman of ISC software, a fee-based membership forum will be established and consists only of qualified bind users [21]. The idea of creating the forum is to bring the qualified bind users together to discuss security issues out of the eyes of the general public.

Securing the BIND

Running a network service as the root user can be dangerous and BIND usually runs as root [16]. This is considered as a very bad security implementation because the hacker will gain root access to the system if he/she finds vulnerability in the BIND named server. However, BIND 8.1.2 (and above) and BIND 9.x.x allows itself to run as a non-root user and chroot into its jail environment. At the time of this writing, the current release of the BIND 9 is version 9.1.3 and the release of the BIND 8 is version 8.2.4.

The introduction of chroot model has raised the level of Internet Service security [19]. The idea of securing the BIND (named) is making it running in its own chrooted jail with "-t" command flag [3c]. To chroot the BIND (named), it is need to [3a, 3b, 3c, 3d, 3e, 3f, 3g]:

(a) create a jail directory e.g. /home/jail/bind
(b) create a dedicated user and group for BIND, e.g. named, named
(c) mknod the required system devices
(d) copy the library files required by named (and named-xfer for BIND 8)
(e) set the most restricted file and directory permissions to your jail directory
(f) copy the zone data files into the jail directory
(g) setup good named.conf files in the jail directory
(h) create the logging channel and setup your syslogd
(i) invoke the named with the flags "-g named –t /home/jail/bind"
(j) Lastly, you may modify your system startup script to start your chroot named during boot time

Note that some people may find it difficult to setup the syslog for BIND (named). If your syslogd supports the –a switch like OpenBSD do, you should have no problem to tell your syslogd listening to /home/jail/bind9/dev/log file by adding a command switch *"-a /home/jail/bind/dev/log"*. But if your syslogd has no "–a" option, maybe you can try to use *holelogd* as a syslogd replacement which is part of the *utils1.0* package at http://www.obtuse.com/ [3d].

If you are using BSD system with ports and cvsup installed, you can schedule a cron job to update your BIND 8 or BIND 9 on a regular basis. The BIND ports can be found at /usr/ports/net/bind8 and /usr/ports/net/bind9 [20a, 20b].

All the sites below contain useful guidelines which may guide you to secure the BIND 8 or BIND 9. This should generally cover most of the BSD/Linux and its variant [3a, 3b].

Chroot-BIND HOWTO, by Scott Wunsch: http://ldp.iol.it/HOWTO/Chroot-BIND-HOWTO.html

Dual chrooted BIND/DNS servers, by Dave Lugo:
http://www.etherboy.com/dns/chrootdns.html

Securing DNS (Linux Version), by Psionic Software:
http://www.psionic.com/papers/dns/dns-linux

Securing DNS (OpenBSD/FreeBSD Version), by Psionic Software:
http://ww.psionic.com/paper/dns/dns-openbsd

Hardening the BIND DNS Server, by Sean Boan for SecurityPortal:
http://securityportal.com/cover/coverstory20001002.html

Chrooting a DNS server on Solaris, by Adam Shostack:
http://www.homneport.org/~adam/dns.html

Secure BIND Template Version 2.1, by Rob Thomas:
http://www.cymru.com/~robt/Docs/Articles/secure-bind-template.html

The Alternatives to BIND

Why do we want to consider the alternatives to BIND?

Good question. Imagine you have a dishonest worker with a bad criminal record. He always swears to you that he "wouldn't do it again" every time you discover his dishonest behaviour. Now it is the time for you to consider whether you should hire another worker to replace him. But who can tell whether your next worker is good? Well, it is all up to you to make the decision.

BIND is complex, just like Sendmail. BIND implements its DNS server in one huge executable, *named*. This makes it very hard to audit. It is advisable to run something simple which delegates tasks.

At the time BIND was first developed, the Internet was shared by a small group of people and organizations. The nature and the design of the BIND did not take serious account of security considerations. But now, the release of the BIND 9 comes after continuous break-ins of BIND 8. As we know, BIND 9 is a total re-write of BIND 8 by a new team at ISC software. This might prove that the nature of design of BIND 4 and BIND 8 does not fit into the modern wild growing Internet.

Some may feel the development of BIND 9 is leading to a secure DNS server implementation. Others perhaps, think that staying with BIND 4 or BIND 8 is just fine and safe. Nevertheless, this paper will give some ideas on other alternatives available and what they offer.

What are the alternatives available?

Dents, djbdns [5] and MaraDNS are always the hot topics being discussed as a secure alternative to BIND. MaraDNS has its interesting modular characteristic and is a customizable DNS server. Other load balancing DNS servers available are lbnamed and lbdns [26].

Dents

Like the ISC BIND, Dents is another server implementation of the DNS (Domain Name System). Dents shares no code with any other project. It is free software released under the terms of the GPL (General Public License) version 2 [4,21]. The Dents developers are Todd Lewis, Johannes Erdfelt, Greg Rumple and their project team members in MindSpring Enterprise [4].

The Dents homepage can be found at http://www.dents.org/, but surprisingly there is not much information available from that site. The Dents has a very slow development history over the years. It is still in beta testing. At the time of this writing, the current release of Dents is version 0.3.1 (on July 11, 1999) [22].

Dents is coded in ANSI C and is oriented towards POSIX-conformant and POSIX-like systems. It uses POSIX threads and several features do not work without pthreads. According to Todd Lewis, Dent's main features are (a) *a modular driver architecture, which permits various means to be used to look up names,* (b) *a CORBA-based control facility, which allows administrators to control a running server,* (c) *a replaceable tree system,* (d) *a clean design* and (e) *good karma* [4, 22]. For more information about the design of dents, please read the "The Design of the Dents DNS Server" written by Todd Lewis.

Dents is designed and developed for performance and better server management purposes, its clean and neat design plays an important role in DNS security. There is also a very interesting feature that Dents offer: a driver module to integrate Dents with the ISC DHCP server [2].

Dents can be download at http://www.dents.org/src/ or http://sourceforge.net/projects/dents/.

Dents Debian Linux Package can be downloaded at http://www.promera.nl/dents.

Latest release of the Dents RPM package can be found at http://www.dents.wl.com/release/.

djbdns

Djbdns is a secure replacement of BIND and developed by Daniel J. Bernstein (also the author of qmail). Security was, and is, one of the primary motivations for the development of djbdns. Unlike BIND, djbdns is not a single, monolithic program. Instead, it is made up of a collection of small, independent, mutually distrusting programs (which are dnscache (caching and recursion name server), tinydns (authoritative name server), rbldns (RBL name server) and walldns (reverse lookup name server)) which each runs as separate user in its own chrooted jail [6].

It is interesting to note that the author is actually offering monetary award to the first

person to publicly report a verifiable security hole in the latest version of djbdns [23].

To quote from the djbdns website, the security features of djbdns are described as follow:

• dnscache, tinydns and walldns runs as dedicated non-root users and each of them runs inside its own chroot jail.

• dnscache discards DNS queries from outside a specified list of IP addresses.

• dnscache and the dns library use a new query ID and a new UDP port for each query packet. They discard DNS responses from any IP address other than the one that the corresponding query was just sent to.

• dnscache uses a cryptographic generator to select unpredictable port numbers and IDs.

• dnscache is immune to cache poisoning.

• tinydns and walldns never cache information. They do not support recursion.

If you are interested in the migration from BIND to djbdns, a good HOWTO document "BIND-to-djbdns Migration Guide / HOWTO" by Adam McKenna can be obtained at http://www.flounder.net/djbdns/bind-to-djbdns.html.

You can find a lot of useful information about djbdns at http://cr.yp.to/djbdns.html.

MaraDNS

Buffer overflow attack is one of the most common attacking methods to DNS server. The idea of buffer overflow attack is to cause the DNS server unable to handle the 'extra input' from the attacker and run any arbitrary command (e.g. by invoking a bourne shell). MaraDNS is an open source DNS server that tempts to be very secure by handling the buffer overflow very well. The author, Sam Trenholme, explained the security features of MaraDNS as: (a) minimizing the buffer overflow by using a special homegrown string library – spam-protection package, and (b) running as an unprivileged user in a chrooted environment [24].

MaraDNS currently has no support for caching or recursion, but the author has planned to add this ability into the next release. Please check the latest information at http://www.maradns.org or the mirror site at http://www2.maradns.org/.

CustomDNS

CustomDNS is a modular DNS server written in Java which based on Brian Wellington's dnsjava package. It is also a customizable DNS server by programming it. Eric Kidd, the author, adds the support for virtual hostnames, SQL databases, and dynamic client updates [25, 26].

The CustomDNS homepage can be found at http://customdns.sourceforge.net/.

lbnamed

lbnamed is a load balancing name server written in Perl. The author is Roland Schemers. For more information about lbnamed, please try http://www.stanford.edu/~riepel/lbnamed/ [26].

lbdns

lbdns is another load balancing server just like lbnamed. At the time I write this paper, the website of lbdns seems not to be accessible [26].

Conclusion

As you can discover from the descriptions of each DNS servers above, the development and the design of the modern name server follows the DNS security considerations I have mentioned above, for examples, BIND 9, djbdns, dents and MaraDNS.

This paper does not discuss about the best and the most secure DNS server; however, this paper suggests that a mixture of using different well-tested DNS servers on different OS platforms could be a good idea to create a higher level of complexity to an attacker. But, it must be tested before the administrator or the network manager put it into their development line.

References:

[1] Life With djbdns, by Henning Brauer, *25 September 2001*
http://www.lifewithdjbdns.org/

[2] Dynamic content in the DNS system using Dents, by Todd Lewis
http://lwn.net/1999/features/dyndnsdents.html

[3a] “Securing DNS (OpenBSD/FreeBSD Version)”, by Psionic Software
http://www.psionic.com/papers/dns/dns-openbsd

[3b] “Securing DNS (Linux Version)”, by Psionic Software
http://www.psionic.com/papers/dns/dns-linux

[3c] “Chroot-BIND HOWTO”, by Scott Wunsch
http://ldp.iol.it/HOWTO/Chroot-BIND-HOWTO.html

[3d] “Dual chrooted BIND/DNS servers”, by Dave Lugo
http://www.etherboy.com/dns/chrootdns.html

[3e] “Hardening the BIND DNS Server”, by Sean Boan for SecurityPortal
http://securityportal.com/cover/coverstory20001002.html

[3f] “Chrooting a DNS server on Solaris”, by Adam Shostack
http://www.homneport.org/~adam/dns.html

[3g] Secure BIND Template Version 2.1, by Rob Thomas
http://www.cymru.com/~robt/Docs/Articles/secure-bind-template.html

[4] The Design of the Dents DNS Server, by Todd Lewis
http://www.usenix.org/events/usenix99/lewis.html

[5] BIND news and DNS alternatives, by Jeremy C. Reed
http://www.mail.archive.com/bsdtoday-text@list4.internet.com/msg00011.html

[6] “A look at djbdns”, by Jonathan Corbet, February 8, 2001
http://lwn.net/2001/features/djbdns.php3

[7] “Securing an Internet Name Server”, by Cricket Liu
http://www.verisign-grs.com/dns/securing_an_internet_name_server.pdf

[8] “Microsoft attack raises concern over new DDOS variant”, by Fisher, Dennis & Callaghan, Dennis
http://www.zdnet.com/eweek/stories/general/0,11011,2679094.00.html

[9] “More Microsoft Web woes; this time, it’s hackers”, by Dudley, Brier
http://seattletimes.nwsource.com/cgi-bin/WebObject/SeattleTimes.woa/wa/gotoArticle?zsection_id=268466359&text_only=0&slug=microsoft26&document_id=134262903

[10] “A Brief History of BIND”, by ISC software
http://www.isc.org/products/BIND/bind-history.html

[11] “Technology’s Monday Morning Quarterback”, by Rowell, Erica D
http://www.abcnews.go.com/sections/scitech/Daily/News/microsoft010129.html

[12a] “Network Attack: Denial of Service (DoS) and Distributed Denial of Service (DDoS), by Hancock, Bill, PhD
http://www.exodus.com/information/ddos/index.html

[12b] “Denial of Service attack”, by Fuller, Edward
http://www.sans.org/infosecFAQ/securitybasics/dos.htm

[13] “Microsoft Network: The Anatomy of a DoS”, by Field, Ben
http://www.securityportal.com/articles/microsoft20010125.html

[14] “Selection and Operation of Secondary DNS Servers”, IETF RFC 2182
http://www.dns.net/dnsrd/rfc/rfc2182.html#4g.Unreachableservers

[15] “Unix Secure Shell”, by Anne Carasik, McGraw Hill, ISBN 0-07-134933-2

[16] “DNS and BIND”, by Paul Albitz & Cricket Liu, O’Reilly, ISBN 1-56592-512-2

[17a] “ISC BIND”, by ISC software
http://www.isc.org/products/BIND/

[17b] “ISC BIND 9”, by ISC Software
http://www.isc.org/products/BIND/bind9.html

[18] “CERT Advisory CA-2001-02 Multiple Vulnerabilities in BIND”, by CERT® Coordination Center
http://www.cert.org/advisories/CA-2001-02.html

[19] “Using jail to imprison processes and their descendants for increased security”, by Davec, March 11, 2001
http://www.bsdpro.com/info.php?cat=security&fileid=00014#article

[20a] “FreeBSD Ports”, by the FreeBSD Project
http://www.freebsd.org/ports/index.html

[20b] “OpenBSD Ports”, by the OpenBSD Project
http://www.openbsd.org/ports.html

[21] “Re: DENTS: How would we describe dents”, by Todd Graham Lewis
http://lists.omnipotent.net/199906/msg00057.html

[22] “Project: Dents: Summary”, the SourceForge
http://sourceforge.net/projects/dents/

[23] “The djbdns security guarantee”, D. J. Bernstein
http://cr.yp.to/djbdns/guarantee.html

[24] “MaraDNS 0.5.13 released”, by Sam Trenholme, *April 22, 2001*
http://security-archive.merton.ox.ac.uk/security-audit-200104/0012.html

[25] “CustomDNS Project”, http://freshmeat.net/projects/customdns/

[26] “Other DNS Software”, by D. J. Bernstein
http://cr.yp.to/djbdns/other.html