
# DNS Security
in
# AUSTRALIA

**Version 1.0**

**Prepared by**

**Adrian Ashbury**
**and**
**Craig S Wright**

**June 2000**

# TABLE OF CONTENTS

# 1. Introduction

BIND (Berkeley Internet Name Domain) is an implementation of the Domain Name System (DNS) protocols and provides an openly re-distributable reference implementation of the major components of the Domain Name System, including:

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

The BIND DNS Server is used on the vast majority of name serving machines on the Internet, providing a robust and stable architecture on top of which an organization's naming architecture can be built. The resolver library included in the BIND distribution provides the standard APIs for translation between domain names and Internet addresses and is intended to be linked with applications requiring name service.

## 1.1 Scope

The scope of this document is limited to revealing the state of DNS security within Australia at the present moment.

## 2. A Brief History of BIND

The Berkeley Internet Name Domain package was originally written at University of California at Berkeley as a graduate student project under a grant from the US Defence Advanced Research Projects Administration (DARPA). Versions of BIND through 4.8.3 were maintained by the Computer Systems Research Group (CSRG) at UC Berkeley. Douglas Terry, Mark Painter, David Riggle and Songnian Zhou made up the initial BIND project team.

After that, additional work on the software package was done by Ralph Campbell. Kevin Dunlap, a Digital Equipment Corporation employee on loan to the CSRG, worked on BIND for 2 years--from 1985 to 1987. Many other people also contributed to its development during that time: Doug Kingston, Craig Partridge, Smoot Carl-Mitchell, Mike Muuss, Jim Bloom and Mike Schwartz. BIND maintenance was subsequently handled by Mike Karels and O. Kure.

BIND versions 4.9 and 4.9.1 were released by Digital Equipment Corporation (now Compaq Computer Corporation). Paul Vixie, then a DEC employee, became BIND's primary caretaker. Paul was assisted by Phil Almquist, Robert Elz, Alan Barrett, Paul Albitz, Bryan Beecher, Andrew Partan, Andy Cherenson, Tom Limoncelli, Berthold Paffrath, Fuat Baran, Anant Kumar, Art Harkin, Win Treese, Don Lewis, Christophe Wolfhugel, and others.

BIND Version 4.9.2 was sponsored by Vixie Enterprises. Paul Vixie became BIND's principal architect/programmer.

BIND versions from 4.9.3 onward have been developed and maintained by the Internet Software Consortium with support being provided by ISC's sponsors. As co-architect/programmers, Bob Halley and Paul Vixie released the first production-ready version of BIND version 8 in May 1997.

(Today, BIND version 4 is officially deprecated in favor of BIND version 8 and no additional development will be done on BIND version 4 other than for security related patches.)

The most current version of BIND available at time of writing was BIND Version 8.2.2 patchlevel 5 *(Released November 12th, 1999)* . The latest beta version is BIND 8.2.3 T5B (RC3), released May 11, 2000. This is pre-release code and is not intended for production use.

# 3. Domain Name Service

## 3.1 What is DNS?

The domain name system (DNS) is the way that Internet domain names are located and translated into IP (Internet Protocol) addresses. A domain name is a meaningful and easy-to-remember "handle" for an Internet address.

Because maintaining a central list of domain name/IP address correspondences would be impractical, the lists of domain names and IP addresses are distributed throughout the Internet in a hierarchy of authority. There is probably a DNS server within close geographic proximity to your access provider that maps the domain names in your Internet requests or forwards them to other servers in the Internet.

## 3.2 How do I make my DNS secure?

To answer this question we must first look at what needs to be secure.

### 3.2.1 Network Level

DNS queries operate over UDP 53. Zone transfers operate over TCP 53.

Ensure your security enforcing devices, such as packet filters and firewalls permit only client query access to the DNS server to UDP 53 for anyone external to your organization.

Ensure your security enforcing devices, such as packet filters and firewalls permit only zone transfers (TCP 53) between your DNS servers and it is not possible from any other server external to your organization.

### 3.2.2 Operating System

The underlying operating system of the DNS server must be secured in a manner that:

- It allows it to be accessible to authorized users only and prevent access from unauthorized users,
- The operating system loaded on the DNS server should have the bare minimum of functionality installed,
- No other services should run on the DNS server,
- File and directory permissions should be the leanest possible for normal operation.

### 3.2.3 DNS Software Version

The version of software you should run on your DNS is the latest supported version available at the time. Make sure you have all of the latest applicable security patches installed.

### 3.2.4 DNS Software Configuration

Securely configuring your DNS software is a must, without this important step, the integrity of your DNS will be compromised.

A general rule of thumb when configuring DNS is to "Enable on that which is required, from only the locations it is required, and disable the rest".

#### 3.2.4.1 Primary DNS

The DNS software needs to be configured in such a manner as to allow:

- Anyone, anywhere, to resolve the names of your externally visible hosts to IP Addresses and vice versa,
- Your primary DNS to forward queries for hosts it does not know to a root server, and
- The primary DNS for your domain to update the configuration of the secondary DNS servers for your domain.

#### 3.2.4.2 Secondary DNS

The software on the secondary DNS needs to be configured in such a manner as to allow:

- Anyone, anywhere, to resolve the names of your externally visible hosts to IP Addresses and vice versa,
- The DNS to forward queries for hosts it does not know to a root server, and
- The primary DNS for your domain to update the configuration of the secondary DNS servers for your domain.

## 3.3 What are some of the threats you could face from an insecure DNS?

The threats are many, we do not plan to cover all of them in this document and they are as limitless as one's imagination. We will briefly cover a few in the following sections.

The threats mentioned below have been broken down into the categories of those against Confidentiality, Availability and Integrity.

### 3.3.1 Threats to Confidentiality

#### 3.3.1.1 Eavesdrop Attack

##### 3.3.1.1.1 Mail

If you run any other software on the DNS server, and the DNS is compromised in such a manner that allows the attacker operating system level access to the server hosting the DNS, any data traversing the server that runs the DNS will be able to be intercepted and captured by the attacker.

For example, if your DNS server was also a mail relay for your organization, the attacker could read all mail messages entering or exiting your domain. If there was a lack of adherence within your organization to obeying your security policy, and sensitive information was being regularly transmitted via email, the attacker could collect a lot of valuable information from this attack.

##### 3.3.1.1.2 General Traffic Sniffing

If the DNS was poorly located in such a way that all traffic entering or exiting your organization had to pass it, the compromised server could be used to eavesdrop on all inbound and outbound traffic, such as:

- E-Commerce transactions,
- Remote access sessions, and
- File transfers.

#### 3.3.1.2 Trust Relationship Exploit

If the security of your organization had been poorly configured to allow the DNS to access other servers within your organization, or even in your bastion zones, it could be used as a springboard by a successful attacker from which to launch attacks against other more valuable information assets.

### 3.3.2 Threats to Integrity

The following are examples of what could happen in the event of a compromise of integrity of your DNS.

#### 3.3.2.1 Mail Redirection

If an attacker can alter the address of your primary mail exchanger (MX) record, they can effectively:

- Deny your ability to receive mail,

- Receive all of your mail and reply to it making it look like it came from your organization and bring your organization into disrepute by sending obscene or inaccurate replies,
- Publicising sensitive mail messages on newsgroups or other media thereby causing loss of trust from your customers/shareholders,
-

#### 3.3.2.2 Web Redirection

In this scenario, if an attacker can alter your DNS records, they could redirect your customers to:

- Your competitors site,
- A bogus site containing anti-social content,
- A site that looks like your site but contains inaccurate content,
- A site that states your site has gone out of business,
- Etc, etc.

#### 3.3.2.3 E-Commerce Redirection

In this scenario, if an attacker can alter your DNS records, they could redirect your customers to another site:

- which takes their orders, accepts the payment but doesn't provide the goods, or
- proxies all traffic back to your real e-commerce server to capture customer details and credit card information.

#### 3.3.2.4 Masquerading

In this scenario, an attacker places their address into your DNS so it appears as they are one of your systems, and then commits acts against other hosts on the Internet pretending to be from your domain.

This is as simple as removing one of your IP Addresses and inserting theirs in the DNS so when their address resolves in someone's log files, appear to look like a server from your domain. They commit the attacks on others and then change it back to normal and when the person suffering the attack goes hunting for the attacker, it looks like you did it.

This type of attack could cause very bad publicity for your organization and subsequent loss of customers/shareholders.

### 3.3.3 Threats to Availability

Your DNS is probably the most critical part of your organization, without it:

- People cannot determine where to send mail to you, and

- People cannot determine how to get to any of the services you provide.

#### 3.3.3.1 Redirection

In this attack, the attacker simply redirects the address of any of your servers to a non-functioning address thereby making your site inaccessible.

Another twist on this attack could be used to direct all of your web and mail traffic to a server within your domain, on another DMZ, which was not running a mail relay or web server. This would have the added effect of causing additional load on your security enforcing devices such as packet filters and firewalls as the traffic bounced in towards the server not running the services and out again as it got rejected resulting in twice the traffic levels normally experienced by your organization.

#### 3.3.3.2 Deletion

In this attack the attacker removes entries from your DNS servers thereby making those hosts inaccessible.

# 4. Just how many DNS in Australia are Vulnerable?

We have been concerned for some time now about the number of DNS compromises in this country and recently decided to run a scan looking for vulnerable DNS servers in an effort to see just how bad the problem was.

During this scan, we performed certain tests and categorized the servers into three main categories:

- Those susceptible to root level compromise due to insecure versions of software,
- Those susceptible to integrity compromise due to poor configuration, and
- Those susceptible to Denial of Service.

## 4.1 Scan Methodology

To prevent the possibility of an all out war being declared by script kiddies on every DNS in the country before the rightful owners have had time to check if they are vulnerable and correct the problems, we will not reveal which tools were used or which vulnerabilities we tested for.

## 4.2 The Results

The results are very disconcerting. Even though we thought things were pretty bad, we never in our wildest imaginations thought this could be quite this bad.

### 4.2.1 How many domains did we check?

The following table indicates the number of servers, by TLD that were checked by our tests.

| **Domain Servers hosting Domains of type** | **Servers** |
|---|---|
| Com.au | 6523 |
| net.au | 2981 |
| org.au | 588 |
| gov.au | 452 |
| Other (inc NZ Domains) | 2667 |
| Com | 11820 |
| **Total Domain Servers Scanned (In Au and NZ Ip's)** | 8871 |

## 4.3 Test results

Detailed in the tables in the following sections are the results of our tests.

### 4.3.1 Results by TLD

| Domains by Subdomain | # Domains | # Vulnerable DoS | # Vulnerable Root Compromise |
|---|---|---|---|
| Com.au | 65528 | 52422 | 16250 |
| net.au | 19813 | 13869 | 3051 |
| org.au | 7199 | 6869 | 3915 |
| gov.au | 2501 | 1633 | 914 |
| other – AU | 3990 | 3491 | 1711 |
| Com | 78633 | 63906 | 16428 |
| Net | 25555 | 17588 | 2287 |
| Org | 6544 | 5476 | 3559 |
| Other (inc NZ Domains) | 75076 | 59060 | 36577 |
| **Total Domains** | **284839** | **224314** | **84692** |

### 4.3.2 Results by Server Types

| Server Types | Vulnerable Root Vulnerable DoS | Vulnerable Root Compromise | Currently Secured Servers | Total Servers |
|---|---|---|---|---|
| Bind Old | 5032 | 4232 | 0 | 5032 |
| BIND 8.2.2 - SP5 | 94 | 0 | 704 | 798 |
| NT (Supplied by MS) | 870 | 146 | 469 | 1339 |
| Mac | 155 | 0 | 179 | 334 |
| Other | 520 | 256 | 848 | 1368 |
| **Total** | 6671 | 4634 | 2200 | 8871 |

### 4.3.3 Percentages

Of the total number of servers tested, 75% were discovered to be vulnerable to Denial of Service attacks though misconfigurations and/or inappropriate version of bind being used whilst 52% were discovered to be vulnerable to root compromise from inappropriate version of bind being used. Some of the servers were vulnerable to both Denial of Service attacks and root level compromise.

# 5. Conclusions

As can be seen by the results in the previous section, most servers tested had vulnerabilities. Some vulnerabilities were caused through misconfigurations due to lack of DNS knowledge by support staff, some were caused from old versions of bind being used (once again through lack of knowledge) and some servers had both.

The primary reasons for the large number of vulnerabilities discovered were due to DNS hosting whereby an organization:

- with a vulnerable version of bind was hosting many domains thereby making ALL of them vulnerable, or
- with unskilled personnel configuring the DNS for the domains they were hosting thereby making ALL of the domains vulnerable.

All of the detected vulnerabilities were ones that could have been prevented through the use of appropriately trained staff.