the CENTER for INTERNET SECURITY

Security Configuration Benchmark For

DNS BIND 9.0 – 9.5

Version 2.0.0
May 2009

http://cisecurity.org
feedback@cisecurity.org

Background.

CIS provides benchmarks, scoring tools, software, data, information, suggestions, ideas, and other services and materials from the CIS website or elsewhere ("**Products**") as a public service to Internet users worldwide. Recommendations contained in the Products ("**Recommendations**") result from a consensus-building process that involves many security experts and are generally generic in nature. The Recommendations are intended to provide helpful information to organizations attempting to evaluate or improve the security of their networks, systems and devices. Proper use of the Recommendations requires careful analysis and adaptation to specific user requirements. The Recommendations are not in any way intended to be a "quick fix" for anyone's information security needs.

No representations, warranties and covenants.

CIS makes no representations, warranties or covenants whatsoever as to (i) the positive or negative effect of the Products or the Recommendations on the operation or the security of any particular network, computer system, network device, software, hardware, or any component of any of the foregoing or (ii) the accuracy, reliability, timeliness or completeness of any Product or Recommendation. CIS is providing the Products and the Recommendations "as is" and "as available" without representations, warranties or covenants of any kind.

User agreements.

By using the Products and/or the Recommendations, I and/or my organization ("**we**") agree and acknowledge that:

No network, system, device, hardware, software or component can be made fully secure;
We are using the Products and the Recommendations solely at our own risk;

We are not compensating CIS to assume any liabilities associated with our use of the Products or the Recommendations, even risks that result from CIS's negligence or failure to perform;

We have the sole responsibility to evaluate the risks and benefits of the Products and Recommendations to us and to adapt the Products and the Recommendations to our particular circumstances and requirements;

Neither CIS, nor any CIS Party (defined below) has any responsibility to make any corrections, updates, upgrades or bug fixes or to notify us if it chooses at it sole option to do so; and

Neither CIS nor any CIS Party has or will have any liability to us whatsoever (whether based in contract, tort, strict liability or otherwise) for any direct, indirect, incidental, consequential, or special damages (including without limitation loss of profits, loss of sales, loss of or damage to reputation, loss of customers, loss of software, data, information or emails, loss of privacy, loss of use of any computer or other equipment, business interruption, wasted management or other staff resources or claims of any kind against us from third parties) arising out of or in any way connected with our use of or our inability to use any of the Products or Recommendations (even if CIS has been advised of the possibility of such damages), including without limitation any liability associated with infringement of intellectual property, defects, bugs, errors, omissions, viruses, worms, backdoors, Trojan horses or other harmful items.

Grant of limited rights.

CIS hereby grants each user the following rights, but only so long as the user complies with all of the terms of these Agreed Terms of Use:

Except to the extent that we may have received additional authorization pursuant to a written agreement with CIS, each user may download, install and use each of the Products on a single computer;

Each user may print one or more copies of any Product or any component of a Product that is in a .txt, .pdf, .doc, .mcw, or .rtf format, provided that all such copies are printed in full and are kept intact, including without limitation the text of this Agreed Terms of Use in its entirety.

Retention of intellectual property rights; limitations on distribution.

The Products are protected by copyright and other intellectual property laws and by international treaties. We acknowledge and agree that we are not acquiring title to any intellectual property rights in the Products and that full title and all ownership rights to the Products will remain the exclusive property of CIS or CIS Parties. CIS reserves all rights not expressly granted to users in the preceding section entitled "Grant of limited rights." Subject to the paragraph entitled "Special Rules" (which includes a waiver, granted to some classes of CIS Members, of certain limitations in this paragraph), and except as we may have otherwise agreed in a written agreement with CIS, we agree that we will not (i) decompile, disassemble, reverse engineer, or otherwise attempt to derive the source code for any software Product that is not already in the form of source code; (ii) distribute, redistribute, encumber, sell, rent, lease, lend, sublicense, or otherwise transfer or exploit rights to any Product or any component of a Product; (iii) post any Product or any component of a Product on any website, bulletin board, ftp server, newsgroup, or other similar mechanism or device, without regard to whether such mechanism or device is internal or external, (iv) remove or alter trademark, logo, copyright or other proprietary notices, legends, symbols or labels in any Product or any component of a Product; (v) remove these Agreed Terms of Use from, or alter these Agreed Terms of Use as they appear in, any Product or any component of a Product; (vi) use any Product or any component of a Product with any derivative works based directly on a Product or any component of a Product; (vii) use any Product or any component of a Product with other products or applications that are directly and specifically dependent on such Product or any component for any part of their functionality, or (viii) represent or claim a particular level of compliance with a CIS Benchmark, scoring tool or other Product. We will not facilitate or otherwise aid other individuals or entities in any of the activities listed in this paragraph.

We hereby agree to indemnify, defend and hold CIS and all of its officers, directors, members, contributors, employees, authors, developers, agents, affiliates, licensors, information and service providers, software suppliers, hardware suppliers, and all other persons who aided CIS in the creation, development or maintenance of the Products or Recommendations ("**CIS Parties**") harmless from and against any and all liability, losses, costs and expenses (including attorneys' fees and court costs) incurred by CIS or any CIS Party in connection with any claim arising out of any violation by us of the preceding paragraph, including without limitation CIS's right, at our expense, to assume the exclusive defense and control of any matter subject to this indemnification, and in such case, we agree to cooperate with CIS in its defense of such claim. We further agree that all CIS Parties are third-party beneficiaries of our undertakings in these Agreed Terms of Use.

Special rules.

CIS has created and will from time to time create special rules for its members and for other persons and organizations with which CIS has a written contractual relationship. Those special rules will override and supersede these Agreed Terms of Use with respect to the users who are covered by the special rules. CIS hereby grants each CIS Security Consulting or Software Vendor Member and each CIS Organizational User Member, but only so long as such Member remains in good standing with CIS and complies with all of the terms of these Agreed Terms of Use, the right to distribute the Products and Recommendations within such Member's own organization, whether by manual or electronic means. Each such Member acknowledges and agrees that the foregoing grant is subject to the terms of such Member's membership arrangement with CIS and may, therefore, be modified or terminated by CIS at any time.

Choice of law; jurisdiction; venue.

We acknowledge and agree that these Agreed Terms of Use will be governed by and construed in accordance with the laws of the State of Maryland, that any action at law or in equity arising out of or relating to these Agreed Terms of Use shall be filed only in the courts located in the State of Maryland, that we hereby consent and submit to the personal jurisdiction of such courts for the purposes of litigating any such action. If any of these Agreed Terms of Use shall be determined to be unlawful, void, or for any reason unenforceable, then such terms shall be deemed severable and shall not affect the validity and enforceability of any remaining provisions. We acknowledge and agree that we have read these Agreed Terms of Use in their entirety, understand them and agree to be bound by them in all respects.

Table of Contents

Overview

This document, *Security Configuration Benchmark for ISC BIND 9.5*, provides prescriptive guidance for establishing a secure configuration posture for BIND versions 9.0.0 – 9.5.0 running on Linux or Solaris. This guide was tested against BIND 9.5.0-P2 on Gentoo Linux 2008.0, BIND 9.3.1 on Red Hat Fedora Core 4, and BIND 9.2.4 on Solaris 10 03/2005. To obtain the latest version of this guide, please visit http://cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

Consensus Guidance

This guide was created using a consensus review process comprised of volunteer and contract subject matter experts. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, and platform deployment personnel who plan to develop, deploy, assess, or secure solutions that incorporate the ISC BIND Domain Name Service (DNS) server on a Linux of Solaris platform.

Acknowledgements

The Center for Internet Security would recognize the individuals that significantly contributed to creation of this guide.

Authors
Dan Berry, *Leviathan Security Group*, v2.0.0
Ralph Durkee, *Durkee Consulting,* v1.0.0

Contributors and Reviews
Glenn Brunette, *Sun Microsystems*
Brian Campbell, *Leviathan Security Group*
Blake Frantz, *Center for Internet Security*
Dave Shackleford
Chad Thunberg, *Leviathan Security Group*
John Traenkenschuh
Rex Warren, *Leviathan Security Group*

Typographic Conventions

The following typographical conventions are used throughout this guide:

Convention	Meaning
`Stylized Monospace font`	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
`Monospace font`	Used for inline code, commands, or examples. Text should

	be interpreted exactly as presented.
<italic font in brackets>	Italic texts set in angle brackets denote a variable requiring substitution for a real value.
Italic font	Used to denote the title of a book, article, or other publication.
Note	Additional information or caveats

Configuration Levels

This section defines the configuration levels that are associated with each benchmark recommendation. Configuration levels represent increasing levels of security assurance.

Level-I Benchmark settings/actions

Level-I Benchmark recommendations are intended to:

- be practical and prudent;
- provide a clear security benefit; and
- do not negatively inhibit the utility of the technology beyond acceptable means

Level-II Benchmark settings/actions

Level-II Benchmark recommendations exhibit one or more of the following characteristics:

- may negatively inhibit the utility or performance of the technology
- acts as defense-in-depth measure

Scoring Status

This section defines the scoring statuses used within this document. The scoring status indicates whether compliance with the given recommendation is discernable in an automated manner.

Scorable

The platform's compliance with the given recommendation can be determined via automated means.

Not Scorable

The platform's compliance with the given recommendation cannot be determined via automated means.

Recommendations

1. Architecture and Foundation

This section provides guidance on the overall architecture and foundation of DNS.

1.1 Name Server Roles and Architecture

DNS name servers are a foundational part of your network architecture. How many name servers you need and what roles they should play depends on your organization's network

architecture. For this reason it is critical that the DNS strategy be considered early on, while decisions about the network topology are being formed. Questions that should be answered include, "how is the e-mail going to be delivered?", "are there going to be DNS sub-domains for the organization?", "is DHCP going to be used?", and "is Microsoft Windows Active Directory going to be used?" Providing the detailed information needed to make a recommendation for every possible DNS architecture is beyond the scope of this CIS Benchmark. However, some important DNS architectural recommendations and principles are discussed in this section.

1.1.1 Define the Name Server's Role (Level 1, Not Scorable)

Description:
The following is a list of the roles for Domain Name Servers:

1.1.1.1 Master Authoritative Only

A master name server is a master or authoritative name server for one or more domains. The master name server is the source of authority where administrators will make their DNS record changes.

1.1.1.2 Slave Authoritative Only

A slave name server is a name server that is authoritative for the domains, but receives all information and updates via zone transfers from a master name server or sometimes from another slave name server.

1.1.1.3 Caching Only

A caching only name server is not authoritative for any domain, but provides DNS service for other clients and systems, and will perform recursive DNS queries on behalf of its clients, and will cache answers to improve performance.

1.1.1.4 Forwarder

A forwarder name server is one that forwards queries to another name server to do the work of looking up the answer. The goal is to aggregate the work in order to make better usage of large caches, or sometimes to save on network bandwidth. Usage of forwarders has multiple security implications as discussed in section 2.5 on page 12.

Rationale:
There are wide varieties of mixed roles that are possible, but not necessarily recommended. Any authoritative name server will also cache answers, and caching name servers may be authoritative for domains. It is also possible to have multiple master name servers and to mix master and slave by having a name server act as a master for some domains while acting as a slave for others. Simplicity in your architecture should be a leading goal. Mixing name server roles is not recommended for most situations. There are specific threats and mitigating controls for each role, and by mixing these roles, you may be aggregating the risks and preventing a wise separation of services. For example, external slave name servers have the highest threat level, and therefore should have the least

information, the least functionality, and most stringent security configuration. While the internal caching name servers should not be exposed to external traffic as, a compromise or a poisoned cache on these servers would constitute an external attack on your internal clients. Likewise, your internal authoritative name servers typically contain sensitive information that should not be exposed to external queries, or even the external responses that the internal caching name server will be receiving. Separating these roles significantly mitigates these threats.

Remediation:
Design and document your DNS architecture, including the specific roles for each DNS server, the major security controls in place, and what networks will be able to query each server. Also, consider what sub-domains will be used and managed, how e-mail will be delivered, and how any updates will be performed.

Audit:
Not Applicable

Default Value:
Not Applicable

References:

1. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).
2. "DNS for Rocket Scientists - Contents," http://www.zytrax.com/books/dns/.

1.1.2 Utilize a Split-Horizon Architecture (Level 1/Not Scorable)

Description:
Running a Split-Horizon DNS architecture refers to running authoritative DNS servers and services for external DNS queries separate from the internal authoritative DNS servers, which answer all queries originating from within the organization. The external servers are configured to provide only a limited amount of information for the services needed for communication with external clients and services. Typically, the information published in the externally available DNS is the minimal needed for the Internet services such as mail, web and gateway systems such as VPNs. The separate internal DNS service typically provides a more rich information set typically needed by internal clients.

Rationale:
The two goals of Split-Horizon are to:

1. Minimize the amount and type of externally available information.

2. Physical and logical separation of external and internal DNS services.

Separating the external and internal DNS servers in this manner adheres to a defense-in-depth approach that limits the potential damage and impact should the external name server be compromised, since it does not service internal clients, nor does it have information on the internal systems and services.

Remediation:
Implement Split-Horizon Architecture to separate external and internal DNS services.

Audit:
Not Applicable

Default Value:
Not Applicable

References:

1. "SANS Institute - DNS Security Considerations and the Alternatives to BIND," https://www2.sans.org/reading_room/whitepapers/dns/567.php.

1.1.3 Slave DNS servers (Level 1/Not Scorable)

Description:
Slave DNS servers are set up to replicate the master servers data. The Slave, along with the master, is also authoritative for its domain, but it gets all of the domain data from the master. Slaves are updated from the master when the version of the domain data changes.

Rationale:
Every DNS architecture should have at least two name servers; a master and one or more slave name servers. There is no easy formula to calculate the number of name servers needed, as it depends on several factors, including; the number of sub-domains, the volume of requests and the geographical distribution of the traffic. For Internet-facing DNS services, at least one slave name server should be geographically remote from the master and connected via a different Internet connection to mitigate DoS attacks, to increase reliability, and to better distribute the DNS traffic. Many ISPs provide inexpensive name server services. Ask about the location and connectivity of the services, and look for one that is remote from your master, as well as reliable and secure. Also, consider getting a remote dedicated server that can be secured and administered remotely by your staff. For large organizations there are several companies specializing in globally distributed and high reliability DNS services. Likewise, your internal clients and servers should be configured to query multiple caching and/or forwarding name servers for increased reliability. These redundant name servers should be an integral portion of your network's architecture for reliability and performance and that places a priority on the DNS security.

Remediation:
Verify appropriate slave DNS servers are in use for each external and internal master name server. For the external Internet DNS servers an independent Internet connection or independent ISP hosted DNS service should be used.

Audit:
Not Applicable

Default Value:
Not Applicable

References:

1. "SANS Institute - DNS Security Considerations and the Alternatives to BIND," https://www2.sans.org/reading_room/whitepapers/dns/567.php.

1.2 Validate Name Registration Security (Level 1/Not Scorable)

Description:
The Name Registration process is where you register your domain and name server so you are part of the DNS system. There are many authorized name registration providers available, and the security of your name registration depends on their process for authenticating registration change requests. To obtain a list of DNS registrars, see www.iana.org, the Internet Assigned Numbers Authority.

Rationale:
If an attacker can take control of your name registration, then there is no need for her to compromise, spoof or otherwise subvert your DNS services, when she can have all of the DNS requests redirected to the DNS servers of her choice. In the past, name registration changes could be easily spoofed by sending an e-mail from the proper address. These days most registrars have raised the bar somewhat by requiring a response via e-mail, or they require that an administrator log onto a web site with a password. Most registrars also have fallback processes in place for handling cases where the contact e-mail is not working. It is highly recommended that you check with your registrar and review the authentication process, including alternative authentication options. In the balance between ease of use and reliable authentication, most registrars seem to heavily favor the ease of use. Many registrars also provide stronger authentication controls which are not required by default, but that are available upon request.

Also consider who is registering the domain. If an organization commissions a site through a third party it should ensure that the registration is assigned to someone within the organization, rather than the commissioned party.

Remediation:
Verify and document what security controls are in place for changes to your DNS registration and who is authorized to make changes. Also, regularly check the results of a "whois" query and verify that the information is correct.

Audit:
Not Applicable

Default Value:
Not Applicable

References:

1. "IANA — Internet Assigned Numbers Authority," http://www.iana.org/.

1.3 Secure DNS service operating platform (Level 1/Not Scorable)

Description:

The need for a hardened system is critical to the reliability of the DNS service. This system should be dedicated to running the minimum services required to support DNS.

Rationale:
Since your organization's DNS security is critical to the network services your organization depends on, it needs a dedicated security hardened system with minimal services running. You must also verify that the system is fully patched to prevent attackers from taking advantage of known vulnerabilities in other services. If you combine your DNS server with other services running on the same system, you are aggregating the risk of compromise associated with the additional services.

Remediation:
Harden the operating system, physical location and hardware according to your organization security policies. Make use of the additional controls recommended in the appropriate CIS benchmark for your platform. Run the BIND scoring tool and evaluate the results. All unnecessary services should be disabled, especially high-risk services such as web, mail, FTP, RPC services and file shares such as NFS and SMB.

Audit:
Not Applicable

Default Value:
Not Applicable

References:

1. "Center for Internet Security - Standards," http://www.cisecurity.org.

1.4 Verify Security of Forwarding Partners (Level 1/Not Scorable)

Description:
Forwarding queries to another name server allows the name service work to be aggregated and may improve performance if a name server is able to take advantage of the cache of an up-stream name server. This may also be a security weakness by relying on servers outside the organization's control and security policies.

Rationale:
One thing to consider when forwarding DNS requests is how secure the server is that is set up as the forwarder. A common recommendation is to use an ISP provided name service that is intended to help performance, and simplify network configuration. However, the down-stream name servers inherit the risk of the name server to which they forward queries. If the up-stream name server is compromised, or has its cache poisoned, then all the name servers that rely on it share the same fate. Forwarding is not necessarily a bad practice, but you need to evaluate the security and risk of the name servers to which you are forwarding queries. If these name servers are not under your organization's control, then evaluating their security can be difficult unless the provider specifically understands security as an integral and very necessary part of its service. In addition, you should be aware that if your forwarders are BIND version 4 or 8 servers, this may leave you

vulnerable to the DNS cache poisoning attacks discussed in the next section on BIND versions.

Remediation:
Verify your DNS architecture forwards queries only to trustworthy DNS servers and verify the security of those servers against appropriate security standards such as the CIS BIND Benchmark.

Audit:
Not Applicable

Default Value:
Not Applicable

References:

1. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).
2. "DNS for Rocket Scientists - Contents," http://www.zytrax.com/books/dns/.

2. Installing BIND (Berkeley Internet Name Domain)

This section gives information pertaining to the installation of BIND.

2.1 Obtaining BIND

The three major versions of BIND available at the time this benchmark was written are 4, 8 and 9. Version 4 of BIND has been officially deprecated by ISC ("Internet Systems Consortium") www.isc.org, and should not be used. In addition, BIND versions 4 and 8 are declared unsuitable for use as a forwarder. Using versions 4 or 8 as a forwarder allows for the classic Kashpureff-style DNS cache corruption attack. Although the usage of BIND 4 and 8 as forwarders has been problematic mostly for Windows DNS servers, it is still not recommended for BIND DNS servers. The original 1997 CERT Advisory is available from http://www.cert.org/advisories/CA-1997-22.html. More information on the forwarders vulnerability was posted by the SANS Internet Storm Center in April 2005. (http://isc.sans.org/diary.php?date=2005-04-07) ISC maintains a security matrix on their website to allow quick cross-reference of vulnerabilities to various BIND versions https://www.isc.org/node/356. BIND version 9 is recommended by this benchmark and will serve as the focus for the remainder of this document.

2.1.1 Secure Installation via ISC Source (Level 1/Not Scorable)

Description:
The latest release of the BIND software may be downloaded from the Internet Systems Consortium at https://www.isc.org/downloadables/11. Most vendors also supply packages for their operating systems.

Rationale:
Using vendor provided packages is recommended for most situations, as it will save effort with respect to support as well as obtaining patches and newer versions as they are

released. However, building from source is suitable for those that want full control of the build process, prefer to build from source, or do not have a suitable package available for their platform.

Remediation:
To obtain the BIND source, download the software and detached PGP signature from a suitable mirror site by following the download links on the ISC web site. After obtaining, importing, and verifying the ISC PGP key from a different source, verify the detached PGP signature. Untar the source and perform the usual configure, make, test and install steps.

```
cd /usr/local/src # <or suitable src directory>
Note that x.x.x represents the bind version number
wget http://... suitable mirror .../bind9/x.x.x/bind-x.x.x.tar.gz
wget http://... suitable mirror .../bind9/x.x.x/bind-x.x.x.tar.gz.asc
gpg --recv-key 0xc3755ff7 <or current key id>
(and/or)
wget http://oldwww.isc.org/about/openpgp/pgpkeyxxxx.txt
gpg --import pgpkeyxxxx.txt
gpg --verify bind-x.x.x.tar.gz.asc
gpg: Signature made Mon 07 Mar 2005 10:49:12 AM EST using DSA key ID
C3755FF7
gpg: Good signature from "Internet Systems Consortium, Inc. (Signing
key, 2004) <pgpkey2004@isc.org>"
tar xzf bind-x.x.x.tar.gz
cd bind-x.x.x
```

Common options to the configure script include `-with-openssl` and `-with-randomdev`, but the configure script will detect OpenSSL and `/dev/random` correctly on most systems. Only the make install should be done with root privilege. Information and references for setting up `sudo` or the appropriate vendor-specific tool is available in the CIS benchmark for the corresponding platform.

```
./configure <add optional configure options>
make
make test
<<sudo or equivalent>> make install
```

Audit:
Not Applicable

Default Values:
Not Applicable

References:

1. "BIND | Internet Systems Consortium," https://www.isc.org/downloadables/11.

- *Secure installation via Linux RPMs (Level 1/Not Scorable)*

Description:
RPM is the Red Hat Package Manager used on some Linux systems. It is an open packaging system available for anyone to use and is used by some distributions. RPM allows users to take source code for software and package it into source and binary form so that binaries can be easily installed and tracked and binaries can be rebuilt easily. It also keeps a database of all packages and their content that can be queried for information about files and/or packages.

Rationale:
Be careful with the vendor packages, as there are likely to be duplicate packages with different roles or architecture options. For example, typical RedHat RPMs include the list below. Having more than one of these installed is normal, but you need to be aware of what is installed so that they are updated properly, and minor overlaps such as with the `named.conf` file may lead to confusion. Installing the RPM is typically an option at installation time and may be done post-install with `yum` or `up2date`.

Red Hat RPMs:

- `bind` (The vanilla flavor)
- `bind-sdb` (with Simplified DB backend)
- `bind-chroot` (chrooted directory environment)
- `caching-nameserver` (a caching only configuration)
- `bind-libs` (required shared libraries)
- `bind-utils` (client utilities such as dig, host, nslookup, nsupdate)

Remediation:

```
# yum install bind
 (or)
# up2date bind
```

Audit:
The `rpm –checksig` option can be used to check signatures for an RPM package and `–verify` will verify the installed files against their original attributes.

Default Value:
Not Applicable

Reference:

1. "Linux Online - Introduction," http://www.linux.org/docs/ldp/howto/RPM-HOWTO/intro.html.

2.1.4 Secure installation on Solaris 10 (Level 1/Not Scorable)

Description:

BIND 9 is available and supported by default in the Solaris 10 Operating System ("Solaris 10 OS").

Rationale:
SUN provides support for the Solaris 10 OS and should be used for maintaining the core applications like BIND.

Remediation:
The software is found in the package, `SUNWbind`, and the Solaris 10 Service Management Facility ("SMF") service manifest for the BIND service is available in the SUNWbindr package. Both of these packages must be installed in order to be able to configure and run the DNS server service. The SMF Fault Management Resource Identifier ("FMRI") for the BIND service is `svc:/network/dns/server:default`. It is with this handle (or a suitable abbreviation such as dns/server) that one will be able to configure and manage the DNS server service using SMF. For example:

```
# svcs dns/server
STATE STIME FMRI
disabled 15:57:11 svc:/network/dns/server:default
# svcadm enable dns/server
```

Audit:
Starting with Solaris 10, each of the ELF objects, including binaries and libraries, delivered in the core OS has been cryptographically signed by Sun. Using the `elfsign(1)` tool, administrators can validate that individual ELF objects are genuine. For example, using the following code, an administrator can validate the ELF objects delivered in the `SUNWbind` package:

```
for file in `nawk '$NF == "SUNWbind" && $2 == "f" { print $1 }' \
      /var/sadm/install/contents`; do
      if [ `file ${file} | grep -c "ELF "` != 0 ]; then
            elfsign verify -e ${file}
      fi
done
```

The code above should generate output such as:

```
elfsign: verification of /usr/lib/dns/libdns.so.16.0.0 passed.
elfsign: verification of /usr/lib/dns/libisc.so.7.1.5 passed.
[...]
elfsign: verification of /usr/sbin/host passed.
elfsign: verification of /usr/sbin/named passed.
elfsign: verification of /usr/sbin/nslookup passed.
[...]
```

It should be noted that this audit does not work with OpenSolaris.

In addition to the use of the `elfsign(1)` command, administrators can also still verify the MD5 fingerprints of Sun provided programs, libraries, etc. using the Solaris Fingerprint Database available from http://sunsolve.sun.com/.

Default Value:
Not Applicable

References:

1. "SunSolve," http://sunsolve.sun.com/.

2.2 Run BIND as a non-root user (Level 1/Scorable)

Description:
To start BIND in Linux you must execute it as the root user. For Solaris, this is not necessary as it can be started as any user with the appropriate privileges for proper operation. BIND has the ability to change users, allowing it to drop the root privileges.

Rationale:
The reason for configuring BIND to run as a non-root user is to limit the impact in case a future vulnerability is discovered and exploited. This is a common practice, which implements the principal of least privilege. This principle states that an entity, such as a service or user, should be granted only those specific privileges necessary to perform authorized actions. The server will still need to be started as root, but it should be configured to give up the root privilege after listening on port 53. The user `named` runs as needs to be created if it does not already exist and needs appropriate access to the DNS configuration and data files. Many systems including Red Hat Linux will come with a `named` user already created. Usage of the user and group id of 53 in the examples is arbitrary but is intended to be easier to recognize as it matches the listening port number.

Remediation:
Create `named` user and group if it does not already exist. Using a shell of `/dev/null` is best practice.

```
if ! id named; then
groupadd -g 53 named
useradd –m -u 53 –g 53 -c "BIND named" -d /var/named -s /dev/null \
named
fi 2>/dev/null
```

In Solaris, consider using `roleadd(1M)` to create the `named` account instead of using `useradd(1M)`. Creating the account as a role adds protections should a password ever be installed: (1) the account still cannot be accessed remotely – even with a correct password and (2) roles can only be assumed by those users who have been granted access.

Also, `usermod`(1m) dictates that UID's 0 through 99 are reserved for the operating systems use, therefore the use of the UID 53 will produce an error. To correct this, you should choose a UID that is outside of its reserved range. Also, it recommended that `/dev/null` not be used as a shell. Instead, the Solaris 10 benchmark recommends that the shell be set to `/usr/bin/false`, and the account be set to locked (`passwd -l`) or non-login (`passwd -N`).

```
if ! id named; then
groupadd -g 153 named
roleadd –m -u 153 –g 153 -c "BIND named" -d /var/named –s \
/usr/bin/false named
passwd –l named
fi 2>/dev/null
```

In Linux, add or verify that `-u named` option is included in the `rc` startup script. For example, the `rc` script is `/etc/rc.d/init.d/named` and should contain the line "`daemon /usr/sbin/$PROG -u named ${OPTIONS};`"

Red Hat / Fedora Core Linux with Bastille

If you use the Bastille-Linux security hardening, the script will ask, if it detects BIND is installed, "Would you like to chroot named and set it to run as a non-root user?"

Answering *Yes* to this question can help automate this step as well as the chroot process, however you still need to verify all of the permissions and verify that the chroot directory contains the necessary devices, directories and files.

Solaris 10

Solaris 10 uses a different approach to configure the BIND service to run as a non-root user. It should first be noted however, that the BIND service in Solaris 10 is already configured to run with significantly reduced privileges. This is accomplished with Solaris 10 `privileges(5)` and configured using SMF. To see what privileges are granted to BIND at service startup, use the following command:

```
$ svcprop -p start/privileges dns/server
basic,!proc_session,!proc_info,!file_link_any,net_privaddr,
file_dac_read,file_dac_search,sys_resource
```

Essentially, this means that even if the UID of the BIND service is still 0 ("root"), it will not start with all of `root`'s privileges. Instead, it will only be granted the `proc_fork`, `proc_exec`, `net_privaddr`, `file_dac_read`, `file_dac_search`, and `sys_resource` privileges. `proc_fork` and `proc_exec` come from the basic privilege listed above. Certainly, this is already a significant reduction in the privilege set granted to the BIND service. Further, during the testing for this article, a simple configuration was created that only used proc_fork and

net_privaddr. As a rule, you may be able to further reduce the default set of privileges depending on your configuration and requirements. To see more detailed information about the privileges, see `privileges(5)`. To change the UID used to start the BIND service, use the `svccfg(1M)` command to configure the service as follows (assuming the `named` account and group have been created in a manner similar to that described above). Note that these steps only change the UID used by the process and does not impact the privileges that have been defined in the start/privileges property.

```
# svccfg -s dns/server:default setprop start/user = named
# svccfg -s dns/server:default setprop start/group = named
# svcadm refresh dns/server
```

There is one additional step that must be taken to successfully start the server. Depending on your configuration you may need to adjust some file paths or even file ownership or permissions to ensure that the `named` account can read or write files as appropriate (since the service is no longer starting as root at all). For example, an options flag must be added to the `/etc/named.conf` file to direct the `pid-file` parameter to a location writable by the `named` user or group. This is necessary since the `named` service is no longer starting as root and therefore will not be able to write to `/var/run`. Technically speaking, the pidfile is no longer used in Solaris since SMF is used to manage service administration, including restarts. For a more detailed example describing how to configure SMF services to start with reduced privileges, see the Sun BluePrint titled "Limiting Service Privileges in the Solaris 10 Operating System", available from: http://www.sun.com/blueprints/0505/819-2680.pdf . Once these changes have been made, the service can be started using the command:

```
# svcadm enable dns/server
```

The status of the dns/server service can be verified using the following commands:

```
# svcs dns/server
STATE STIME FMRI
online 18:43:01 svc:/network/dns/server:default
# pcred `pgrep named`
5842: e/r/suid=53 e/r/sgid=53
# ppriv -S `pgrep named`
5842: /usr/sbin/named
flags = <none
E:
file_dac_read,file_dac_search,net_privaddr,proc_exec,proc_fork,sys_resourc
I:
file_dac_read,file_dac_search,net_privaddr,proc_exec,proc_fork,sys_resourc
P:
file_dac_read,file_dac_search,net_privaddr,proc_exec,proc_fork,sys_resource
L: zone
```

If you are interested in fine tuning which privileges you want Solaris to grant each service, more information can be found in the BluePrint titled "Privilege Debugging in the Solaris 10 Operating System" found at http://www.sun.com/blueprints/0206/819-5507.pdf

Audit:
In Linux, verify that `-u named` option is included in the `rc` startup script. For example, the `rc` script is `/etc/rc.d/init.d/named` and should contain the line "`daemon /usr/sbin/$PROG -u named ${OPTIONS};`"

On Solaris, execute the following command to determine the user context the `named` service is executing under:

```
# pcred `pgrep named`
5842: e/r/suid=53 e/r/sgid=53
```

If the `suid` or `sgid` values are `0` then `named` is executing as `root`.

Default Value:
On Solaris, `named` executes under the context of `root:root`. However, it does operate with all of `root`'s privileges.

On Red Hat/Fedora, the `named` service executes under the context of the `named` user.

References:

1. "819-2680.pdf (application/pdf Object)." http://www.sun.com/blueprints/0505/819-2680.pdf.
2. "819-5507.pdf (application/pdf Object)." http://www.sun.com/blueprints/0206/819-5507.pdf.
3. "Solaris 10 Benchmark v.4.0." The Center for Internet Security, September 28, 2007. http://www.cisecurity.org/.

2.3 Isolate BIND via chroot or Solaris Zones

This section describes the process for isolating a BIND server from access to the entire system using a chroot jail or Solaris zones.

2.3.1 Using chroot with no packages (Level 1/ Scorable)

Description:
The `chroot` command allows you to create a running environment that can be severely limited from the rest of the operating system. The `chroot` command does this by running a command with the root file structure replaced by a given sub-directory. When this is done, the command executed by `chroot` no longer has access to the entire file structure but is limited to the given sub-directory. It should be noted that chroot-ing is not a completely sound security measure. Chrooting was not originally designed to be used for security.

There are ways that a chroot jail can be broken, such as file-system links pointing outside this jail. Most methods require that a process be running as root in order to escape. BIND can be run as a different user than root, as we discuss in section Restricting BIND Access.

Rationale:
Use the `chroot` command to further limit potential damage from a successful exploit, the server should be running in an isolated compartment such as a chroot-ed jail or a Solaris 10 zone. This way the daemon will be restricted in terms of what it can see or do. For example, in a chroot-ed jail, the service will not have access to the full file system, but instead a minimal file system with just the necessary data, libraries and devices. Either chroot or a restricted zone is required as a defense-in-depth measure even if the system is a dedicated DNS server as indicated section 1.3. The good news is that BIND 9 has been improved to be significantly easier to chroot than previous versions. This fact, combined with some great vendor tools to further simplify and improve the process, there no longer remains an excuse for any BIND implementation not to be chroot-ed. Chroot-ing is extremely important to the basic security of BIND and should not be overlooked or skipped. There are four solutions given below. The first, a generic chroot solution working directly with source code, should work with minor modifications for most Unix systems. The second is a simple implementation using the RedHat `bind-chroot` RPM, the third uses the Linux-Bastille script. Finally, the fourth option leverages the Solaris 10 zones capability to implement isolation rather than through the creation of a chroot jail.

Remediation:
Create the chroot directories. Details on the directories usage and permissions are important and are provided in section Restricting BIND Access. It is possible that some systems may also require some run time libraries within the chroot file hierarchy. Typically, BIND 9 does not require extra libraries as it performs the `chroot(2)` call later in the startup process. Check your system's dynamic linking man pages (such as `ld` and `ldd`) for additional information.

```
mkdir -p -m 750 /var/named/chroot
cd /var/named/chroot
mkdir -p -m 750 etc dev var/named/data var/run var/log var/tmp
```

Create the necessary devices such as `/dev/null`, `/dev/zero` and `/dev/random` using the proper major and minor device numbers for your platform. Using the command `ls-al` will show you what the major and minor numbers are for each file. A long listing of /dev/null, `/dev/zero` and `/dev/random` will provide the required major and minor device numbers for your platform. You should also create a syslog socket for logging by adding an option such as "`-a /var/named/chroot/dev/log`" to your `syslogd` command line. The logging section 5.3 describes how to configure syslog.

```
ls -al /dev/null /dev/zero /dev/random
mknod dev/null c ? ?
mknod dev/random c ? ?
```

```
mknod dev/zero c ? ?
chown root:named dev/null dev/zero dev/random
chmod ug=rw,o= dev/null dev/zero dev/random
cp /etc/named.conf etc/
cp /etc/localtime etc/
```

Edit `/etc/named.conf` file to match the created paths for chroot directory `/var/named`, `/var/named/data` and `/var/run/`. Also, copy the appropriate zone files referenced in your `named.conf` file.

Audit:
On Red Hat/Fedora, ensure `ROOTDIR` token in `/etc/sysconfig/named` is defined and points to the chroot jail location.

Default Value:
Not Applicable

References:

1. "Building and configuring BIND 9 in a chroot jail," http://www.unixwiz.net/techtips/bind9-chroot.html#mkjail.

2.3.2 RedHat bind-chroot Rpm (Level 1/Scorable)

Description:
RedHat Linux systems have a `bind-chroot` RPM containing a directory environment for running BIND in a chroot-ed file system. This is implemented by installing the RPMs below or even checking off BIND during the install process.

Rationale:
See Using chroot with no packages

Remediation:
1. Install the following RPM's are installed:

    - bind
    - bind-chroot
    - bind-libs
    - bind-utils

2. Directory permissions should be hardened, as they are not secure by default; they are detailed in section 2.4.1.

3. A syslog socket needs to be created. Add "`-a /var/named/chroot/dev/log`" to `SYSLOGD_OPTIONS` in `/etc/sysconfig/syslog`. See section 5.3 for details on configuring `named` to use syslog. If a local log file is to be used, create a `/var/log` directory in `/var/named/chroot`.

4. Add "ROOTDIR=/var/named/chroot" to /etc/sysconfig/named

Audit:

1. Validate the bind-chroot package is installed:

```
yum list bind-chroot | grep Installed
```

   If the above command yields no output, bind-chroot is not installed.

Default Value:
Not Applicable

2.3.3 Solaris 10 Zones (Level 1/Not Scorable)

Description:
It is possible to use the previously mentioned chroot methods in Solaris 10. However, Solaris 10 offers an additional option for isolating BIND servers. Solaris 10 Zones allow BIND to be run in a restricted environment (e.g. local zone) which offers significant protection from attack. In particular, services, processes, and users running within local zones inherently have a reduced set of privileges. This is because a local zone is viewed as a virtualized application environment – a safe container within which services can be deployed.

Rationale:
Users and processes running within a zone are severely restricted in what they are able to do. For example, they are not able to:

- Load or unload kernel modules;
- Access raw memory, devices, or networking;
- View or access users, services, or devices that exist outside of the local zone itself;
- Change system, network, or resource configurations including processor sets, networking interfaces, routing tables, system time, etc;
- Alter files under /lib, /platforms, /sbin, or /usr (by default);

All actions that occur within a local zone can be monitored from the global zone (through kernel event auditing, application log monitoring, file integrity checks, etc,) which is a significant improvement over using the global zone for service deployment. If the global zone is used, an attacker that has root or equivalent privilege would be able to view and modify the audit records or logs on the system. In addition to auditing, if extended process accounting has been configured and enabled from the global zone, then the accounting data can still be archived for the local zone even if the attacker turns off process accounting within the zone (which in fact only turns off a local accounting stream).

Remediation:
Use the following commands to create a local zone that is able to support the BIND service. It is assumed that the SUNWbind and SUNWbindr packages are already installed in the global zone. Be sure to use the appropriate path to the root of the local zone, the name of the

physical interface to be used virtually, and the IP address to be assigned to the virtual interface

```
# zonecfg -z bind
bind: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:bind> create
zonecfg:bind> set autoboot=true
zonecfg:bind> set zonepath=<<zonepath>>
zonecfg:bind> add net
zonecfg:bind:net> set physical=<<physical-device>>
zonecfg:bind:net> set address=<<network-address>>
zonecfg:bind:net> end
zonecfg:bind> verify
zonecfg:bind> commit
zonecfg:bind> exit
```

A local zone called "bind" is now configured. There are other options that can be configured depending on your specific needs. Now that the zone has been configured, the following command will install it.

```
 # zoneadm -z bind install
Preparing to install zone <bind>.
Creating list of files to copy from the global zone. Copying <2574>
files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <987> packages on the zone.
[...]
Initialized <987> packages on zone.
Zone <bind> is initialized.
The file </export/zones/bind/root/var/sadm/system/logs/install_log>
contains a log of the zone installation.
```

With the zone installed it can be booted and any final configuration can be completed using the following commands.

```
# zoneadm -z bind install
# zlogin –C bind
```

Note that the final `zlogin(1)` command attaches to the console of the zone so that any final configuration steps can be completed. In most cases, these configurations can also be completed using a `sysidcfg(4)` file installed as `/etc/sysidcfg` in the local zone.

With the bind zone ready, the steps presented earlier in section 2 can be used for creating a `named` role and group, configuring the service to run as a non-root user, etc.

Audit:
Not Applicable

Default Value:
No named-specific zone is present by default.

References:

1. "820-7017.pdf (application/pdf Object)." http://www.sun.com/offers/docs/820-7017.pdf.

2.4 Restricting BIND Access

Restricting the files and directories for which BIND has read or write access is another layer of security that should be added for a name server. This should be done with the traditional Unix permissions, but may also be accomplished with mandatory access controls implemented by SELinux (Secure Enhanced Linux) and with non-discretionary role based access controls using Solaris 10 Zones (not to be confused with DNS zones). In addition to restricting file and directory access, SELinux and Solaris 10 Zones have additional controls described in the following sections for each. First, we define the restricted access goals, which are independent of the implementation, and then we provide implementation specific details. The usage of SELinux or Solaris 10 zones offer protections beyond simply setting the Unix permissions, and is recommended as a defense-in-depth measure. As many can attest from experience when a service is not working it is often these restrictive security controls that are suspected and disabled in order to debug the problem. The issue is that once the problem is fixed the security controls may not be put back in place. This is a good reason to re-run the score tools periodically to audit your system and find unexpected problems.

BIND Restricted Access Goals

The BIND service, that runs as the account `named`, should not have write access to any of the configuration files, key files or any directories in which the files are contained, nor any parent directories. Write access to these directories and files needs to be restricted to authorized administrators and operators only.

Only the BIND account, `named`, and administrators should have read access to the BIND configuration and keys files. Other users should have no access to these files.

The BIND service should only have read access to the BIND working directory, and all parent directories. It is important to check the access on parent directories, as it could allow directories to be renamed and recreated with inadequate restrictions. To be safe, minimal access requires that the parent directories need to be checked all the way up to the actual root directory, rather than just the chroot-ed directory, if used.

The master zone files and the directory in which they are stored should have read only access by the `named` account. The parents to the directory should also be included to have

read only access, although they are typically the same directories checked in the previous item.

The master zone files should not allow read access by users other than the `named` account and BIND administrators.

All of the directories within the BIND `chroot` should have read-only access except the following:

- `var/run` - May be used for pid file
- `var/log` - May contain local log files
- `data` - May be used for dump files.
- `ddns` - If used for Dynamic DNS updates
- `slave` - For receiving slave zones.

If chroot is not used, the same should apply to the real root, except that standard writable directories that also have the sticky-bit set, such as `/tmp` and `/var/tmp` are also excluded.

2.4.1 Set permissions on BIND chroot-ed directories (Level 1/ Scorable)

Description:
The following actions are specific to chroot-ed BIND directories and have been tested on both the build from source instructions given, and the Red Hat bind-chroot RPM.

Rationale:
You should not assume the vendor provided permissions are secure. The permissions set for the chroot are a bit more restrictive than those outlined in beginning of this section, as other access is mostly zero (no access). However since other users have no need to read the BIND chrooted directories, it is a prudent application of the principle of least privilege. In the instructions `$ROOTDIR` refers the chroot-ed directory. Any directories beyond the expected directories (`etc`, `dev`, `proc` and `var`) should be reviewed for appropriate minimal access.

Remediation:
Check that no parent directories to the chroot are writable by the `named` user. The following command will check every parent directory except `/`. Any writable directories need to be corrected. If the command echo's the directory names without reporting any directories as writable, then the permissions are ok.

```
cd $ROOTDIR
su -m named -c 'D=$PWD; while [ "$D" != "/" ]; do echo $D;
test -w $D && echo $D is writable.; D=`dirname $D`; done'
```

First, set the ownership and permissions on the chroot directory.

```
```

```
chown root:named $ROOTDIR
chmod u=rwx,g=rx,o= $ROOTDIR
```

Change everything to be owned by `root`, read-only by `named`, no access for other.

```
cd $ROOTDIR
chown -R root:named etc var
chmod -R g-w,o= etc var
chown root:named dev proc
chmod g,o=rx dev
chmod a=rx proc
```

Next, add the minimal write access to the necessary directories and any files in the directories. Create and `chmod` the slave directory only if slave zones are configured. Likewise only create and `chmod g+w` the `ddns` directory only if dynamic updates are configured.

```
chmod -R g+w var/run/named var/tmp var/log var/named/data
chmod -R g+w var/named/slaves
chmod -R g+w var/named/ddns
```

Audit:
Ensure that only the minimal directories are writeable with the following command.

```
find / -user named -type d -perm -222
```

Default Value:
Not Applicable

References:

1. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).

2.4.2 Restrict BIND Access with SELinux (Level 1, Scorable)

Description:
The Security Enhanced Linux (SELinux) project started by the NSA provides targeted mandatory access controls, which may be used to restrict BIND to minimal access. SELinux is included in the RedHat Enterprise and Fedora Core installation options. Make sure you have the latest versions of following RPM's, or install them if needed.

- `libselinux-devel`
- `libselinux`

- selinux-policy-targeted
- selinux-policy-targeted-sources
- selinux-doc
- checkpolicy

Rationale:
See section Set permissions on BIND chroot-ed directories.

Remediation:
Make sure you have the latest versions of following RPM's, or install them as needed.

- libselinux-devel
- libselinux
- selinux-policy-targeted
- selinux-policy-targeted-sources
- selinux-doc
- checkpolicy

```
sudo yum install libselinux-devel libselinux selinux-policy-targeted
selinux-policy-targeted-sources selinux-doc checkpolicy
sudo setenforce Enforcing

Edit the file /etc/selinux/config or use the commands below to set the
values for
SELINUX to enforcing and SELINUXTYPE to targeted, to ensure that
SELinux is
enabled, enforcing and is in targeted mode after each system reboot.
See man pages
setenforce(8) and sestatus(8) for details.
sudo cp -p /etc/selinux/config /etc/selinux/config.orig
sudo cat > /etc/selinux/config
sudo echo "SELINUX=enforcing" > /etc/selinux/config
sudo echo "SELINUXTYPE=targeted" >> /etc/selinux/config
```

Edit the file `/etc/selinux/config` or use the commands below to set the values for SELINUX to enforcing and SELINUXTYPE to targeted, to ensure that SELinux is enabled, enforcing and is in targeted mode after each system reboot. See man pages `setenforce(8)` and `sestatus(8)` for details.

There are two booleans associated with the `named` targeted policy, `named_disable_trans`, which disables the `named` policies if set, and `named_write_master_zones`, which allows the `named` to write its master zones if set. Neither of these booleans should be set.

```
sudo setsebool named_disable_trans=0
sudo setsebool named_write_master_zones=0
```

```
```

Reboot the system now to ensure that SELinux is enforced in targeted mode.

Audit:
Ensure that the following commands output values corresponding to the remediation above.

```
sudo grep "SELINUX=enforcing" /etc/selinux/config
sudo grep "SELINUXTYPE=targeted" >> /etc/selinux/config
sudo getsebool named_disable_trans
sudo getsebool named_write_master_zones
```

Default Value:
Not Applicable

References:

1. "Security-Enhanced Linux," http://www.nsa.gov/selinux/.
2. "redhat.com | Taking advantage of SELinux in Red Hat Enterprise Linux," http://www.redhat.com/magazine/006apr05/features/selinux/.

2.4.3 Restrict BIND Access Within Solaris 10 (Level 1, Not Scorable)

Description:
The following instructions outline the changes that must be made to run the BIND service as a non-root account, `named`, within a Solaris 10 zone. REW

Rationale:
The changes discussed below are needed in order to permit the `named` account to write configuration, log or other related files. Note that when run within a Solaris 10 zone, the BIND service, by default, will not be able to write to any files or directories under `/usr`, `/lib`, `/sbin`, or `/platform` as their directory trees are mounted read-only from the global zone. Further restrictions are implemented using standard Unix permissions and ACLs. The changes noted below apply specifically to BIND directories and files.

Remediation:
Adjust the BIND configuration file, `/etc/named.conf`, to ensure that it does not refer to directories that are not writable by the `named` account. In Solaris 10, it will not be able to write to files such as `/var/log/named.log`. It is recommended that directories be created under the `/var/named` hierarchy (similar to what is accomplished in the chroot case). Therefore, references to `/var/log/named.log` would become `/var/named/log/named.log` and references to `/var/log/secure.log` would become `/var/named/log/secure.log`. Similarly, as noted previously, the pid-file BIND option should be configured to reference a writable directory as the service will not be able to create the file under `/var/run` (since it is no longer running as `root`). It is recommended that `/var/named/run` be used in this case.

```
# mkdir -p /var/named/data
# mkdir -p /var/named/log
```

Once these directories are created, ownership and permissions can be assigned as follows:

```
# chown -R root:named /var/named
# chmod 750 /var/named
# chmod 770 /var/named/data /var/named/log /var/named/tmp
```

Lastly, the permissions of the actual DNS zone files can be configured:

```
# chmod 640 <<DNS Zone Files>>
```

Audit:
Not Applicable

Default Value:
Not Applicable

3. Security Configurations

This section provides guidance on the secure configuration of BIND.

3.1 Hide BIND Version String (Level 1, Scorable)

Description:
The version string contains the version of BIND that is running.

Rationale:
Additional information hiding or obscurity can be provided by preventing the version information being returned to `TXT` queries to the pseudo-domain "version.bind" in the chaos class.

Remediation:
Place the following in the global options of `named.conf`:

```
options {
      version "None";
. . .
}
```

Audit:
From the command prompt, use `dig` to test the version string:

```
dig @localhost version.bind chaos txt | grep '^version.bind.'
version.bind. 0 CH TXT "None"
version.bind. 0 CH NS version.bind.
```

Default Value:
The default of version will report the version via a query of the name `version.bind`

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
2. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).
3. "DNS for Rocket Scientists - Contents," http://www.zytrax.com/books/dns/.

3.2 Restrict Queries

This section provides guidance on how to restrict queries in BIND.

3.2.1 Restrict Recursive Queries (Level 1, Scorable)

Description:
A recursive DNS query is your typical DNS query from a client. It places the burden of finding the answer on the DNS server which will recursively query other DNS servers authoritative for the domains, until it gets the answer which is then returned to the client. Typically, the DNS server will then cache the answer to that query until its time-to-live expires in order to provide a quick answer to future queries for the same name.

BIND can be configured to restrict fulfillment of recursive lookups to only authorized network segments and hosts. This is made possible by the `allow-recusion` option. It is recommended that this option be utilized to restrict access to the server's recursive lookup capabilities.

Rationale:
If the `allow-recursion` is not properly configured, malicious entities may abuse this capability for asymmetric load based denial of service attacks, associate the DNS server with SPAM campaigns, or increase the probability of poisoning the server's cache.

Remediation:
Insert the following either into the global options or to every zone section

```
allow-recursion { localhost; trusted_clients };
```

Audit:
From the command prompt, send a non-related request to the server from a client that is not permitted:

```
nslookup www.google.com ns1.example.com
```

```
nslookup google.com 192.168.0.1
*** Can't find google.com: No answer
```

Note: `allow-recursion` will not prevent a client from having access to data in the server's cache. This is because no recursion is required to look at the cache. Use `allow-query-cache` to prevent this from happening.

Default Value:
The default allows `localnets` and `localhost`.

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
2. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).
3. "SP800-81.pdf (application/pdf Object)," http://csrc.nist.gov/publications/nistpubs/800-81/SP800-81.pdf.

3.2.2 Restrict Query Origins (Level 1, Scorable)

Description:
BIND can be configured to limit restrict access to its query services. This is made possible by the `allow-query` option. It is recommended that this option be utilized to restrict access to the server's query services.

It is also recommended that caching-only servers limit all queries to only the expected internal networks by adding ACL's to define the allowed local networks and restrict recursive queries. Note: that `localhost` and `localnets` are BIND predefined ACLs, and should not be used for new ACL names.

Rationale:
Using `allow-query` in conjunction with an ACL of trusted clients will prevent unauthorized access to name services content. Additionally, the exposure of vulnerabilities present in BIND's query handlers is reduced by this configuration as requests originating from untrusted entities will be rejected before the request is fully parsed by `named`.

Remediation:
1. Create ACLs for the local networks in `named.conf`

```
acl "local" { 127.0.0.1; };
acl "mynets" { 10.1.2.0/24; 10.1.3.0/24; 10.1.4.0/24; };
. . .
```

```
```

2. Add the following to the `named.conf` global options:

```
options {
. . .
      allow-query { local; mynets; };
. . .
}
```

Audit:
Verify that the ACL that is used with `allow-query` is restricted to only the hosts that are allowed to have access.

```
$ grep allow-query /etc/named.conf  ; test $? -eq 0 && echo "allow-
query is active" || echo " allow-query was not found"

      allow-query { local; mynets; };
allow-query is active
$
```

Default Value:
The default allows queries from all hosts

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
2. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).
3. "SP800-81.pdf (application/pdf Object)," http://csrc.nist.gov/publications/nistpubs/800-81/SP800-81.pdf.

3.2.3 Restrict Access to Cache (Level 1, Scorable)

Description:
BIND can be configured to restrict access to its resolver cache. This is made possible by the `allow-query-cache` option. It is recommended that this option be utilized to restrict access to the server's cache.

Rationale:
Using `allow-query-cache` in conjunction with an ACL of trusted clients will prevent unauthorized access to cached content. Additionally, the exposure of vulnerabilities present in BIND's query handlers is reduced by this configuration as requests originating from untrusted entities will be rejected before the request is fully parsed by `named`.

Remediation:

1. Set up an ACL in `named.conf` containing clients that are allowed to query the cache.

```
acl "trusted" {
      192.168.0.0/16;
      localhost;
      localnets;
};
```

2. Set `allow-query-cache` and `allow-recursion` in the global options of `named.conf`

```
options {
      allow-recursion { trusted; };
      allow-query-cache { trusted; };
};
```

Note: In BIND 9.3 and earlier there was no discrimination between queries of a servers cache and of authoritative data. Upon the release of BIND 9.4 tighter control of queries was added through the "`allow-query-cache`" option.
When this feature was released, the default action was to permit all queries. This was found to be the cause of a vulnerability, along with allow-recursion, that allowed attackers to bypass security measures, obtain sensitive data from the server, and launch denial of service attacks. This was quickly patched in BIND 9.4.1-P1, and the behavior was changed to allow only localnets and localhost by default. The above suggestion solves for this problem regardless of your version of BIND.

Audit:
Verify that the ACL that is used with `allow-query-cache` is restricted to only the hosts that are allowed to have access.

```
$ grep allow-query-cache /etc/named.conf  ; test $? -eq 0 && echo
"allow-query-cache is active" || echo " allow-query-cache was not
found"

      allow-query-cache { trusted; };
allow-query-cache is active
$
```

Default Value:
The default allows only localnets and localhost to query the cache

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.

3.2.4 Do not use BIND9 Views for split horizons (Level 1, Scorable)

The views feature of BIND 9, allows BIND to present different information and restrictions for the same zone depending on the IP address of the client. They could be used to have an internal "view" with detailed information and an external "view" with minimal information presented depending on the source IP address of the request. Although this could be used to provide something similar to the split horizon implementation, it would do so without separating the server or even the BIND services, and would place trust in the source IP address which could be spoofed. Therefore using views to split internal vs. external DNS is not recommended. That is not to say there is no legitimate usage of views.

Description:
The views feature of BIND 9, allows BIND to present different information and restrictions for the same zone depending on the IP address of the client. They could be used to have an internal "view" with detailed information and an external "view" with minimal information presented depending on the source IP address of the request.

Rationale:
Using BIND views causes all services to run on one server and places trust in the source IP address which can be spoofed. Therefore using views to split internal vs. external DNS is not recommended.

Remediation:
Although the previous restrictions on recursion and queries can be done with a view, using the `allow-query` and `allow-recursion` options is the recommended approach.

Audit:
Search `named.conf` for a view statement that pertains to an internal and external view, such as:

```
view "external" {
      match-clients { any; };
};
```

```
$ grep view /etc/named.conf  ; test $? -eq 0 && echo "These are
your active views" || echo " No views were  found"
No views were found
$
```

Default Value:
BIND does not use views by default.

References:

1. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).
2. "SP800-81.pdf (application/pdf Object)," http://csrc.nist.gov/publications/nistpubs/800-81/SP800-81.pdf.

3.3 Transaction Signatures -- TSIG

Transaction Signature (TSIG) serves to authenticate the responses and updates sent to our own servers and is implemented by generating a secure hash of the DNS data combined with a shared secret. Since TSIG depends on a shared secret between the two DNS servers it is really only suitable for authenticating your organization's or possible partnering organization's DNS servers. There are two critical DNS functions for which using TSIG works well to provide authentication: zone transfers and dynamic updates. We will discuss both functions in detail in the next sections.

3.3.1 dnssec-keygen Algorithms (Level 1, Scorable)

Description:
The TSIG key is created using the `dnssec-keygen` tool that is included with BIND.

Rationale:
If zone transfers or dynamic updates are allowed, generate TSIG keys to authenticate the servers, one for each host-to-host trust relationship. Use of the MD5 hash is not recommended because it has been shown to be weaker than expected.

Remediation:
1. Use dnssec-keygen to generate the key using an algorithm from the SHA-2 family :

```
$ dnssec-keygen -a HMAC-SHA256 -b 256 -n HOST ns1-ns2.cisecurity.org
Kns1-ns2.cisecurity.org.+163+24158
```

2. You should name the key using the names of the two hosts to avoid confusion.
3. Do NOT use the HMACMD5 algorithm.

Audit:
Check the created .key and .private files that were created:

```
$ cat Kns1-ns2.cisecurity.org.+163+24158.key
ns1-ns2.cisecurity.org. IN KEY 512 3 163
Z98Q4i9XU0NtmYaWPiuso/RTjyR3wM3uZ8OH2vVU0EU=

$ cat Kns1-ns2.cisecurity.org.+163+24158.private
Private-key-format: v1.2
Algorithm: 163 (HMAC_SHA256)
Key: Z98Q4i9XU0NtmYaWPiuso/RTjyR3wM3uZ8OH2vVU0EU=
Bits: AAA=
```

Default Value:
There is no key generated by default.

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
2. "SP800-81.pdf (application/pdf Object)," http://csrc.nist.gov/publications/nistpubs/800-81/SP800-81.pdf.

3.3.2 Include TSIG key in named.conf (Level 1, Scorable)

Description:
Use the include option to add the generated key into `named.conf`.

Rationale:
Although the key configuration may be done directly in the `named.conf` file, putting it in a separate file will limit the number of times it needs to be viewed. It is essential that this secret be protected properly by having limited file permissions (same as the `named.conf`), and to be protected in transit to the other DNS server.

Remediation:
1. Ensure that the file is placed in the appropriate chroot-ed directory on each system.
2. Use the include option to have BIND import the key when started on each server.

```
include "/etc/ns1-slave1.key";
include "/etc/ns1-slave2.key";
. . .
```

3. Verify that the files permissions are limited.

```
ls -l /etc ns1-slave1.key
```

Audit:
Verify that the key has not been written directly in `named.conf`.

```
$ grep "include" /etc/named.conf  ; test $?  -eq 0 && \
echo "files included" && \
echo "Please verify that keys are correct" ; \
grep -q secret named.conf ; test $?  -eq 0 && \
echo "secret found in named.conf"

   include "/etc/ns1-slave1.key";
   include "/etc/ns1-slave2.key";
   . . .
   Please verify that keys are correct
```

If a secret is found in `named.conf`, relocate it to a separate key file and run the command again. Audit the key files permissions as shown in section 2.4.

Default Value:
Key files are not included by default

References:

1. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).
2. "SP800-81.pdf (application/pdf Object)," http://csrc.nist.gov/publications/nistpubs/800-81/SP800-81.pdf.
3. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.

3.4 Restrict Zone-Transfers (Level 1, Scorable)

Description:
A zone transfer, or `AXFR` request, is a mechanism commonly used by DNS deployments to replicate zone information from master/primary servers to slave/secondary servers. BIND can be configured to respond only to `AXFR` requests that originate from a predefined server. This is made possible by the `allow-transfer` option in `named.conf`. It is recommended that the fulfillment of zone transfer requests be reserved for only predefined slave/secondary servers.

Rationale:
A zone transfer is not something you want most systems to be able to request, as it would give out the entire list of resource records. There should be very few systems besides the slave name servers that need to be able to perform a zone transfer for your domains. Allow-transfer specifies who can request a zone transfer. Restrictions should not be made using IP address, but rather by using TSIG keys.

Remediation:
1. Add a similar configuration to `named.conf` on the master server:

```
options {
. . .
      allow-transfer { key "ns1-slave1_key"; key "ns1-slave2_key"};
. . .
}
```

2. Configure each slave to include its keys and add a statement to `named.conf` so that the slave will sign its communications to the master.

```
server 192.168.1.53 {
      keys { "ns1-slave1_key"; };
};
```

Note: The name of the key must match the name in the key file, and the key names and content must match between the master and slave servers as well.

Audit:
On a client that is not allowed to request zone transfers:

```
dig @ns1.example.com example.com axfr
```

If transfers are enabled you will see a list of resource records, and must recheck your configuration.

Default Value:
The default action of `allow-transfer` is to allow transfers to all hosts.

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
2. "SP800-81.pdf (application/pdf Object)," http://csrc.nist.gov/publications/nistpubs/800-81/SP800-81.pdf.
3. "Men & Mice | DNS, DHCP, IPAM, IPv6, DNS Monitoring, IP Address Management, DNSSEC, OSS," http://www.menandmice.com/knowledgehub/.

3.5 Restrict Dynamic Updates

3.5.1 Using Update Policy (Level 1, Scorable)

Description:
Dynamic updates are used to automate the updating of zones. Dynamic updates are typically used with DHCP, however updates may include deleting or adding any resource records of a zone except the SOA and NS records. Allowing other systems to make permanent updates to your zones is of course not allowed by default, and needs to be carefully restricted.

Rationale:
In the update from BIND 9.3 to 9.4, the option "`allow-update`" was changed and is now able to be defined in the global options of `named.conf`. Previous versions also only allowed the use of IP addresses, or a network prefix, whereas a match list can now be used.

The usage of the "`allow-update`" option with IP addresses is discouraged as the source address of the UDP packet can be forged and could lead to a compromise of data. Due to this, "`allow-update`" should not be used. If it is found to be necessary to use "`allow-update`," the use of TSIG authentication is encouraged as well as "`allow-update-`

forwarding" to specify which slaves to accept updates from. Again, this configuration is strongly discouraged as it allows compromised slave servers to push updates to the master.

Instead of the "allow-update" option, use "update policy" to grant "A" record updates to the dynamically updated sub-domain for the host specific TSIG key.

Remediation:

1. Add a similar statement to a server's zone statement using the appropriate TSIG key for the target.

```
include "ns1-dhcp_server1.key";
zone "dynamic.example.com" {
      type master;
. . .
      update-policy {
            grant ns1-dhcp_server1_key name dynamic.example.com A;
      };
};
```

2. Include a server statement

```
server 192.168.1.53 {
      keys { ns1-dhcp_server1_key; };
};
```

Audit:

Use nsupdate with the appropriate key to send a similar update to the server

```
% nsupdate -k Kns1-dhcp.example.+157+00000.private
> update add test12.example. 86400 IN TXT "Test"
> send
```

If properly configured, this will succeed.

Default Value:

There is no update-policy by default

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
2. "DNS, BIND Nameserver, DHCP, LDAP and Directory Services," http://www.bind9.net/.

3.5.2 Enable GSS-TSIG (Level 1, Scorable)

Description:
BIND 9.5.0 introduced support for the proprietary GSS-TSIG algorithm that is used by Microsoft's DNS solution. It is recommended that GSS-TSIG be utilized to integrate service with Microsoft DNS whenever possible.

Rationale:
GSS-TSIG provides support for authenticated transactions between BIND and Windows DHCP servers. This use of this mechanism reduces the probability of an attacker compromising the integrity of the DNS cache.

Remediation:
1. GSS-TSIG can be enable in BIND through the use of the following options in `named.conf`:

```
options {
...
        tkey-gssapi-credential   "DNS/dns.example.com";
        tkey-domain              "dns.example.com";
...
};
```

2. Create the Kerberos key and include it in a key statement and use the key in the zone statements:

```
key my-gss-key
{
      algorithm gss-tsig;
      key Secret_GSS_Key;
}
```

Audit:
Not Applicable

Default Value:
`gss-tsig` is not enabled by default.

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
2. RFC 3645

3.5.3 DHCID (Level 1, Scorable)

Description:
DHCID (DHCP Client Identifier) is a new resource record type that is used by DHCP servers in a fully qualified domain to uniquely associate domain names with the clients using them.

Rationale:
Configuration problems can occur when multiple clients try to use the same FQDN, or when multiple DHCP servers are used in the same FQDN. The use of DHCID resolves both problems by allowing DHCP servers to determine which client is associated with a FQDN.

Remediation:
DHCID was added in BIND 9.5.0. It is used in conjunction with a one-way hash to help protect the client's identity. However, the DHCID could still be vulnerable to a brute force attack, so using TSIG is recommended to help authenticate and verify the integrity of the data.

Audit:
Verify that TSIG is enabled on the server in `named.conf` by verifying that appropriate TSIG key was included:

```
$ grep "ns1-dhcp.cissecurity.key" /etc/named.conf ; \
test $?  -eq 0 && echo "files included" && \
echo " Please verify that keys are correct" \
|| echo " The key was not found." ; grep -q secret \
named.conf ; test $?  -eq 0 && \
echo "secret found in named.conf"

   include "/etc/ns1-dhcp.cissecurity.key";
   Please verify that keys are correct
$
```

If the key was not found, follow the above steps and include the key from a separate file.

Default Value:
DHCID is not used by default.

References:

1. RFC 4701
2. RFC 4703

3.6 Implement DNSSEC (Level 1, Scorable)

Description:
DNS Security Extensions or DNSSEC for short provides authentication of the name servers through public key cryptography. With DNSSEC, the name server signs its responses with its private key. This allows other name servers that have the public key of the name server to verify the integrity and authenticity of the response. DNSSEC also provides for signing of public keys so that delegated sub-domains may have their keys signed by a higher level authority. This creates a chain of trust so that any name server that trusts the public key of the higher level signing authority can trust the signed key. If a higher authority does not have the ability to certify a zone's DNSSEC records, such as many of the top-level domains,

then BIND can specify the public key that corresponds to a particular zone using the trusted-keys statement in `named.conf`.

The chain of trust that is created with DNSSEC is currently useful within an organization with delegated sub-domains, and progress has been made with regard to getting top-level domain authorities to have their keys authenticated and signed. Recently the .org top-level domain became the first to be authorized to implement DNSSEC.

Rationale:
DNSSEC reliably authenticates DNS responses to prevent the DNS spoofing and cache poisoning attacks.

Remediation:
1. Use the `dnssec-keygen` tool to create the key pair for your name server. The dnssec-keygen supports both RSA and DSA key types and a variety of key sizes. Keep in mind that signing the zone information greatly increases the information transferred; current estimates are about four to seven times larger, for the average zone. Therefore keeping the key length to a modest length of 512 bits may be helpful.

```
dnssec-keygen -n ZONE -a RSA -b 512 example.com.
```

2. Add a line, typically at the end, to your master zone file to include the key. Then generate the sign zone file. The `-o` option is the origin and is required, while the second `example.com` is the zone file name. The generated file will be `example.com.signed`.

```
echo '$INCLUDE Kexample.com.+001+13453.key' >> example.com
dnssec-signzone –o example.com example.com
```

3. Next, modify the `named.conf` to reference the signed zone file, and ensure that DNSSEC is enabled in the global options for it to be able to respond to DNSSEC aware clients. DNSSEC validation is set. Also, enable DNSSEC validation so that `named` can validate answers from other servers.

```
options {
...
      dnssec-enable yes;
      dnssec-validation yes;
};
```

4. Recheck the file permission

5. Reload `named`

6. The BIND servers that need to verify the signed records will need to have the generated public key added as a trusted key by placing the key file in the `etc` directory, and including the key file in a trusted-keys clause similar to below:

```
trusted-keys {
      example.com. 256 3 1 "AQPJpW8SZjNJKgq6. . . 7/cTxpeFpmrt1";
};
```

Audit:
Inspect the `named` logs and confirm the loaded zone (example.com) is signed. The following demonstrates this:

```
. . .
validating @0x1823e00: example.com SOA: starting
validating @0x1823e00: example.com SOA: attempting positive response
validating @0x1823e00: example.com DNSKEY: starting
validating @0x1823e00: example.com DNSKEY: attempting positive response
validating @0x1823e00: example.com DNSKEY: verify rdataset: success
validating @0x1823e00: example.com DNSKEY: signed by trusted key;
marking as secure
. . .
```

Note: Logging is covered in the Logging and Monitoring section.

Default Value:
The default of `dnssec-enable` is yes
The default of `dnssec-validation` is no

References:

1. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).
2. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
3. "Men & Mice | DNS, DHCP, IPAM, IPv6, DNS Monitoring, IP Address Management, DNSSEC, OSS," http://www.menandmice.com/knowledgehub/.
4. "DNSSEC - DNS Security Extensions," http://www.dnssec.net/.
5. "DNS for Rocket Scientists - Contents," http://www.zytrax.com/books/dns/.

3.7 Disable dnssec-accept-expired option (Level 1, Scorable)

Description:

Introduced in BIND 9.4, the `dnssec-accept-expired` option allows `named` to accept expired Signed RRSets (RRSIGs).

Rationale:
Accepting expired RRSIGs may increase the server's exposure to replay attacks.

Remediation:
If present, remove the `dnssec-accept-expired` option from `named.conf`.

Audit:
Verify that `dnssec-accept-expired` is **not** in `named.conf`:

```
$ grep dnssec-accept-expired /etc/named.conf  ; test \
$? -eq 0 && echo "dnssec is accepting expired RRSIG's" \
|| echo "dnssec-accept-expired is correctly disabled"

dnssec-accept-expired is correctly disabled
$
```

If dnssec is accepting expired RRSIG's, you must change your configuration.

Default Value:
The default of `dnssec-accept-expired` is `no`

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.

3.8 Ignore erroneous or unwanted traffic (Level 1, Not Scorable)

Description:
BIND can be configured to ignore requests originating from specified network segments. This is accomplished by implementing the `blackhole` option in `named.conf`. It is recommended that this feature be implemented to ignore requests that originate outside of expected network segments.

Rationale:
By ignoring traffic that originates from unexpected locations, the server's exposure to malicious entities is reduced.

Remediation:
Add a `blackhole` option for multicast and link local addresses, and all private RFC 1918 addresses that are not being used.

```
blackhole {
// Private RFC 1918 addresses
10/8; 192.168/16; 172.16/12;
// Multicast
```

```
224/8;
// Link Local
169.254/16;
};
```

Audit:
Attempt to query the server from an address that has been placed in the blackhole list. If properly configured, the query will fail.

```
nslookup www.google.com ns1.example.com
```

Default Value:
The default of `blackhole` is none.

References:

1. "Secure BIND Template v6.5 12 NOV 2008 Rob Thomas noc@cymru.com," http://www.cymru.com/Documents/secure-bind-template.html.
2. RFC 1918
3. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
4. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).

4. Administration

This section provides guidance on the secure administration of BIND.

4.1 Ensure revision current (Level 1, Scorable)

Description:
Over time, patches will be released to resolve defects in BIND. It is recommended that such patches be applied.

Rationale:
By ensuring that BIND remains current and patched, the probability of an attacker successfully compromising BIND is reduced.

Remediation:
Update BIND to the most current revision. Institute a patch process that aims to apply security updates within 30 days of their release. Subscribe to bind-announce@lists.isc.org on the http://www.isc.org web site to receive notifications of available BIND updates.

Audit:
Verify that the latest patch for your version of BIND is installed.

```
$ /usr/sbin/named –v
BIND 9.5.0
```

Default Value:
Not Applicable

References:

1. "BIND | Internet Systems Consortium," https://www.isc.org/downloadables/11.

4.2 Remove Nameserver ID (Level 1, Scorable)

Description:
RFC 5001 suggested a new `EDNS0` option that is able to identify a DNS server with a Nameserver ID tag. NSID is a method to identify servers in an environment where there are multiple DNS servers sharing the same IP address. With the use of load balancing and other IP sharing mechanisms, it can become difficult to discern exactly which name server is responding to a particular query. NSID allows a name server to respond with identifying information.

The ability to respond to NSID queries was added in BIND 9.5 to help ease the identification of DNS servers. The payload of the NSID can vary according to the wants and needs of the server's administrator. It can contain any string the administrator likes. It is recommended that NSID support be left off.

Rationale:
Enabling this can allow external parties to obtain information about the configuration and architecture of the DNS server. If it is found to be necessary to enable this service, then the identifying information should be generic. You should not use the server's geographic location, IP address or any other privileged information

Remediation:
Use the following in `named.conf` to explicitly disable NSID support:

```
server-id none;
```

Audit:
Check `named.conf` to verify that server id is disabled:

```
$ grep server-id /etc/named.conf  ; test $? -eq 0 && echo "server-id is
set to the above" || echo "server-id was not found and must be set"

      server-id none;
server-id is set to the above
```

If the `server-id` option is not set, or is set to something other than "`none`," use the above steps to set it.

Default Value:
NSID is disabled by default

References:

1. RFC 5001
2. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.

4.3 Logging and Monitoring

This section provides details on logging and monitoring the BIND server.

4.3.1 Configure a syslog channel (Level 1, Scorable)

Description:
Configuring a syslog allows BIND to log any information the administrator sees as important to the health and security of BIND.

Rationale:
Logging is key to monitoring the health and security of the name server and for detecting potential abuse and malicious attacks. Most requests that are invalid or violate an ACL will be logged, so it is important for such logs to come to the attention of the appropriate system administer. Logging is also helpful for debugging your BIND configuration. It is important that you know what kind of logs are going to what channel.

Remediation:
The name server should be configured with at least two channels, a syslog channel to receive a majority of the messages, and a local file to log with duplicates of logs that may be of interest for security and possibly a second log file to be used for debugging. It is possible to configure more channels for more specific types of information. Configure the default and general categories to log to the local syslog.

```
logging {
      channel local_syslog {
      // Specifies the syslog facility to use, check your syslog.conf
      // Some prefer usage of a local<N> facility specific to BIND.
            syslog daemon;
      // debug messages cannot be sent to syslog, info is the lowest.
            severity info;
      };
// Default in BIND 9 includes everything except general
      category default { local_syslog; };
      category general { local_syslog; };
};
```

Audit:
Restart BIND and check the syslog to verify that data is being recorded.

Default Value:
There is no syslog channel by default.

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
2. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).
3. "HP-UX IP Address and Client Management Administrator's Guide" , "5991-6548.pdf (application/pdf Object)," http://docs.hp.com/en/5991-6548/5991-6548.pdf.

4.3.2 Configure a File Channel (Level 1, Scorable)

Description:
To capture logs to a local file, setup a channel for the file. You may want to consider one log file for security related logs, and a second one with a dynamic severity level to be used as needed for debugging.

Rationale:
Logging security related events allows you to see what is affecting the server and adjust the server to prevent attacks.

Remediation:
In `named.conf`, configure a channel for a local security log file with the categories `config`, `dnssec`, `network`, `security`, `updates`, `xfer-in` and `xfer-out`. The local log file will be within the chroot directory.

```
logging {
. . .
      channel local_security_log {
            file "/var/run/named/secure.log" versions 10 size 20m;
            severity debug;
            print-time yes;
      };
      // Config file processing
      category config { local_security_log; };
      // Processing signed responses
      category dnssec { local_security_log; };
      // Network Operations
      category network { local_security_log; };
      // Approved or unapproved requests
      category security { local_security_log; };
      // dynamic updates
      category update { local_security_log; };
      // transfers to the name server
      category xfer-in { local_security_log; };
      // transfers from the name server
```

```
        category xfer-out { local_security_log; };
        // Optional debug log file, may be enabled dynamically.
        channel local_debug_log {
              file "/var/run/named/debug.log";
              severity dynamic;
              print-time yes;
        };
        category default { local_debug_log; };
        category general { local_debug_log; };
};
```

Audit:
Restart BIND and check the appropriate log to verify that data is being recorded. The following command can be used to check this. Ensure that `named` is outputting to the proper log file and the dates match the start time of `named`.

```
# grep named  /var/run/named/secure.log | more
```

Default Value:

There is no security log by default.

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
2. Cricket Liu and Paul Albitz, DNS and BIND, 5th ed. (O'Reilly Media, 2006).
3. "HP-UX IP Address and Client Management Administrator's Guide" , "5991-6548.pdf (application/pdf Object)," http://docs.hp.com/en/5991-6548/5991-6548.pdf.

4.3.3 Disable the HTTP Statistics Server (Level 1, Scorable)

Description:
In the update to BIND 9.5.0 there was a new statistics server included, that is a useful debugging tool in a non-production environment. The HTTP server provides data in XML format about the condition of a BIND 9 server. The statistics server provides the same statistics that are available to the statistics-file dump. This server should be left disabled.

Rationale:
The statistics server should NOT be enabled to prevent potential vulnerabilities.

Remediation:
Leave the statistics server disabled

Audit:
Verify that there is NOT a statistics channel statement:

```
```

```
$ grep statistics-channel /etc/named.conf  ; test $? -eq 0 && \
echo "The statistics channel is currently enabled" \
|| echo "The statistics channel is disabled"

The statistics channel is disabled
$
```

If the statistics channel is found, remove the configuration from `named.conf` and restart BIND.

Default Value:

The HTTP server is disabled by default

References:

1. "Internet Systems Consortium, Inc.," http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php.
2. "Configuring and accessing Bind 9.5.0 statistics," http://netlinxinc.com/index.php?option=com_content&view=article&catid=25:bind&id=47:configuring-and-accessing-bind-950-statistics.

4.4 Defend against Denial of Service Attacks (Level 1, Not Scorable)

Description:
DNS servers have been prime targets in the past for DoS attacks. Although the effect is not as immediate as DoS attacks against a Web server, DNS servers are often easier to attack. In addition, a DoS attack against a DNS server can have a wider affect by denying effective usage of a wide range of services that depend on DNS.

Rationale:
Attacks on DNS servers can cause a variety of problems, including crashing the server, exhausting resources on the server, and flooding the network with bogus traffic

Remediation:
Although there are no 100% solutions for DoS attacks, the usual risk mitigations also apply.

1. Security hardening of the server, including DoS mitigation configurations recommended in the appropriate Unix or Linux CIS benchmark

2. Install redundant distributed DNS servers externally and internally.

3. Add filtering controls on the firewalls and routers in front of the DNS servers to eliminate many forms of unwanted traffic.

Audit:
Review your vendor specific documentation for proper configuration of the above procedures.

Default Value:
Not Applicable

4.5 Do not define a static source port (Level 1, Scorable)

Description:
BIND can be configured to reuse the same source port when communicating with other DNS servers. This capability is made possible through the `query-source` option. It is recommended that this option **not** be used.

Rationale:
Enabling the `query-source` option will increase the probability of an attacker successfully poisoning the DNS cache.

Remediation:
Ensure the `query-source` option in not present in `named.conf`.

Audit:
Verify that `query-source` is not used in `named.conf`:

```
$ grep query-source named.conf ; test $? -eq 0 && \
echo "query-source is set. This is dangerous and should be removed." \
|| echo "query-source is not set."
query-source is not set
```

Default Value:
By default, `query-source` is disabled and BIND will randomize UDP source ports.

References:

1. "BIND Security Matrix | Internet Systems Consortium," https://www.isc.org/node/356.

Summary and the Future of DNS

Historically, DNS continues to be a problematic service with regard to security in that it is unauthenticated and easily spoofed. By its globally distributed nature, it is a protocol that cannot adjust rapidly to changes requiring extensive coordination. Progress is being made, however. Recently, top-level domains have started using DNSSEC to help secure their zones. The history of vulnerability and attacks has brought DNS security to the forefront and recent flaws and a working exploit have helped raise awareness of the need for patching quickly. However, many administrators still struggle to understand the necessity of protecting DNS and awareness is still much lower than it should be. Development of BIND is constantly changing, adding new features and fixing problems. With the BIND 9 being under constant development for the last decade, ISC has begun designing BIND 10, focusing on making the server more flexible to the evolving needs of networks, and continuing to keep an eye on security.

Appendix A: References

1. Internet Systems Consortium (2009). *BIND DOWNLOADS*. Available: http://www.isc.org/downloadables/11. Last accessed 23 April 2009.

2. National Institute of Standards and Technology (2006). *Secure Domain Name System (DNS) Deployment Guide.* Available: http://csrc.nist.gov/publications/nistpubs/800-81/SP800-81.pdf. Last accessed 23 April 2009.

3. Cricket Liu and Paul Albitz (2006). DNS and BIND 5th ed. USA: O'Reilly Media.

4. BIND9.net (2009). *DNS, BIND Nameserver, DHCP, LDAP and Directory Services.* Available: http://www.bind9.net. Last accessed 23 April 2009.

5. Men & Mice (2009). *DNS, DHCP, IPAM, IPv6, DNS Monitoring, IP Address Management, DNSSEC, OSS.* Available: http://www.menandmice.com/knowledgehub/. Last accessed 23 April 2009.

6. Hewlett-Packard (2006). *HP-UX IP Address and Client Management Administrator's Guide.* Available: http://docs.hp.com/en/5991-6548/5991-6548.pdf. Last accessed 23 April 2009.

7. Whitehats.ca (2000). *Jeff Holland DNS/BIND Security*. Available: http://www.whitehats.ca/main/members/Jeff/jeff_dns_security/jeff_dns_security.html . Last accessed 23 April 2009.

8. Brian Wellington et al (2002). *Secure dynamic DNS howto*. Available: http://ops.ietf.org/dns/dynupd/secure-ddns-howto.html. Last accessed 23 April 2009.

9. DNSSEC.net (2009). *DNSSEC: DNS Security Extensions Securing the Domain Name System.* Available: http://www.dnssec.net/. Last accessed 23 April 2009.

10. Red Hat, Inc. (2009). *Taking advantage of SELinux in Red Hat Enterprise Linux.* Available: http://www.redhat.com/magazine/006apr05/features/selinux/. Last accessed: 23 April 2009.

11. Netlinx, Inc. (2009). *Configuring and accessing Bind 9.5.0 statistics.* Available: http://netlinxinc.com/index.php?option=com_content&view=article&catid=25:bind&id=47:configuring-and-accessing-bind-950-statistics. Last accessed 23 April 2009.

12. Steve Friedl (2009). *Building and configuring BIND 9 in a chroot jail.* Available: http://unixwiz.net/techtips/bind9-chroot.html#mkjail. Last accessed 23 April 2009.

13. ZyTrax, Inc. (2009). *DNS for Rocket Scientists.* Available: http://www.zytrax.com/books/dns/. Last accessed 23 April 2009.

14. Sun Microsystems (2005). *Limiting Service Privileges in the Solaris™ 10 Operating System.* Available: https://www.sun.com/blueprints/0505/819-2680.pdf. Last accessed 23 April 2009.

15. Internet Systems Consortium (2008). *BIND 9 Administrator Reference Manual.* Available: http://oldwww.isc.org/index.pl?/sw/bind/arm95/index.php. Last accessed 23 April 2009.

16. Internet Systems Consortium (2009). *BIND Security Matrix.* Available: https://www.isc.org/node/356. Last accessed 23 April 2009.

17. Internet Engineering Task Force (1997). *Dynamic Updates in the Domain Name System (DNS UPDATE).* Available: http://www.ietf.org/rfc/rfc2136.txt. Last accessed 23 April 2009.

18. Internet Engineering Task Force (1997). *Secure Domain Name System Dynamic Update.* Available: http://www.ietf.org/rfc/rfc2137.txt. Last accessed 23 April 2009.

19. Internet Engineering Task Force (2000). *Secret Key Transaction Authentication for DNS (TSIG).* Available: http://www.ietf.org/rfc/rfc2845.txt. Last accessed 23 April 2009.

20. Internet Engineering Task Force (2005). *DNS Security Introduction and Requirements*. Available: http://www.ietf.org/rfc/rfc4033.txt. Last accessed 23 April 2009.

21. Internet Engineering Task Force (2005). *Resource Records for the DNS Security Extensions.* Available: http://www.ietf.org/rfc/rfc4034.txt. Last accessed 23 April 2009.

22. Internet Engineering Task Force (2005). *Protocol Modifications for the DNS Security Extensions.* Available: http://www.ietf.org/rfc/rfc4035.txt. Last accessed 23 April 2009.

23. Internet Engineering Task Force (2006). *A DNS Resource Record (RR) for Encoding Dynamic Host Configuration Protocol (DHCP) Information (DHCID RR).* Available: http://www.ietf.org/rfc/rfc4701.txt. Last accessed 23 April 2009.

24. Internet Engineering Task Force (2006). *Resolution of Fully Qualified Domain Name (FQDN) Conflicts among Dynamic Host Configuration Protocol (DHCP) Clients.* Available: http://www.ietf.org/rfc/rfc4703.txt. Last accessed 23 April 2009.

25. Internet Engineering Task Force (2007). *DNS Name Server Identifier (NSID) Option.* Available: http://www.ietf.org/rfc/rfc4703.txt. Last accessed 23 April 2009.

Appendix B: Change History

Date	Version	Changes for this version
May 4th, 2009	2.0.0	Updated all previous information with updated security practices. Included several new technologies included in updates up to BIND 9.5.0-P2, including DHCID, NSID, and the HTTP statistics server.
January, 2006	1.0	Public Release