全面提升 BIND DNS 服务器安全

作者: 曹江华

此 文 是 it168 网 站 的 - 篇 约 稿 , 希 望 网 友 指 正 。 原 文 链 接 : $\underline{http://publish.it168.com/2005/1228/20051228077401.shtml}$

一、DNS 服务器的重要性

DNS 是因特网建设的基础, 几乎所有的网络应用, 都必须依赖 DNS 系统做网址查询的 指引动作。如果 DNS 系统运作不正常, 即使 Web 服务器都完好如初, 防火墙系统都善尽 其职,相关的后端应用服务器以及数据库系统运作正常,因为无法在期限时间内查得到网址, 将会导致电子邮件无法传递, 想要使用网域名称去连接某个网页, 也会因查不出网络地址, 以致联机失败。2001年1月24日,美国微软公司所管理的相关网络系统,遭受网络黑客的 拒绝服务攻击后导致全球各地的用户 接近 24 小时的时间 无法连上该公司相关的网站, 造成该公司相当严重的商业损失。根据以往的经验之中, 网络攻击的对象 多数主要集中在 控制网络路由的设备(路由器, 防火墙等)和各类应用服务器(Web、邮件 等)。因此,目前多数 的网络系统安全保护,通常都集中在路由设备和应用服务器本身。然而,这一次的微软公司 被攻击事件, 与以往其它网站攻击事件的最大不同, 就在于这一次被攻击的对象是 DNS 服 务器 而不是 WEB 服务器本身。这次的事件宣告另一种新型的网络攻击类别,往后将可能 成为常态。 就这个观点而言, 如何能加强确保 DNS 系统的运作正常, 或者当 DNS 系统在 遭受网络攻击时候, 能够让管理者及早发现, 然后加以排除, 诸如此类, 正是日益重要的系 统安全的重要课题。互联网上 DNS 服务器的事实标准就是 ISC 的 BIND, Netcraft 在 DNS 服务器上的统计(http://www.netcraft.com/)显示 2003 年第二季度进行的一个调查发现,在互 联网上运行着的 DNS 服务器中, ISC 的 bind 占据了 95%的市场份额。互联网是由很多不可 见的基础构件组成。这其中就包含了 DNS, 它给用户提供了易于记忆的机器名称(比如 yahoo.com),并且将它们翻译成数字地址的形式。对于那些用于公共服务的机器一般还提供 "反向查询"的功能,这种功能可以把数字转换成名字。由于历史的原因,这种功能使用的 是被隐藏的 "in-addr.arpa"域。对 in-addr 域的调查,可以让我们更加了解整个 Internet 是如 何运作的。Bill Manning 对 in-addr 域的调查发现,有 95%的域名服务器(2 的 2000 次方个服 务器中)使用的是各种版本的"bind"。这其中包括了所有的 DNS 根服务器,而这些根服务 器对整个服务器的正常运转起着至关重要的作用。

二、DNS 服务面临的安全隐患:

DNS 服务面临的安全隐患主要包括: DNS 欺骗 (DNS Spoffing)、拒绝服务 (Denial of service, DoS) 攻击、分布式拒绝服务攻击和缓冲区漏洞溢出攻击 (Buffer Overflow)。

1.DNS 欺骗

DNS 欺骗即域名信息欺骗是最常见的 DNS 安全问题。当一个 DNS 服务器掉入陷阱,使用了来自一个恶意 DNS 服务器的错误信息,那么该 DNS 服务器就被欺骗了。DNS 欺骗会使那些易受攻击的 DNS 服务器产生许多安全问题,例如:将用户引导到错误的互联网站点,或者发送一个电子邮件到一个未经授权的邮件服务器。网络攻击者通常通过两种方法进行DNS 欺骗。

(1) 缓存感染

黑客会熟练的使用 DNS 请求,将数据放入一个没有设防的 DNS 服务器的缓存当中。这些缓存信息会在客户进行 DNS 访问时返回给客户,从而将客户引导到入侵者所设置的运行木马的 Web 服务器或邮件服务器上,然后黑客从这些服务器上获取用户信息。

(2) DNS 信息劫持

入侵者通过监听客户端和 DNS 服务器的对话,通过猜测服务器响应给客户端的 DNS 查询 ID。每个 DNS 报文包括一个相关联的 16 位 ID 号,DNS 服务器根据这个 ID 号获取请求源位置。黑客在 DNS 服务器之前将虚假的响应交给用户,从而欺骗客户端去访问恶意的网站。

(3) DNS 重定向

攻击者能够将 DNS 名称查询重定向到恶意 DNS 服务器。这样攻击者可以获得 DNS 服务器的写权限。

2.拒绝服务攻击

黑客主要利用一些 DNS 软件的漏洞,如在 BIND 9 版本(版本 9.2.0 以前的 9 系列)如果有人向运行 BIND 的设备发送特定的 DNS 数据包请求,BIND 就会自动关闭。攻击者只能使 BIND 关闭,而无法在服务器上执行任意命令。如果得不到 DNS 服务,那么就会产生一场灾难:由于网址不能解析为 IP 地址,用户将无方访问互联网。这样,DNS 产生的问题就好像是互联网本身所产生的问题,这将导致大量的混乱。

3、分布式拒绝服务攻击

DDOS 攻击通过使用攻击者控制的几十台或几百台计算机攻击一台主机,使得服务拒绝攻击更难以防范:使服务拒绝攻击更难以通过阻塞单一攻击源主机的数据流,来防范服务拒绝攻击。Syn Flood 是针对 DNS 服务器最常见的分布式拒绝服务攻击。SYN Flood 攻击利用的是 IPv4 中 TCP 协议的三次握手(Three-Way Handshake)过程进行的攻击。大家知道协议规定,如果一端想向另一端发起 TCP 连接,它需要首先发送 TCP SYN 包到对方,对方收到后发送一个 TCP SYN+ACK 包回来,发起方再发送 TCP ACK 包回去,这样三次握手就结束了。我们把 TCP 连接的发起方叫作"TCP 客户机(TCP Client)",TCP 连接的接收方叫作"TCP 服务器(TCP Server)"。值得注意的是在 TCP 服务器收到 TCP SYN request 包时,在发送 TCP SYN+ACK 包回 TCP 客户机前,TCP 服务器要先分配好一个数据区专门服务于这个即将形成的 TCP 连接。一般把收到 SYN 包而还未收到 ACK 包时的连接状态成为半开连接(Half-open Connection)。在最常见的 SYN Flood 攻击中,攻击者在短时间内发送大量的 TCP SYN 包给受害者,这时攻击者是 TCP 客户机,受害者是 TCP 服务器。根据上面的描述,受害者会为每个 TCP SYN 包分配一个特定的数据区,只要这些 SYN 包具有不同的源地址(这一点对于攻击者来说是很容易伪造的)。这将给 TCP 服务器系统造成很大的系统负担,最终导致系统不能正常工作。

4.缓冲区漏洞溢出攻击

黑客利用 DNS 服务器软件存在漏洞,比如对特定的输入没有进行严格检查,那么有可能被攻击者利用,攻击者构造特殊的畸形数据包来对 DNS 服务器进行缓冲区溢出攻击。如果这一攻击成功,就会造成 DNS 服务停止,或者攻击者能够在 DNS 服务器上执行其设定的任意代码。例如,前一阵子针对 Linux 平台的 BIND 的攻击(e.g. Lion worm)程序,就是利用某些版本的 BIND 漏洞,取得 root 权限,一旦入侵完成之后,入侵者就可以完全控制整个相关的网络系统,影响非常严重。主要包括:

- (1) 更改 MX 记录,造成邮件被截获、修改或删除。
- (2) 更改 A 记录,使您的 WWW 服务器的域名指向黑客的具有同样 WWW 内容的主机,诱使访问者登录,获取访问者的密码等相关信息。添加 A 记录,使黑客的主机拥有被相信的域名,以此来入侵通过启用域名信任机制的系统。
- (3) 利用这台主机作为攻击其他机器的"跳板"。

应对以上这些安全隐患方法有两个最有效的原则:

1.选择安全没有缺陷的 DNS 版本:

BIND 主要分为三个版本:

- 1、v4, 1998 年多数 UNIX 捆绑的是 BIND4, 已经被多数厂商抛弃了, 除了 OpenBSD 还在 使
- 用。OpenBSD 核心人为 BIND8 过于复杂和不安全, 所以继续使用 BIND4。这样一来 BIND8/9 的很多优点都不包括在 v4 中。
- 2、v8,就是如今使用最多最广的版本,其详细内容可以参阅"BIND 8+ 域名服务器安全增强" http://security.nsfocus.com/showQueryL.asp?libID=530>
- 3、v9,最新版本的 BIND,全部重新写过,免费(但是由商业公司资助),BIND9 在 2000 年十月份推出,根据调查 v9 版本的 BIND 是最安全的,它的最新安全版本在其官方网站: http://www.isc.org/ 下载下载源代码安装即可。例如使用 Linux 系统针对拒绝服务攻击只要将 BIND 9 升级为版本 9.2.1 即可。

2.保持 DNS 服务器配置正确、可靠

这一点相对困难。Linux 上的域名服务由 named 守护进程控制,该进程从主文件: /etc/named.conf 中获取信息。它包括一组将主机名称映射为 IP 地址的各种文件。Linux 下主要 DNS 配置文件见表一:

文件名称	中文名称	作用
/etc/hosts	主机的一个列表文件	包含(本地网络中)已知主机的一个列表。如
		果系统的 IP 不是动态生成,就可以使用它。
		对于简单的主机名解析(点分表示法),在请
		求 DNS 或 NIS 网络名称服务器之前,
		/etc/hosts.conf 通常会告诉解析程序先查
		看这里。
/etc/host.conf	转化程序控制文件	告诉网络域名服务器如何查找主机名。(通常
		是 /etc/hosts,然后就是名称服务器; 可通
		过 netconf 对其进行更改)
/etc/resolv.conf	转化程序配置文件	在配置程序请求 BIND 域名查询服务查询主机
		名称时必须告诉程序使用那个域名服务器和
		IP 地址来完成这个任务。

表-1 DNS 主要配置文件

可以看到 DNS 配置文件是一个复杂的系统。伴随 DNS 建立出现的许多问题都会引起相同的结果,但起因却不同。但大多数问题是由于配置文件中的语法错误而导致的。 DNS 是一组文件构成的,所以需要不同工具检查对应文件。一个配置存在缺陷的 DNS 服务器会存在很大的安全漏洞。这里可以通过一些工具: nslookup、dig、 named-checkzone、 host、 named-checkconf、dlint 等对 DNS 配置文件进行检查。其中安装 BIND 软件包时自动安装的工具包括: nslookup、dig、 named-checkzone、 host、 named-checkconf。 dlint 是一个专门检查 DNS 配置文件开放源代码软件。可以在 http://www.domtools.com/dns/dlint.shtml 下载。

三、详细 DNS 服务器工作状态

在维护 DNS 服务器时, 网管员希望知道到底是哪些用户在使用 DNS 服务器, 同时也希望能

对 DNS 状态查询做一个统计。这里我们可以使用 dnstop 查询 DNS 服务器状态: 软件下载和安装:

#wegt http://www.stearns.org/dnstop/dnstop-20040309-1.i386.rpm

#rpm - ivh dnstop-20040309-1.i386.rpm

运行软件:

如果想查看通过 eth0 的 DNS 网络流量可以使用命令

#dnstop -s eth0

0 new queries, 6 total queries Mon Jul 6 09:54:35 2004

Sources count %
-----192.168.0.117 4 66.7
192.168.0.143 1 16.7
192.168.0.42 1 16.7

在运行 dnstop 的过程中,可以敲入如下命令: s, d, t, 1, 2, 3, ctrl+r, ctrl+x, 以交互方式来显示不同的信息。

- 1-(TLD)记录查询的顶级域名,
- 2-(SLD)记录查询的二级域名,
- 3-(SLD)记录查询的三级域名,
- s-(Source)记录发送 dns 查询的客户端 ip 地址表,
- d一(Destinations)记录 dns 查询的目的服务器的 ip 地址表,
- t-(Query Type)记录查询详细类型。

ctrl+r-重新纪录。

ctrl+x一退出。更详细信息可以查看 man dnstop。

二、配置一个安全 BIND 服务器

下面以使用最为广泛的 Unix 类 DNS 软件 BIND 为例,介绍如何配置一个安全 DNS 服务器。本文以 RHEL 4.0 为工作平台。

1. 隔离 DNS 服务器

首先应该隔离 BIND 服务器,不应该在 DNS 服务器上跑其他服务,尤其是允许普通用户登陆。减少其它的服务可以缩小被攻击的可能性,比如混合攻击。

2. 隐藏 BIND 的版本号

通常软件的 BUG 信息是和特定版本相关的,因此版本号是黑客寻求最有价值的信息。黑客使用 dig 命令可以查询 BIND 的版本号,然后黑客就知道这个软件有那些漏洞。因此随意公开版本号是不明智的。隐藏 BIND 版本号比较简单,修改配置文件:/etc/named.conf,在 option部分添加 version声明将 BIND 的版本号信息覆盖。例如使用下面的配置声明,当有人请求版本信息时,迫使 nmaed 显示: "Unsupported on this platform"

options {

version "unnow on this platform";

};

3. 避免透露服务器信息

和版本号一样,也不要轻易透露服务器其他信息。为了让潜在的黑客更难得手,建议不要在 DNS 配置文件中使用这 HINFO 和 TXT 两个资源记录。

4. 关闭 DNS 服务器的 glue fetching 选项

当 DNS 服务器返回一个域的域名服务器纪录并且域名服务器纪录中没有 A 纪录,DNS 服务器会尝试获取一个纪录。就称为 glue fetching,攻击者可以利用它进行 DNS 欺骗。关闭 glue fetching 是一个好方法,修改配置文件:/etc/named.conf.加入一行:

options no-fetch-glue

5. 使用非 root 权限运行 BIND

在 Linux 内核 2.3.99 以后的版本中,可以以一u 选项以非 root 权限运行 BIND。命令如下: /usr/local/sbin/named -u nobody

上面的命令表示以 nobody 用户身份运行 BIND。使用 nobody 身份运行能够降低缓冲区溢出 攻击所带来的危险。

6. 控制区域(zone)传输

默认情况下 BIND 的区域(zone)传输是全部开放的,如果没有限制那么 DNS 服务器允许对任何人都进行区域传输的话,那么网络架构中的主机名、主机 IP 列表、路由器名和路由 IP 列表,甚至包括各主机所在的位置和硬件配置等情况都很容易被入侵者得到。因此,要对区域传输进行必要的限制。可以通过在 / etc / named. conf 文件当中添加以下语句来限制区域传输:

```
acl "zero-transfer" { 192.168.0.52; 192.168.0.109; };
zone "yourdomain.com" {
type master;
file "yourdomain.com";
allow-rtransfer { zero-transfer; }; };
这样只有 IP 地址为: 从 192.168.0.52 到 192.168.0.109 的主机能够同 DNS 服务器进行区域传输。
```

7. 请求限制

如果任何人都可以对 DNS 服务器发出请求,那么这是不能接受的。限制 DNS 服务器的服务范围很重要,可以把许多入侵者据之门外。修改 BIND 的配置文件:/etc/named.conf 加入以下内容:

```
options {
  allow-query { 168.192.1.0/24; };
};

zone "yourdomain.com" {
  type master;
  file "yourdomain.com";
  allow-query { any; };
};

zone "1.168.192.in-addr.arpa" {
  type master;
  file "db.192.168.1";
  allow-query { any; };};
```

这样所有的用户都可以访问 yourdomain.com 的 DNS 服务器,但是只有 168.192.1.0 网段的 主机用户可以请求 DNS 服务器的任意服务。

另外也不要允许其他网段的主机进行递归询问,在上面文件最后加入一行即可: allow-recursion { 192.168.1.0/24; };

8. 其他強化措施:

(1) 使用存取控制清单 (Access Control Lists), 主要目的在于产生地址配对清单。

语法: acl "name" { address_match_list }; address match list: 地址匹对清单。

例子: acl "mis" {192.168.200.15,192.168.143.192/26};

(2) 使用 Forwarders 代询服务器机制,它将自己无法解析的查询转送到某特定的服务器。

语法: forwarders ip_address_liest

例子: 以下是建议的 forwarders 设定(在/etc/named.conf 中) orwarders { // 指定提供查询的上层 DNS。 www.twnic.net.tw; // 到上层 (twnic) 的 DNS 查询。

需注意的是通常我们指定的是到本身上一层 dns,但也可能因 dns 缓存有误而转送到错误的服务器上。

(3) 使用 allow-transfer: 目的在于只允许授权的网域主机能更新、读取 DNS 辖区内的记录。

语法: allow-transfer { <address_match_list> };

例子: address match list: 允许进行 DNS 辖区数据传输主机的 IP 列表。

(4) allow-update: 目的在于指定能向本 dns 服务器提交动态 dns 更新的主机

语法: allow-update { <address_match_list> };

例子: address_match_list: 允许能向本 DNS 服务器提交动态 DNS 更新的主机 IP 列表

9. 使用 DNSSEC

DNS 欺骗 spoofing 对目前网络应用,最大的冲击在于 冒名者借着提供假的网域名称与网址的对照信息,可以将不知情用户的网页联机,导引到错误的网站,原本属于用户的电子邮件也可能因而遗失,甚而进一步空开成为阻断服务的攻击。所幸,目前较新的 BIND 版本,针对这一类问题,已经有加入许多改进的方法,不过真正的解决方案,则有赖封包认证机制的建立与推动。DNSSEC 就是试图解决这一类问题的全新机制,BIND9 已经完整加以设计并完成。DNSSEC 引入两个全新的资源记录类型: KEY 和 SIG,允许客户端和域名服务器对任何 DNS 数据的来源进行密码验证。

DNSSEC 主要依靠公钥技术对于包含在 DNS 中的信息创建密码签名。密码签名通过计算出一个密码 hash 数来提供 DNS 中数据的完整性,并将该 hash 数封装进行保护。私/公钥对中的私钥用来封装 hash 数,然后可以用公钥把 hash 数译出来。如果这个译出的 hash 值匹配接收者刚刚计算出来的 hash 树,那么表明数据是完整的。不管译出来的 hash 数和计算出来的 hash 数是否匹配,对于密码签名这种认证方式都是绝对正确的,因为公钥仅仅用于解密合法的 hash 数,所以只有拥有私钥的拥有者可以加密这些信息。

10. 为 DNS 服务器配置 DNS Flood Detector

DNS Flood Detector 是针对 DNS 服务器的 Syn Flood 攻击检测工具,用于侦测恶意的使用 DNS 查询功能. DNS Flood Detector 会监控像服务器查询名称解析的数量,分成守护进程(daemon) 和后台(bindsnap) 模式。守护进程模式:会通过 syslog 发出警示(/var/log/messages),bind 捕捉模式:可以得到几近实时的查询状态。下载安装:

wegt http://checksuite.sourceforge.net/dl/dnsflood-1.08-1.i386.rpm rpm —ivh dnsflood-1.08-1.i386.rpm

命令格式:

dns_flood_detector [选项]

主要选项:

- -i IFNAME 监听某一特定接口。
- -t N 当每秒查询数量超过 N 值时发出警示.
- -a N 经过 N 秒后重置警示。
- -w N 每隔 N 秒显示状态。
- -x N 创建 N个 buckets。
- -m N 每隔 N 秒显示所有状态。
- -b 以后台模式执行(bindsnap)。
- -d 以守护进程模式执行(daemon)。
- -v 显示较多的输出信息。
- -h 显示使用方式。

应用实例:

dns flood detector -b -v -v - t3

messages 的纪录: 以守护进程模式执行,纪录每秒超过 3 次查询的纪录,显示最多信息,包含 A PTR MX 纪录等。

11.建立完整的域名服务器

Linux 下的 DNS 服务器用来存储主机的域名信息。包括三种:

(1) 惟高速存域名服务器(Cache-only server)

惟高速存域名服务器(Cache-only server)不包含域名数据库。它从某个远程服务器取得每次域名服务器查询的回答,一旦取得一个回答,就将它放入高速缓存中。下次查询相同的信息就用它予以回答。

(2) 主域名服务器(Primary Name server)

主域名服务器是特定域所有信息的权威来源。它从域管理员构造的本地文件中加载域信息,该"区文件"包含着服务器具有管理权的部分域结构的最精确的信息。主域名服务器需要配置包括一组完整的文件:主配置文件(named.conf)、正向域的区文件(named.hosts)、反向域的区文件(named.rev)、高速缓存初始化文件(named.ca)和回送文件(named.local)。

(3) 辅助域名服务器 (Second Name Server)

辅助域名服务器用来从主服务器中转移一整套域信息。辅助域名服务器是可选的配置选项。区文件是从主服务器转移出的,作为磁盘文件保存在辅助域名服务器中。辅助域名服务器不需配置本地区文件。只要配置主配置文件、高速缓存初始化文件、回送文件。

说明:多数域名服务器要根据网络实际情况将以上三种服务器组合,进行合理配置。所有的域名服务器都需要一个设置惟高速缓存服务器提供名字解析。一个域只能建立一个主域名服务器,另外至少要创建一个辅助域名服务器作为主域名服务器的备份。一个域的主服务器可以是其他域的辅助域名服务器。 为了提高域名系统的可靠性,应建立辅域名服务器。 当主域名服务器不能正常工作的情况下,能够替代主域名服务器对外提供不间断的服务,并起到数据备份的作用。

12.增强 DNS 服务器的防范 Dos/DDoS 功能

1.使用 SYN cookie

SYN Cookie 是对 TCP 服务器端的三次握手协议作一些修改,专门用来防范 SYN Flood 攻击的一种手段。它的原理是,在 TCP 服务器收到 TCP SYN 包并返回 TCP SYN+ACK 包时,不分配一个专门的数据区,而是根据这个 SYN 包计算出一个 cookie 值。在收到 TCP ACK 包时,TCP 服务器在根据那个 cookie 值检查这个 TCP ACK 包的合法性。如果合法,再分配专门的数据区进行处理未来的 TCP 连接。在 linux 下以 root 权限执行:

#echo 1 > /proc/sys/net/ipv4/tcp_syncookies

2. 增大 backlog

通过增加 backlog 的数值,可以一定程度减缓大量 SYN 请求导致 TCP 连接阻塞的状况,一般这个数值系统默认是 1024,可以增加到 1280 至 2048:

echo "2048" > /proc/sys/net/ipv4/tcp_max_syn_backlog

这样在强度不是很高的攻击下,系统响应能力提高了一点。

3.缩短 retries 次数

Linux 系统默认的 tcp_synack_retries 是 5 次,将这个数值减少可以提高系统响应能力,为 2 次:

echo "2" > /proc/sys/net/ipv4/tcp_synack_retries

修改后, SYN RECV 的数量有了少量减少,系统响应也快了一些。

4.限制 SYN 频率

上述的几个方法 DDOS 攻击效果不太好,目前比较有效的是对 SYN 的频率和次数进行限制,这样最大限度的控制了单个 IP 地址发动攻击的能力。例如将 SYN 请求的次数限制在 30 次每分钟,系统默认是 5 次/秒,显然太高,同时将 burst 从默认的 5 个降低到 2 个。

/sbin/iptables -A INPUT -p tcp --tcp-flags SYN,ACK,FIN,RST SYN -m \

limit --limit 30/m --limit-burst 2 -j ACCEPT

进行此操作后,对正常的用户而言无任何感觉上的差异,而并发的 SYN 请求量下降了不少,服务响应基本正常了。

13. 使用分布式 DNS 负载均衡

在众多的负载均衡架构中,基于 DNS 解析的负载均衡本身就拥有对 DDOS(SYN Flood)的免疫力,基于 DNS 解析的负载均衡能将用户的请求分配到不同 IP 的服务器主机上,攻击者攻击的永远只是其中一台服务器,一来这样增加了攻击者的成本,二来过多的 DNS 请求可以帮助我们追查攻击者的真正踪迹(DNS 请求不同于 SYN 攻击,是需要返回数据的,所以很难进行 IP 伪装)。但是基于 DNS 解析的负载均衡成本很高。很多中小公司没有部署这个技术。

14.防范对于 DNS 服务器网络

嗅探器技术被广泛应用于网络维护和管理方面,它工作的时候就像一部被动声纳,默默的接收看来自网络的各种信息,通过对这些数据的分析,网络管理员可以深入了解网络当前的运行状况,以便找出网络中的漏洞。在网络安全日益被注意的今天.我们不但要正确使用嗅探

器.还要合理防范嗅探器的危害.嗅探器能够造成很大的安全危害,主要是因为它们不容易被发现。对于一个安全性能要求很严格的企业,同时使用安全的拓扑结构、会话加密、使用静态的 ARP 地址是有必要的。

15.及时更新系统补丁

你应该经常到你所安装的系统发行商的主页上去找最新的补丁。目前操作系统维护主要分两种模式:对于私有操作系统(Windows/Solaris等)由于个人用户不能直接接触其源代码,其代码由公司内部开发人员维护,其安全性由同样的团队保证,内核的修正与其他应用程序一样,以 patch/SP 包的方式发布。对于 Linux 这样的开放式系统,是一种开放的结构。应该说,开放的模式是双刃剑。从机制上讲,全世界的开发人员都能获得源代码,从而找出其中的纰漏,似乎安全性应该更好;但是同时,如果网络管理人员不能及时更新内核,也会留下安全隐患。如果你是一个 Linux 网管员,你经常需要上相应的网站看,是否有补丁,是否有了 bug fix,是否需要升级。Linux 服务器运行的软件主要包括: Bind, Apache 等软件大都是开源软件,而且都在不停升级,稳定版和测试版交替出现。在 www.apache.org 上,最新的 changeLog 中都写着: bug fix, security bug fix 的字样。所以 Linux 网管员要经常的关注相关网站及时升级或添加补丁。

总结: DNS 是网络服务的基础建设,要长期不断地保持其正常运作,每一个 DNS 服务器都应该定期检测。域名系统的配置和管理是一项比较复杂和繁琐的系统管理任务,它对整个网络的运行影响极大。为了保证 DNS 服务器的安全运行,不仅要使用可靠的服务器软件版本,而且要对 DNS 服务器进行安全配置,同时还要跟踪服务器软件和操作系统的各种漏洞,