TechTarget 中国

Linux 服务器安全技巧及工具总结

Linux 服务器安全技巧及工具总结

随着开源系统 Linux 的盛行，其在大中型企业的应用也在逐渐普及，很多企业的应用服务都是构筑在其之上，例如 Web 服务、数据库服务、集群服务等等。因此，Linux 的安全性就成为了企业构筑安全应用的一个基础，是重中之重，如何对其进行安全防护是企业需要解决的一个基础性问题。基于此，本技术手册介绍了保护 Linux 服务器安全的技巧和工具。

工具

Linux 最大的特点莫过于功能强大，性能稳定的服务器。本节将介绍保护 Linux 服务器安全的工具：chroot jail、TCP wrapper、chroot……

- ❖使用 chroot jail 或 TCP wrapper 确保 Linux 服务器安全
- ❖加强 Linux 服务器安全的 Choot 使用
- ❖如何用命令行将 LINUX 服务器安全地接入 WIFI 网络？
- ❖quid 访问控制：ACL 元素以及访问列表
- ❖使用 Squid 配置反向代理服务器

技巧

Linux 应用广泛，维护这样一个企业级的安全的计算环境需要设计策略和过程，本节将讨论 Linux 服务器的安全风险评估和防护要点。

- ❖最佳实践：如何确保 Linux 服务器安全？
- ❖企业级 Linux 系统下的进程安全管理方法
- ❖构建企业级 Linux 服务器安全的十大要点（上）
- ❖构建企业级 Linux 服务器安全的十大要点（中）
- ❖构建企业级 Linux 服务器安全的十大要点（下）

使用 chroot jail 或 TCP wrapper 确保 Linux 服务器安全

系统管理员的任务是确保一个或者多个系统方便用户使用。在 Linux 系统中，管理员和用户都可以是你，你和电脑的距离也不过几十厘米而已。系统管理员可能在半个地球远的地方支持网络系统，而你只是成千上万用户中的一个。系统管理员可能是利用业余时间维护系统的兼职人员，他同时也可能是这个系统的用户。管理员也可能由几个人组成，他们都全职负责维持多个系统正常运行。

确保服务器安全

你可以通过使用 TCP wrapper，或者通过建立一个 chroot jail 来确保服务器的安全。

TCP Wrappers 客户服务器安全（hosts.allow 和 hosts.deny）

当你打开一个本地系统去访问远程系统时，你必须确保满足以下条件：

- 只对你希望允许访问的系统开发本地系统。
- 允许每个远程系统只访问你允许访问的数据。
- 允许每个远程系统只能以适当的方式（只读/读写/只写）去访问数据。

TCP wrapper 作为客户服务器模型的一部分，依赖/etc/hosts.allow 和/etc/hosts.deny 文件作为简单访问控制语言的基础，可用于任何包含了 libwrap.so 的 daemon 程序使用。访问控制语言限定的规则是：基于客户端地址和客户端试图访问的 daemon 程序，选择性地允许客户端访问服务器在本地系统上的 daemon 程序。

hosts.allow 和 hosts.deny 文件中的每行代码都遵循以下格式：
daemon_list client_list [command]

其中，daemon_list 是一个或多个服务器 daemon 程序（如 rpcbind、vsftpd、或 sshd）的逗号分隔列表。client_list 是一个或者多个客户端的逗号分隔列表。命令是可选的，当 client_list 的客户端试图访问从 daemon_list 访问服务器 daemon 程序时指代被执行的命令。

当客户端请求连接一个本地服务器时，hosts.allow 和 hosts.deny 文件按以下方式进行查询，直到找到匹配的为止。

- 如果 daemon 客户端对匹配 hosts.allow 中一行，可以允许访问。
- 如果 daemon 客户端对匹配 hosts.deny 中一行，访问将被拒绝。
- 如果在 hosts.allow 文件或 hosts.deny 文件中都没有匹配，可以允许访问。

第一个匹配决定客户端是否允许访问服务器。当 hosts.allow 和 hosts.deny 都不存在时，就意味着这个文件是空的。虽然不建议这样做，但通过删除这两个文件，你可以允许所有客户端访问所有 daemon 程序。

一个更安全系统的例子是，在 hosts.deny 写入以下语句，即可阻止所有访问。

```
$ cat etchosts.deny
...
ALL:ALL:echo '%c tried to connect to %d and was blocked'>> /var/log/tcpwrappers.log
...
```

这行语句阻止了所有试图连接到服务器上的客户端，而在 hosts.allow 中特别允许的客户端除外。当这条规则满足匹配时，它在名为/var/log/tcpwrappers.log 的文件中增加了一行。%c 是扩展到客户端的信息，%d 是扩展到客户端试图连接到 daemon 程序的名称。

在 hosts.deny 文件已存在的基础上，你可以在 hosts.allow 写入代码，明确地允许访问指定的服务器和系统。例如，以下 hosts.allow 文件允许任何人连接 OpenSSH 的 daemon 程序（ssh、scp、sftp），但只允许从本地系统与 192.168.子网在同一网络的远程连接。

```
$ cat etchosts.allow
sshd   ALL
in.telnet   LOCAL
in.telnet   192.168. 127.0.0.1
...
```

第一行允许从任何系统（ALL）连接到 sshd。第二行允许与服务器（LOCAL）有相同域名的系统连接。第三行表示匹配所有 IP 地址以 192.168.开头的系统，包括本地系统。

建立一个 chroot jail

在早期的 UNIX 系统中，根目录在文件系统中是固定的。在包括 Linux 的现代 UNIX 中，你可以为每个进程都定义一个根目录。而 chroot 命令则可以允许你运行/目录以外的进程。

根目录出现在目录层次结构的顶部，没有父目录。一个进程不能访问根目录上的任何文件（因为它们不存在）。如果你运行一个程序（进程），指定它的根目录为/home/sam/jail，程序将不会意识到在/home/sam 或者更上层目录还有文件的存在：jail 是程序的根目录，且被标记为/（而不是 jail）。

通过人工建立一个根目录，通常称为（chroot）jail，你可以从根本上阻止程序访问或者修改（可能是恶意的）文件以外的目录。为了增加安全性，你必须建立一个适当的 chroot jail。如果你没有正确设置 chroot jail，还不如没有 chrootjail，因为你可能让恶意用户更容易成功地访问系统。

(来源：TechTarget 中国　作者：Mark G. Sobell, Prentice Hall　译者：Dan)

加强 Linux 服务器安全的 Choot 使用

使用 chroot

创建一个 chroot jail 很简单：用 root 身份登录，输入命令/usr/sbin/chroot directory（directory 为相应的目录）。该 directory 目录就变成了根目录，并且该程序试图运行默认的 shell。利用/home/sam 目录中的根权限，在/home/sam/jail 目录下创建 chroot jail 的命令如下：

#/usr/sbin/chroot/home/sam/jail

/usr/sbin/chroot: 不能运行命令'/bin/bash'，无此文件或目录。

该示例创建了一个 chroot jail，但是当它试图运行 bash shell 时却失败了。Jail 一旦创建，jail 目录就会取代根目录的名字，/，所以 chroot 找不到/bin/bash 路径指定的文件。此种情况下，chroot jail 虽然工作，但是没有任何用处。

让 chroot jail 按照你的意愿工作有点麻烦。为了在上述例子中的 chroot jail 里面运行 bash ,你需要在 jail 中（/home/sam/jail/bin）创建一个 bin 目录，并把/bin/bash 拷贝到这个目录下。由于 bash 二进制文件动态链接某些共享库，你还需要把这些库复制到 jail 中。这些库应该放在 lib 目录下。

下面的例子中创建了必要的目录、拷贝了 bash、使用 ldd 显示 bash 所需要的共享库，并把必要的库复制到 lib 目录下。

linux-gate.so.1 文件是由内核提供的动态共享对象（DSO），它能加速系统的调用；你不必把它复制到 lib 目录下。

$ pwd

/home/sam/jail

$ mkdir bin lib

```
$ cp binbash bin

$ ldd binbash

linux-gate.so.1 = (0x0089c000)

libtinfo.so.5 = liblibtinfo.so.5 (0x00cdb000)

libdl.so.2 = liblibdl.so.2 (0x00b1b000)

libc.so.6 = liblibc.so.6 (0x009cb000)

libld-linux.so.2 (0x009ae000)

$ cp /lib/{libtinfo.so.5,libdl.so.2,libc.so.6,ld-linux.so.2} lib
```

现在什么都设置好了，你可以重新启动 chroot jail。尽管普通用户可以进行所有设置，但是你必须用超级用户运行 chroot。

```
$ su
Password:
# /usr/sbin/chroot .
bash-3.2# pwd
/
bash-3.2# ls
bash ls command not found
bash-3.2#
```

这一次 chroot 找到并开启了 bash，显示出它的默认提示符（bash- 3.2#）。Pwd 命令可以正常运行，因为它是一个 shell 内置命令。然而，bash 无法找到 ls 功能（它不在 chroot jail 中）。如果你想让用户使用 ls 命令，你可以把/bin/ls 文件夹以及它的库拷贝到 jail 目录下。

为了创建一个有用的 chroot jail，首先请确定 chroot jail 用户需要哪些功能。然后把相应的二进制文件和它们的库拷贝到 jail 中。或者，你可以创建二进制文件的静态拷贝，并把它们放在 jail 中，不必安装单独的库。

（静态链接二进制文件要比他们的动态形式大得多。带有 bash 和核心功能的基础系统大小超过了 50 兆字节）。你可以在 bash 和 coreutils SRPMS 包中找到大多数常用功能的源代码。

不管你选择哪种技术，你都必须把 su 拷贝到 jail 中。非 root 用户需要用 su 命令来运行程序。因为 root 用户可以从 chroot jail 中跳出来，所以在 chroot jail 中运行程序必须使用非 root 用户。

Fedora RHEL 发布的动态 su 版本需要 PAM，并且不会在 jail 中运行。如果想在 jail 中使用 su，你就要从源代码中创建一个 su 的拷贝。默认情况下，你创建的任何一个 su 拷贝都不需要 PAM。

为了使用 su 命令 ,你必须把/etc/passwd 和 /etc/shadow 文件中的相关内容都拷贝到 jail 里的 etc 目录下相应的文件中。

提示

同时使用多个 chroot jail

如果你打算使用多个 chroot jail，那么请在某个地方保持一个干净的 bin 和 libfile 文件拷贝，而不是在激活的 jail 中进行拷贝。

在 chroot jail 中运行服务

在 jail 中运行 shell 的用处有限。实际上，你更有可能在 jail 中运行一项特定的服务。为了在 jail 中运行服务，你必须保证该服务需要的所有文件都必须在 jail 中。在 chroot jail 中开始一项服务的命令格式如下：

```
# /usr/sbin/chroot  jailpath  /bin/su  user daemonname  &
```

其中 jailpath 是 jail 目录的路径名称，user 是运行 daemon（后台程序）的用户名，daemonname 是提供该项服务的 daemon 的路径（jail 内部）。

有些服务器已经能够利用 chroot jail 功能。比如，你可以设置 DNS，以便指定 jail 中运行的程序，而且 vsftpd FTP 服务器能够自动为客户端开启 chroot jail。

安全注意事项

有些服务需要用 root 用户运行，但是它们在开始之后会释放它们的 root 权限（Procmail 和 vsftpd 就是例子）。如果你运行这种服务，你就不必把 su 命令放在 jail 中。

以 root 用户身份运行的进程可能会从 chroot jail 中跑出去。因此，你应该在 jail 中开始运行程序之前总是对另外一个用户进行 su 命令。同时，请注意 jail 中存在哪些 setuid 二进制文件，它们中的任何安全漏洞都会破坏到 jail 的安全。此外，请确保用户不能访问他上传的可执行文件。

(来源：TechTarget 中国　作者：Mark G. Sobell, Prentice Hall　译者：Dan)

如何用命令行将 LINUX 服务器安全地接入 WIFI 网络？

现在大多数 LINUX 都提供有图形化的工具帮助你配置无线网络连接。但是如果你想把一台服务器接入某无线网络，大部分服务器（没有系统内置）就没有类似的图形工具选项。在这篇文章中，你可以学到如何用命令行工具将服务器接入到无线网络。

配置一个无线连接，包含了若干步骤。首先，你需要检查当天连接。假设你的网络已经知道无线端口是 wlan0，那么扫描命令 iwlist wlan0 就很好用。这个命令可能给你两个不同的输出结果。它将提供无线网卡的配置信息或者列出一个完整的可用网络清单。如果你看不到后者，则需要用命令 ifconfig wlan0 up 确认 wlan0 已经启用。这样你就应该能看见一个可用网络清单。

用命令 ‘iwlist wlan0’ 扫描将提供一个可用网络及属性的清单：

root@texas:/etc/wpa_supplicant# iwlist wlan0 scanning
wlan0 Scan completed :
Cell 01 - Address: 00:1D:7E:0F:D3:38 Channel:3 Frequency:2.422 GHz (Channel 3) Quality=70/70 Signal level=-39 dBm Encryption key:on ESSID:"kippis" Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s 9 Mb/s; 12 Mb/s; 18 Mb/s Bit Rates:24 Mb/s; 36 Mb/s; 48 Mb/s; 54
Mb/s Mode:Master Extra:tsf=00000895bfd8b177 Extra: Last beacon: 170ms ago IE: Unknown: 00066B6970706973 IE: Unknown: 010882848B960C121824 IE: Unknown: 030103 IE: IEEE 802.11i/WPA2 Version 1 Group Cipher : TKIP Pairwise Ciphers (1) : CCMP Authentication Suites (1) : PSK IE: WPA Version 1 Group Cipher : TKIP Pairwise Ciphers (1) : TKIP Authentication Suites (1) : PSK IE: Unknown: 0406000200000000 IE: Unknown: DD180050F2020101800003A4000027A4000042435E0062322F00 IE: Unknown: 2D1A6E1803FFFF000000000000000000000000000000000000 IE: Unknown:

3D160305000000000000000000000000000000000000000000 IE: Unknown:
DD1E00904C336E1C03FFFF000000000000000000000000000000000000000000000000 IE:
Unknown: DD1A00904C340305000000000000000000000000000000000000000000 IE:
Unknown: 2A0103 IE: Unknown: 32043048606C Cell 02 - Address:
00:18:39:2C:5D:94 Channel:11 Frequency:2.462 GHz (Channel
11) Quality=25/70 Signal level=-85 dBm Encryption
key:on ESSID:"ilestvivant" Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11
Mb/s Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s 36
Mb/s; 48 Mb/s; 54
Mb/s Mode:Master Extra:tsf=000001981a060177 Extra:
Last beacon: 5150ms ago IE: Unknown:
000B696C657374766976616E74 IE: Unknown: 010482848B96 IE:
Unknown: 03010B IE: WPA Version 1 Group Cipher :
TKIP Pairwise Ciphers (1) : TKIP Authentication Suites (1) :
PSK IE: Unknown: 0406010200000000 IE: Unknown:
050400010000 IE: Unknown: 2A0100 IE: Unknown:
32080C1218243048606C Cell 03 - Address:
00:26:F2:5F:6A:36 Channel:6 Frequency:2.437 GHz (Channel
6) Quality=20/70 Signal level=-90 dBm Encryption
key:on ESSID:"valkrustlaan9" Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11
Mb/s; 18 Mb/s 24 Mb/s; 36 Mb/s; 54 Mb/s Bit Rates:6 Mb/s; 9
Mb/s; 12 Mb/s; 48
Mb/s Mode:Master Extra:tsf=000007b3896f1192 Extra:
Last beacon: 5740ms ago IE: Unknown:
000D76616C6B727573746C61616E39 IE: Unknown:
010882840B162430486C IE: Unknown: 030106 IE: Unknown:
050400030000 IE: Unknown: 2A0100 IE: Unknown:
2F0100 IE: IEEE 802.11i/WPA2 Version 1 Group Cipher :
TKIP Pairwise Ciphers (2) : CCMP TKIP Authentication Suites (1) :
PSK IE: Unknown: 32040C121860 IE: Unknown:
2D1A7C181BFFFF000000000000000000000000000000000000 IE: Unknown:
3D160608000000000000000000000000000000000000000000 IE: Unknown:

DD0E0050F204104A0001101044000102 IE: Unknown: DD090010180200F0050000 IE: WPA Version 1 Group Cipher : TKIP Pairwise Ciphers (2) : CCMP TKIP Authentication Suites (1) : PSK IE: Unknown: DD180050F2020101800003A4000027A4000042435E0062322F00 IE: Unknown: DD1E00904C337C181BFFFF0000000000000000000000000000000000000000 IE: Unknown: DD1A00904C340608000000000000000000000000000000000000 Cell 04 - Address: 00:0C:F6:3B:83:C0 Channel:11 Frequency:2.462 GHz (Channel 11) Quality=18/70 Signal level=-92 dBm Encryption key:on ESSID:"Sitecom" Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s 9 Mb/s; 12 Mb/s; 18 Mb/s Bit Rates:24 Mb/s; 36 Mb/s; 48 Mb/s; 54 Mb/s Mode:Master Extra:tsf=0000003836a9619b Extra: Last beacon: 5130ms ago IE: Unknown: 000753697465636F6D IE: Unknown: 010882840B160C121824 IE: Unknown: 03010B IE: Unknown: 050402030000 IE: Unknown: 2A0104 IE: Unknown: 32043048606C IE: WPA Version 1 Group Cipher : TKIP Pairwise Ciphers (1) : TKIP Authentication Suites (1) : PSK IE: Unknown: DD0700E04C01020300

有的时候，命令 iwlist wlan0 扫描出来的结果特别长，你需要在清单中搜索可用网络的 ESSID’ s。而某些地方，你可能需要寻找你想加入的网络，包括该网络所使用的加密方式信息。大多数时候会采用 WPA 加密，所以现在你只需要确认连接密码就可以了。要实现这个，你需要创建一个文 wpa_supplicant.conf。典型地，你可以在目录/etc/wpa_supplicant 下找到它。这里，你需要定义一个你想使用的授权机制名（大多数时候是 PSK），就是该网络的 SSID，和需要连接的网络的 WPA 密码。

在配置文件 wpa_supplicant.conf 中定义 WPA 参数：

root@texas:/etc/wpa_supplicant# cat wpa_supplicant.conf
network={ ssid="mynetwork" key_mgmt=WPA-PSK psk="69ABC9DF20" }
After creating this file, you can run a first test to see if you can connect to the network that is specified using the wpa_supplicant command. To do this, use the following command:
wpa_supplicant -i wlan0 -c /etc/wpa_supplicant/wpa_supplicant.conf

这个时候，你应该能看到你的服务器和无线网络在协商并且最终成功连接。一旦这确实起作用，你可以用命令 wpa_supplicant 加上参数 - B 在后台将它作为 daemon 启用。在这之后，你只需要配置接口的 IP 信息就可以了。如果网络中有可用的 DHCP 服务器，你还可以用命令 dhclient wlan0 来自动获取一个 IP 地址。这将向 DHCP 服务器申请一个 IP 地址，等拿到地址后，你就可以连入网络了。

在某些情况下，知道如何用命令行接入无线网络非常实用。要完成这些，wpa_supplicant 是关键命令。在确认启用无线网络之后，可以用一个包含所有目标连接无线参数的配置文件来运行这个命令。

(来源：TechTarget 中国　作者：Tom Nolle　译者：曾少宁)

Squid 访问控制：ACL 元素以及访问列表

代理服务器是介于浏览器和 Web 服务器之间的另一台服务器。有了该服务器之后 ,浏览器不是直接到 Web 服务器去取回网页而是向代理服务器发出请求，信息会先送到代理服务器，由代理服务器来取回浏览器所需要的信息并传送给客户的浏览器。

现代企业应用代理服务器，除了提高访问速度外，同时，它在实际的应用过程中又通常被企业作为“安全网关” ,可以根据企业设定的代理规则来过滤和屏蔽一些用户的非法请求和信息 ,从而达到保护企业网的目的。在企业开源系统的代理服务中，可以通过设置安全访问控制规则、配置带认证的代理服务以及反向代理服务来确保企业网络安全，本文将详细对这些防护手段进行介绍。

开源代理服务器 Squid 简介

Squid 可以工作在很多的操作系统中 ,如 AIX、Digital、UNIX、FreeBSD、HP-UX、Irix、Linux、NetBSD、Nextstep、SCO、Solaris、OS/2 等。对于 Web 用户来说，Squid 是一个高性能的代理缓存服务器，和一般的代理缓存软件不同、Squid 用一个单独的、非模块化的、I/O 驱动的进程来处理所有的客户端请求。Squid 由一个主要的服务程序 Squid，一个 DNS 查询程序 DNS server，几个重写请求和执行认证的程序，以及几个管理工具组成。当 Squid 启动以后，它可以派生出预先指定数目的 DNS server 进程，而每一个 DNS server 进程都可以执行单独的 DNS 查询，这样就大大减少了服务器等待 DNS 查询的时间。

用户可以从 Red Hat Enterprise Linux 发行套件中获取该软件的 RPM 包进行安装并使用 #/etc/rc.d/init.d/squid start 或者使用#service squid start 命令进行服务开启。

使用安全访问控制限制企业用户上网行为

使用访问控制特性，可以控制在访问时根据特定的时间间隔进行缓存、访问特定站点或一组站点等等。Squid 访问控制有两个要素：ACL 元素和访问列表。访问列表可以允许或拒绝某些用户对此服务的访问。下面分别介绍 ACL 元素以及访问列表的使用方法。

ACL 元素

该元素定义的语法如下：

acl aclname acltype string1...
acl aclname acltype “file” ...

当使用文件时，该文件的格式为每行包含一个条目。

其中，acltype 可以是 src、dst、srcdomain、dstdomain、url_regex、urlpath_regex、time、port、proto、method 中的一任意一种。

src：指明源地址。可以用以下的方法指定：
acl aclname src ip-address/netmask ... 客户 ip 地址
acl aclname src addr1-addr2/netmask ... 地址范围

dst：指明目标地址，即客户请求的服务器的 IP 地址。语法为：
acl aclname dst ip-address/netmask ...

srcdomain：指明客户所属的域，Squid 将根据客户 IP 反向查询 DNS。语法为：
acl aclname srcdomain foo.com ...

dstdomain：指明请求服务器所属的域，由客户请求的 URL 决定。语法为：
acl aclname dstdomain foo.com ...

time：指明访问时间。语法如下：
acl aclname time [day-abbrevs] [h1:m1-h2:m2][hh:mm-hh:mm]

日期的缩写指代关系如下：

S：指代 Sunday
M：指代 Monday
T：指代 Tuesday
W：指代 Wednesday
H：指代 Thursday
F：指代 Friday
A：指代 Saturday

另外，h1：m1 必须小于 h2：m2，表达式为[hh：mm-hh：mm]。

port：指定访问端口。可以指定多个端口，比如：
acl aclname port 80 70 21 ...
acl aclname port 0-1024 ... 指定一个端口范围

proto：指定使用协议。可以指定多个协议：
acl aclname proto HTTP FTP ...

method：指定请求方法。比如：
acl aclname method GET POST ...

url_regex：URL 规则表达式匹配，语法为：
acl aclname url_regex[-i] pattern

urlpath_regex：URL-path 规则表达式匹配，略去协议和主机名。其语法为：
acl aclname urlpath_regex[-i] pattern

在使用上述 ACL 元素的过程中，要注意如下几点：

acltype 可以是任一个在 ACL 中定义的名称。
任何两个 ACL 元素不能用相同的名字。
每个 ACL 由列表值组成。当进行匹配检测的时候，多个值由逻辑或运算连接；换句话说，任一 ACL 元素的值被匹配，则这个 ACL 元素即被匹配。
并不是所有的 ACL 元素都能使用访问列表中的全部类型。
不同的 ACL 元素写在不同行中，Squid 将这些元素组合在一个列表中。

http_access 访问控制列表

根据访问控制列表允许或禁止某一类用户访问。如果某个访问没有相符合的项目，则默认为应用最后一条项目的“非”。比如最后一条为允许，则默认就是禁止。通常应该把最后的条目设为“deny all”或“allow all”来避免安全性隐患。

使用该访问控制列表要注意如下问题：

这些规则按照它们的排列顺序进行匹配检测，一旦检测到匹配的规则，匹配检测就立即结束。

访问列表可以由多条规则组成。

如果没有任何规则与访问请求匹配，默认动作将与列表中最后一条规则对应。

一个访问条目中的所有元素将用逻辑与运算连接（如下所示）：

http_access Action 声明 1 AND 声明 2 AND

多个 http_access 声明间用或运算连接，但每个访问条目的元素间用与运算连接。

列表中的规则总是遵循由上而下的顺序。

(来源：TechTarget 中国　作者：羽扇纶巾)

使用 Squid 配置反向代理服务器

配置反向代理服务器确保企业网络服务安全

反向代理（Reverse Proxy）方式是指以代理服务器来接受 Internet 上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给 Internet 上请求连接的客户端，此时代理服务器对外就表现为一个服务器。

值得注意的是：通常的代理服务器，只用于代理内部网络对 Internet 的连接请求，客户机必须指定代理服务器,并将本来要直接发送到 Web 服务器上的 http 请求发送到代理服务器中。由于外部网络上的主机并不会配置并使用这个代理服务器，普通代理服务器也被设计为在 Internet 上搜寻多个不确定的服务器，而不是针对 Internet 上多个客户机的请求访问某一个固定的服务器，因此普通的 Web 代理服务器不支持外部对内部网络的访问请求。

当一个代理服务器能够代理外部网络上的主机 ,访问内部网络时 ,这种代理服务的方式称为反向代理服务。此时代理服务器对外就表现为一个 Web 服务器 ,外部网络就可以简单把它当作一个标准的 Web 服务器而不需要特定的配置。不同之处在于，这个服务器没有保存任何网页的真实数据，所有的静态网页或者 CGI 程序，都保存在内部的 Web 服务器上。因此对反向代理服务器的攻击并不会使得网页信息遭到破坏 ,这样就增强了 Web 服务器的安全性。

反向代理方式和包过滤方式或普通代理方式并无冲突，因此可以在防火墙设备中同时使用这两种方式，其中反向代理用于外部网络访问内部网络时使用，正向代理或包过滤方式用于拒绝其他外部访问方式并提供内部网络对外部网络的访问能力。因此可以结合这些方式提供最佳的安全访问方式。

目前有许多反向代理软件，比较有名的有 Nginx 和 Squid 。其他还包括 Socks、Apache、Jigsaw、Delegate 等。

使用 Squid 配置反向代理服务器

Squid 作为反向代理服务器使用时，其工作原理为：客户端请求访问 Web 服务时，DNS 将访问的域名解析为 Squid 反向代理服务器的 IP 地址 ,这样客户端的 URL 请求将被发送到反向代理服务器。如果 Squid

反向代理服务器中缓存了该请求的资源，则将该请求的资源直接返回给客户端，否则反向代理服务器将向后台的 Web 服务器请求资源，然后将请求的应答返回给客户端，同时也将该应答缓存在本地，供下一个请求者使用。

Squid 反向代理一般只缓存可缓冲的数据(比如 HTML 网页和图片等) ,而一些 CGI 脚本程序或者 ASP、JSP 之类的动态程序默认不缓存。它根据从 Web 服务器返回的 HTTP 头标记来缓冲静态页面。有四个最重要 HTTP 头标记：

Last-Modified：告诉反向代理页面什么时间被修改
Expires：告诉反向代理页面什么时间应该从缓冲区中删除
Cache-Control：告诉反向代理页面是否应该被缓冲
Pragma：用来包含实现特定的指令，最常用的是 Pragma:no-cache

要配置反向代理服务器，需要在 squid 的主配置文件里面添加如下内容：

```
http_port 80 accel vhost vport
cache_peer 192.172.1.133 parent 80 0 no-query originserver
cache_peer_domain www.test.com 192.172.1.133
acl sites dstdomain www.test.com
http_access allow sites
http_access deny all

cache_dir ufs /var/spool/squid3 100 16 256
cache_mgr yourmail@somesite.com
cache_mem 64 MB
maximum_object_size_in_memory 1028 KB
access_log /var/log/squid3/access.log squid
```

上述配置的详细解释如下：

http_port 80 accel vhost vpor：指定 Squid 所服务的端口为 80，vhost 和 vport 指的是所采用的虚拟主机的方式：基于 IP 地址和基于端口的；
cache_peer 192.172.1.133 parent 80 0 no-query originserver：指定真实 Web Server 的 IP 地址；
cache_peer_domain www.test.com 192.172.1.133：告诉反向代理服务器，当客户端有对

www.test.com 的访问请求时，需要从真实 Web Server 192.172.1.133 上取得数据；

acl sites dstdomain www.test.com：定义客户端能够通过反向代理服务器访问的主机；

http_access allow sites、http_access deny all：限制客户端通过反向代理服务器能够访问的范围；

cache_dir ufs /var/spool/squid3 100 16 256、cache_mgr yourmail@somesite.com、cache_mem 64 MB、maximum_object_size_in_memory 1028 KB、access_log /var/log/squid3/access.log squid：代理服务器的常规配置。

(来源：TechTarget 中国　作者：羽扇纶巾)

最佳实践：如何确保 Linux 服务器安全？

维护一个企业级的安全的计算环境需要设计策略和过程从而使得对系统和数据的未授权访问降至最低。为了保护基于 Linux 的计算机资产免于这些威胁，像许多其它以安全为核心的过程一样，你必须知道你想保护什么以及别人可能会如何尝试获取访问。成功的安全管理是心态。也就是说，像坏孩子那样思考。

在本文中，我们将会讨论基于 Linux 的服务器系统的风险评估。

确保你的 Linux 服务器系统安全的第一步是正确地评估所面临的风险。只有这之后企业才能部署一套有效的防护措施来预防、侦测，并且如果需要的话对于可能发生的违规正确地做出反应。

首先，辨识需要保护的 Linux 资产。资产可能包括硬件、软件、数据或像 email 或 Web 站点主机这样运转的服务。每个资产都具有价值，要么是货币价值要么是未来可能带来收入。

接着，辨识每个资产面临的潜在威胁。威胁可能来自组织的内部或外部。一些内部威胁只不过是偶然的，但是有些可能是恶意的。

对于资产的威胁依赖于攻击的动机和攻击者如何获得对资产的访问。动机可能是纯粹的挑战，伤害资产的拥有者，或是为了谋利。攻击者可能想访问你的数据或只是拒绝合法用户的访问。每个威胁都有被利用的必然的可能性，这通常与资产的价值相关。虽然在组织内广泛地了解和变化会有困难，但是使用一个风险管理框架来给每个辨识的威胁分配一个可能性，会帮助你对缓和这些风险需要采取的行动进行优先级安排。

尽管不可能列出所有潜在的威胁途径，但一份最常见风险的概述能让你开始自己的风险评估。

最棘手的威胁途径是用户。尽管有各种的保护机制，人们仍然会被愚弄或被要挟做出不恰当的行为。用户的意识、培训和访问权限是缓和任何 Linux 风险的重要部分。

密码在任何计算环境中经常代表着最常见的软肋。为了辨识不安全的登录密码，运行如 John the Ripper 这样的密码有效性检查器。应用和数据库的密码也应该检查“可被破解性”或是进行修改以便满足这样的需求。同时，辨识 Linux 服务器上不需要的访问授权。例如，如果密码文件（/etc/passwd）被远程地分发（通过 rcp/rcopy 程序或 NIS 服务），用户可能会对从未使用的服务器具有登录访问权限，从而创造了毫无好处的潜在的威胁途径。

另外一个主要的威胁途径是网络。任何能访问你的本地网络（物理的或是无线方式）的用户有可能试图连接到网络上任何其它的资产。所有的 Linux 系统运行开放的网络端口，并等待来自网络查询的程序。每个这种服务都代表者一个威胁途径，要么通过欺诈的认证，或是由于软件瑕疵可能错误地允许访问。使用 netstat 命令来找到系统所有的开放端口。

使用 Nmap 工具扫描网络上其它机器的开放端口。每个开放端口代表着一个威胁途径，应该被关闭或是监控非法的访问。不要忽视任何传统的拨号访问点。防火墙是可信网络和不可信网络（如因特网）之间的边界。你的防火墙应该配置只在已知和需要的端口上传输数据。防火墙传输数据的每个端口同样都是一个威胁途径。

除了正常的监控以外，你还应该同时检查日志来关联需要的访问。Lastlog 命令显示用户的登录信息。可以在路径/var/log/messages 下发现各种各样的日志信息。许多应用和数据库也提供记录机制来追溯用户的访问。检查这些日志，你可以观察当前谁在使用和（可能）需要访问特定的资源。

无论什么复杂的软件都是有缺陷的，但是只有当缺陷以不受欢迎的行为表现它们时才会被了解。通常的 bug 只会破坏数据或是引起宕机，但是有一些会造成无法预料的后果，比如允许未授权的访问。这明显地表示为主要的危害。攻击者不断地搜索着这些类型的 bug，而厂商们则在发现这些 bug 时，尽力快速地修补它们和提供软件补丁。你所能做的是确保定期地检查和更新你的操作系统和应用软件。

检查 Linux 服务器上软件更新的过程依赖于应用或是 Linux 版本。例如 Ubuntu 版本的 Linux 提供一个更新管理器（通过菜单“系统>管理>软件源”可以发现），可以配置每天进行检查更新。你越经常性地检查更新，你的漏洞窗口越小。同样对于来自未校验来源或作者的免费程序或软件要小心谨慎。

需要监控和保持更新的最重要的软件是面临外部环境的软件，如 Web 服务器和网络应用（如 VPN 或 SSH）。Web 服务器软件定期地检测糟糕的配置和 bug。Web 应用可能会遇到恶作剧的输入数据来进行不正当的应用。大多数的 Web 应用语言，像 Perl、Python、Ruby 或 PHP 有工具或可用的附件来净化输入数据以及禁止用户输入的代码，如 SQL 或 Java 脚本。你的 Web 服务器或是其它面临外部环境的应用接收来自用户的数据都意味着可能的威胁。同样，检查这些程序产生的任何日志文件有助于你辨识合法和非法的访问。

(来源：TechTarget 中国　作者：King Ables　译者：Odyssey)

企业级 Linux 系统下的进程安全管理方法

在企业级的 Linux 应用中，进程是整个计算机系统的一个主体，它需要通过一定的安全等级来对客体（包括系统中的文件、数据、设备等）发生作用。进程在一定条件下可以对诸如文件、数据库等客体进行操作。如果进程用作其他不法用途，将给系统带来重大危害。在当前形形色色的面向 Linux 系统的攻击中，许多网络黑客都是通过种植“木马”的办法来达到破坏计算机系统和入侵的目的，而这些“木马”程序无一例外的是需要通过进程这一方式在系统中运行才能发挥作用的。

作为服务器中占绝大多数市场份额的 Linux 系统，要切实保证计算机系统的安全，我们必须对其进程进行安全管理。

Linux 进程管理的方法主要包括：（1）确定并综合分析系统中当前运行进程的状态及信息，包括内存、CPU、执行用户身份、进程 ID 等，以确定其是否合法以及状态是否正常；（2）事先限制进程所占用的系统资源，如文件系统资源和派生进程数目等，以合理控制进程的运行状况。下面将对这些手段进行详细介绍。

一、管理手段一：使用基本命令进行进程查看

传统的方法可以通过 Linux 系统的一些基本命令进行 Linux 系统的进程查看和分析。Linux 系统提供了 who、w、ps 和 top 等察看进程信息的系统调用，安全工作者可以通过结合使用这些系统调用，清晰地了解进程的运行状态以及存活情况，从而采取相应的措施，来确保 Linux 系统的安全。

其中，who 命令主要用于查看当前在线上的用户情况。系统管理员可以使用 who 命令监视每个登录的用户此时此刻的所作所为；w 命令也用于显示登录到系统的用户情况，但是与 who 不同的是，w 命令功能更加强大，它不但可以显示有谁登录到系统，还可以显示出这些用户当前正在进行的工作，w 命令是 who 命令的一个增强版；ps 和 top 命令则是最基本同时也是非常强大的进程查看命令。使用这些命令可以动态和静态地确定有哪些进程正在运行和运行的状态、进程是否结束、进程有没有僵死、哪些进程占用了过多的资源等等。

举个例子，黑客在入侵系统后通过植入一些系统本没有的非法进程来留作“后门”，以达到下次使用该系统资源或者利用该系统作为“肉鸡”发动拒绝服务等来攻击其他目标主机的目的，而我们就可以结合上述命令来找出异常进程。

二、管理手段二：使用进程文件系统进行管理

管理手段一中所使用的命令行方式对 Linux 系统中的进程管理比较粗略和不全面 如果要进行全面地管理，可以借助进程文件系统（即 PROC 文件系统）来获取系统中运行进程所占用的内存、CPU、中断、命令行等情况，以辅助安全管理员进行恶意进程的发现和排查。

PROC 文件系统是一个虚拟的文件系统，通过文件系统的接口实现，用于输出系统的运行状态。它以文件系统的形式，为操作系统本身和应用进程之间的通信提供了一个界面，使应用程序能够安全、方便地获得系统当前的运行状况和内核的内部数据信息，并可以修改某些系统的配置信息。另外，由于 PROC 以文件系统的接口实现，因此用户可以像访问普通文件一样对其进行访问，但它只存在于内存之中，并不存在于真正的物理磁盘当中。所以，当系统重启和电源关闭的时候，该系统中的数据和信息将全部消失。

表 1 说明了该文件系统中一些重要的文件和目录。

文件或目录	说　明
/proc/1	关于进程 1 的信息目录。每个进程在/proc 下有一个名为其进程号的目录
/proc/cpuinfo	处理器信息，如类型、制造商、型号和性能
/proc/devices	当前运行的核心配置的设备驱动的列表
/proc/dma	显示当前使用的 DMA 通道
/proc/filesystems	核心配置的文件系统
/proc/interrupts	显示使用的中断
/proc/ioports	当前使用的 I/O 端口
/proc/kcore	系统物理内存映像
/proc/kmsg	核心输出的消息，也被送到 syslog
/proc/ksyms	核心符号表
/proc/loadavg	系统的平均负载
/proc/meminfo	存储器使用信息，包括物理内存和 swap

/proc/modules	当前加载了哪些核心模块
/proc/net	网络协议状态信息
/proc/stat	系统的不同状态
/proc/version	核心版本
/proc/uptime	系统启动的时间长度
/proc/cmdline	命令行参数

表 1　重要的 PROC 文件系统文件和目录

下面举个简单的例子，说明安全管理员如何来全面查看系统中一个运行进程的相关信息。

（1）进程的基本信息都会存放在/proc 文件系统中，具体位置是在/proc 目录下。通过使用如下命令可以查看系统中运行进程的相关信息，如图 1 所示，其中显示为系统中运行进程的信息所存放的目录，每个进程对应一个目录，3193 为例子使用的进程的详细信息所在目录：

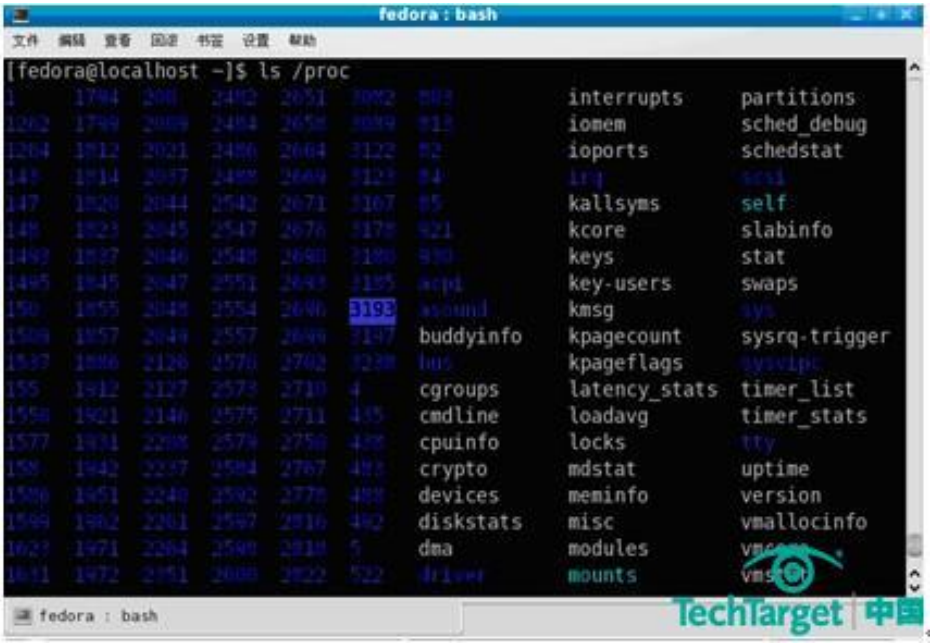

图 1　/ls/proc 命令显示结果

（2）切换到 3193 目录，以方便详细的查看进程信息，并列出进程详细的状态信息文件，如图 2 所示：

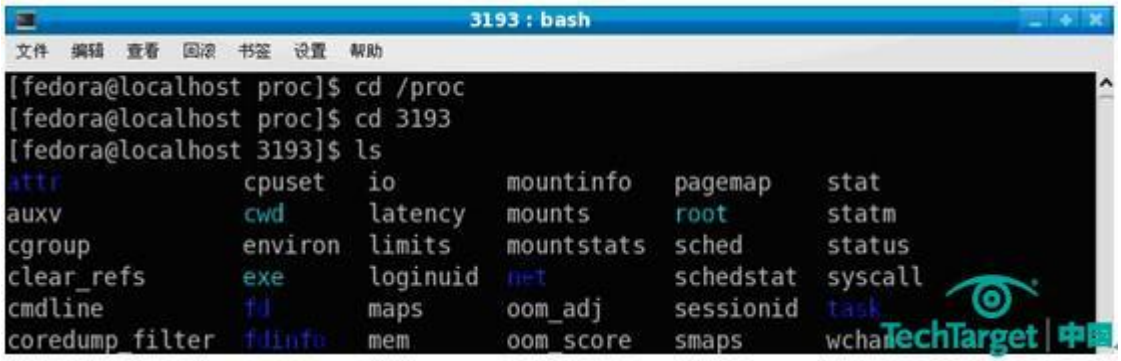

图 2　进程 3193 信息所在目录

（3）在这些文件当中，status 这个状态文件是比较重要的，包含了很多关于进程的有用的信息，用户可以从这个文件获得信息，如下所示：

其中，比较重要的字段详细含义如下：

Name:　scientific_comp　//进程名
State:　R (running)　//进程运行状态
Tgid:　3193　//进程组 ID
Pid:　3193　//进程 ID
PPid:　3123　//父进程 ID
TracerPid:　0　//跟踪调试进程 ID
Uid:　6004　6004　6004　6004　//进程所对应程序的 UID
Gid:　6004　6004　6004　6004 //进程所对应程序的 GID
FDSize: 256　//进程使用文件句柄大小
Groups: 6004　//组信息

这样，安全管理员就可以通过进程名、进程 ID、父进程 ID、UID、GID 等信息来综合判定系统中进程的合法状态，以捕捉非法进程，并进行后续处理。

三、管理手段三：限制进程使用的资源

在系统使用过程中，一些用户编写的进程可能无意识地创建一些大型的文件或者派生(fork)过多地进程，从而过度消耗系统资源，引起系统的不稳定。同时，一些病毒也可能有派生多个进程的行为出现，如臭名昭著的“震荡波”病毒。这些都使得我们有必要来限制进程使用的资源，保证系统安全。

为了防止进程或者其子进程创建大型文件，可以使用 ulimit 命令来进行限制，具体的命令使用 ulimit –f 后接以 K 字节为单位指定的最大文件尺寸。图 3 举出了一个具体的例子加以说明。在该例子中，首先采用 ulimit 命令限制当前 shell 进程可以创建的文件大小；然后，采用 yes 命令不断写入大量的字符串到 test.txt 文件中，该文件大小超过了 ulimit 命令许可的范围，结果系统提示文件超过了大小，并终止了 yes 命令的不断写入过程。从后面使用 ls 命令来查看 test.txt 文件的大小来看，ulimit 命令很好地将该文件大小限制在 20KB 的范围之内。

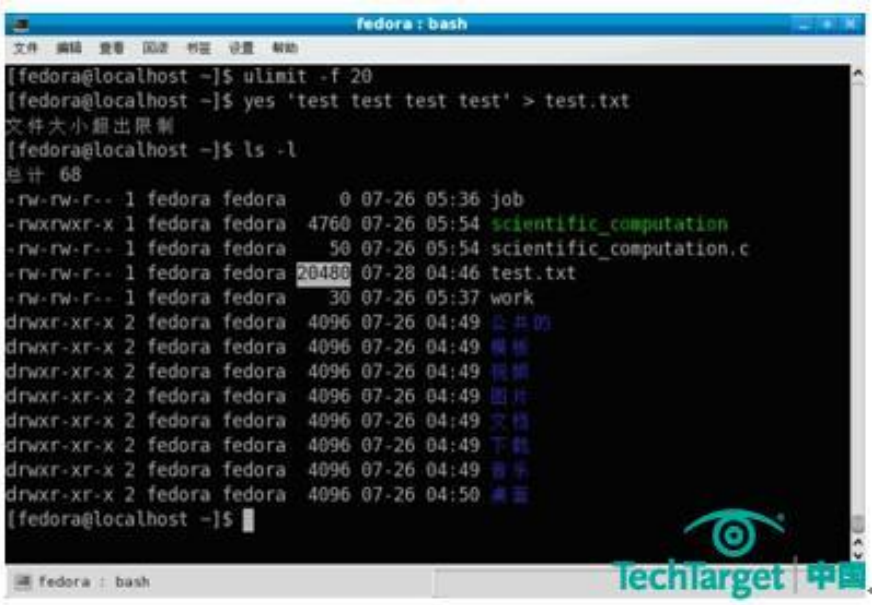

图 3　ulimit 命令使用示意

在实际的使用过程中，用户可以降低自身的限制值，但是不能增加限制值。并且，只有 root 用户才能在 /etc/profile 文件中增加 ulimit 选项的设置。因此，图 4 中所示的增加自身限制值大小的操作是被禁止的；反之，图 5 种所示的 root 用户的操作就是允许的（请读者注意图 4 和图 5 中使用不同的用户进行操作）。

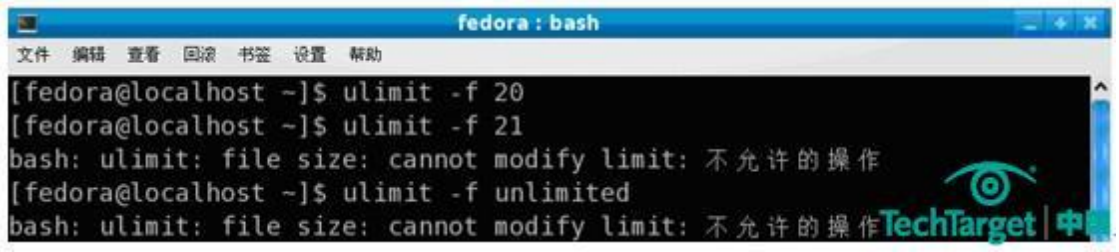

图 4　不允许非 root 用户增加 ulimit 值

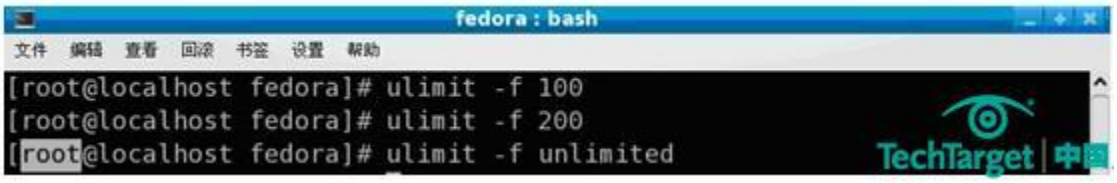

图 5　允许 root 用户增加 ulimit 值

另外，值得注意的是：虽然能够采用 ulimit 值来限制进程创建文件的大小，但是该机制并不能保证用户创建多个相同大小的文件。比如，ulimit 的限制值是 20KB，那么该机制只能限制进程创建的单个文件大小不能超过 20KB，而不能限制进程创建 10 个甚至 100 个 20KB 大小的文件。

Ulimit 命令还可以用来限制单个用户（父进程）所能调用的最大子进程个数，以避免某个父进程由于无所限制的创建子进程而造成整个系统崩溃。

图 6 给出了一个使用 ulimit 命令限制子进程无限调用的例子。首先，使用脚本编辑来自动生成进程；然后，使用 ulimit 命令来限制父进程调用的最大子进程个数为 8。最后，可以看到当创建到第 9 个时，系统报错并阻断子进程的再度调用。

图 6　使用 ulimit 限制单个用户调用的最大进程个数

(来源：TechTarget 中国　作者：羽扇纶巾)

构建企业级 Linux 服务器安全的十大要点（上）

随着开源系统 Linux 的盛行，其在大中型企业的应用也在逐渐普及，很多企业的应用服务都是构筑在其之上，例如 Web 服务、数据库服务、集群服务等等。因此，Linux 的安全性就成为了企业构筑安全应用的一个基础，是重中之重，如何对其进行安全防护是企业需要解决的一个基础性问题，基于此，本文将给出十大企业级 Linux 服务器安全防护的要点。

1、强化：密码管理

设定登录密码是一项非常重要的安全措施，如果用户的密码设定不合适，就很容易被破译，尤其是拥有超级用户使用权限的用户，如果没有良好的密码，将给系统造成很大的安全漏洞。

目前密码破解程序大多采用字典攻击以及暴力攻击手段，而其中用户密码设定不当，则极易受到字典攻击的威胁。很多用户喜欢用自己的英文名、生日或者账户等信息来设定密码，这样，黑客可能通过字典攻击或者是社会工程的手段来破解密码。所以建议用户在设定密码的过程中，应尽量使用非字典中出现的组合字符，并且采用数字与字符相结合、大小写相结合的密码设置方式，增加密码被黑客破解的难度。而且，也可以使用定期修改密码、使密码定期作废的方式，来保护自己的登录密码。

在多用户系统中，如果强迫每个用户选择不易猜出的密码，将大大提高系统的安全性。但如果 passwd 程序无法强迫每个上机用户使用恰当的密码，要确保密码的安全度，就只能依靠密码破解程序了。实际上，密码破解程序是黑客工具箱中的一种工具，它将常用的密码或者是英文字典中所有可能用来作密码的字都用程序加密成密码字，然后将其与 Linux 系统的/etc/passwd 密码文件或/etc/shadow 影子文件相比较，如果发现有吻合的密码，就可以求得明码了。在网络上可以找到很多密码破解程序，比较有名的程序是 crack 和 john the ripper 。用户可以自己先执行密码破解程序，找出容易被黑客破解的密码，先行改正总比被黑客破解要有利。

2、限定：网络服务管理

早期的 Linux 版本中，每一个不同的网络服务都有一个服务程序（守护进程，Daemon）在后台运行，后来的版本用统一的/etc/inetd 服务器程序担此重任。Inetd 是 Internet daemon 的缩写，它同时监视多个网络端口，一旦接收到外界传来的连接信息，就执行相应的 TCP 或 UDP 网络服务。由于受 inetd 的统一指挥，因

此 Linux 中的大部分 TCP 或 UDP 服务都是在/etc/inetd.conf 文件中设定。所以取消不必要服务的第一步就是检查/etc/inetd.conf 文件，在不要的服务前加上“#”号。

一般来说，除了 http、smtp、telnet 和 ftp 之外，其他服务都应该取消，诸如简单文件传输协议 tftp、网络邮件存储及接收所用的 imap/ipop 传输协议、寻找和搜索资料用的 gopher 以及用于时间同步的 daytime 和 time 等。还有一些报告系统状态的服务，如 finger、efinger、systat 和 netstat 等，虽然对系统查错和寻找用户非常有用，但也给黑客提供了方便之门。例如，黑客可以利用 finger 服务查找用户的电话、使用目录以及其他重要信息。因此，很多 Linux 系统将这些服务全部取消或部分取消，以增强系统的安全性。Inetd 除了利用/etc/inetd.conf 设置系统服务项之外，还利用/etc/services 文件查找各项服务所使用的端口。因此，用户必须仔细检查该文件中各端口的设定，以免有安全上的漏洞。

在后继的 Linux 版本中(比如 Red Hat Linux 7.2 之后),取而代之的是采用 xinetd 进行网络服务的管理。

当然，具体取消哪些服务不能一概而论，需要根据实际的应用情况来定，但是系统管理员需要做到心中有数，因为一旦系统出现安全问题，才能做到有步骤、有条不紊地进行查漏和补救工作，这点比较重要。

(来源：TechTarget 中国　作者：羽扇纶巾

构建企业级 Linux 服务器安全的十大要点（中）

3、严格审计：系统登录用户管理

在进入 Linux 系统之前，所有用户都需要登录，也就是说，用户需要输入用户账号和密码，只有它们通过系统验证之后，用户才能进入系统。

与其他 Unix 操作系统一样，Linux 一般将密码加密之后，存放在/etc/passwd 文件中。Linux 系统上的所有用户都可以读到/etc/passwd 文件，虽然文件中保存的密码已经经过加密，但仍然不太安全。因为一般的用户可以利用现成的密码破译工具，以穷举法猜测出密码。比较安全的方法是设定影子文件/etc/shadow，只允许有特殊权限的用户阅读该文件。

在 Linux 系统中，如果要采用影子文件，必须将所有的公用程序重新编译，才能支持影子文件。这种方法比较麻烦，比较简便的方法是采用插入式验证模块（PAM）。很多 Linux 系统都带有 Linux 的工具程序 PAM，它是一种身份验证机制，可以用来动态地改变身份验证的方法和要求，而不要求重新编译其他公用程序。这是因为 PAM 采用封闭包的方式，将所有与身份验证有关的逻辑全部隐藏在模块内，因此它是采用影子档案的最佳帮手。

此外，PAM 还有很多安全功能：它可以将传统的 DES 加密方法改写为其他功能更强的加密方法，以确保用户密码不会轻易地遭人破译；它可以设定每个用户使用电脑资源的上限；它甚至可以设定用户的上机时间和地点。

Linux 系统管理人员只需花费几小时去安装和设定 PAM，就能大大提高 Linux 系统的安全性，把很多攻击阻挡在系统之外。

4、设定：用户账号安全等级管理

除密码之外，用户账号也有安全等级，这是因为在 Linux 上每个账号可以被赋予不同的权限，因此在建立一个新用户 ID 时，系统管理员应该根据需要赋予该账号不同的权限，并且归并到不同的用户组中。

在 Linux 系统中的部分文件中，可以设定允许上机和不允许上机人员的名单。其中，允许上机人员名单在/etc/hosts.allow 中设置，不允许上机人员名单在/etc/hosts.deny 中设置。此外，Linux 将自动把允许进入或不允许进入的结果记录到/var/log/secure 文件中，系统管理员可以据此查出可疑的进入记录。

每个账号 ID 应该有专人负责。在企业中，如果负责某个 ID 的职员离职，管理员应立即从系统中删除该账号。很多入侵事件都是借用了那些很久不用的账号。

在用户账号之中，黑客最喜欢具有 root 权限的账号，这种超级用户有权修改或删除各种系统设置，可以在系统中畅行无阻。因此，在给任何账号赋予 root 权限之前，都必须仔细考虑。

Linux 系统中的/etc/securetty 文件包含了一组能够以 root 账号登录的终端机名称。例如，在 Red Hat Linux 系统中，该文件的初始值仅允许本地虚拟控制台（rtys）以 root 权限登录，而不允许远程用户以 root 权限登录。最好不要修改该文件，如果一定要从远程登录为 root 权限，最好是先以普通账号登录，然后利用 su 命令升级为超级用户。

5、谨慎使用：“r 系列”远程程序管理

在 Linux 系统中有一系列 r 字头的公用程序，比如 rlogin，rcp 等等。它们非常容易被黑客用来入侵我们的系统，因而非常危险，因此绝对不要将 root 账号开放给这些公用程序。由于这些公用程序都是用.rhosts 文件或者 hosts.equiv 文件核准进入的，因此一定要确保 root 账号不包括在这些文件之内。

由于 r 等远程指令是黑客们用来攻击系统的较好途径，因此很多安全工具都是针对这一安全漏洞而设计的。例如，PAM 工具就可以用来将 r 字头公用程序有效地禁止掉，它在/etc/pam.d/rlogin 文件中加上登录必须先核准的指令，使整个系统的用户都不能使用自己 home 目录下的.rhosts 文件。

6、限制：root 用户权限管理

Root 一直是 Linux 保护的重点，由于它权力无限，因此最好不要轻易将超级用户授权出去。但是，有些程序的安装和维护工作必须要求有超级用户的权限，在这种情况下，可以利用其他工具让这类用户有部分超级用户的权限。sudo 就是这样的工具。

sudo 程序允许一般用户经过组态设定后，以用户自己的密码再登录一次，取得超级用户的权限，但只能执行有限的几个指令。例如，应用 sudo 后，可以让管理磁带备份的管理人员每天按时登录到系统中，取得超级用户权限去执行文档备份工作，但却没有特权去作其他只有超级用户才能作的工作。

sudo 不但限制了用户的权限，而且还将每次使用 sudo 所执行的指令记录下来，不管该指令的执行是成功还是失败。在大型企业中，有时候有许多人同时管理 Linux 系统的各个不同部分，每个管理人员都有用 sudo 授权给某些用户超级用户权限的能力，从 sudo 的日志中，可以追踪到谁做了什么以及改动了系统的哪些部分。

值得注意的是，sudo 并不能限制所有的用户行为，尤其是当某些简单的指令没有设置限定时，就有可能被黑客滥用。例如，一般用来显示文件内容的/etc/cat 指令，如果有了超级用户的权限，黑客就可以用它修改或删除一些重要的文件。

(来源：TechTarget 中国　作者：羽扇纶巾)

构建企业级 Linux 服务器安全的十大要点（下）

7、追踪黑客踪迹：日志管理

当用户仔细设定了各种与 Linux 相关的配置（最常用日志管理选项），并且安装了必要的安全防护工具之后，Linux 操作系统的安全性的确大为提高，但是却并不能保证防止那些比较熟练的网络黑客的入侵。

在平时，网络管理人员要经常提高警惕，随时注意各种可疑状况，并且按时检查各种系统日志文件，包括一般信息日志、网络连接日志、文件传输日志以及用户登录日志等。在检查这些日志时，要注意是否有不合常理的时间记载。例如：

λ

正常用户在半夜三更登录；

不正常的日志记录，比如日志只记录了一半就切断了，或者整个日志文件被删除了；

用户从陌生的网址进入系统；

因密码错误或用户账号错误被摈弃在外的日志记录，尤其是那些一再连续尝试进入失败，但却有一定模式的试错法；

非法使用或不正当使用超级用户权限 su 的指令；

重新开机或重新启动各项服务的记录。

上述这些问题都需要系统管理员随时留意系统登录的用户状况以及查看相应日志文件，许多背离正常行为的蛛丝马迹都应当引起高度注意。

8、横向扩展：综合防御管理

防火墙、IDS 等防护技术已经成功地应用到网络安全的各个领域，而且都有非常成熟的产品。

在 Linux 系统来说，有一个自带的 Netfilter/Iptables 防火墙框架，通过合理地配置其也能起到主机防火墙的功效。在 Linux 系统中也有相应的轻量级的网络入侵检测系统 Snort 以及主机入侵检测系统 LIDS（Linux Intrusion Detection System），使用它们可以快速、高效地进行防护。

需要提醒注意的是：在大多数的应用情境下，我们需要综合使用这两项技术，因为防火墙相当于安全防护的第一层，它仅仅通过简单地比较 IP 地址/端口对来过滤网络流量，而 IDS 更加具体，它需要通过具体的数据包（部分或者全部）来过滤网络流量，是安全防护的第二层。综合使用它们，能够做到互补，并且发挥各自的优势，最终实现综合防御。

9、评测：漏洞追踪及管理

Linux 作为一种优秀的开源软件，其自身的发展也日新月异，同时，其存在的问题也会在日后的应用中慢慢暴露出来。黑客对新技术的关注从一定程度上来说要高于我们防护人员，所以要想在网络攻防的战争中处于有利地位，保护 Linux 系统的安全，就要求我们要保持高度的警惕性和对新技术的高度关注。用户特别是使用 Linux 作为关键业务系统的系统管理员们，需要通过 Linux 的一些权威网站和论坛上尽快地获取有关该系统的一些新技术以及一些新的系统漏洞的信息，进行漏洞扫描、渗透测试等系统化的相关配套工作，做到防范于未然，提早行动，在漏洞出现后甚至是出现前的最短时间内封堵系统的漏洞，并且在实践中不断地提高安全防护的技能，这样才是一个比较的解决办法和出路。

10、保持更新：补丁管理

Linux 作为一种优秀的开源软件，其稳定性、安全性和可用性有极为可靠的保证，世界上的 Linux 高手共同维护着个优秀的产品，因而起流通渠道很多，而且经常有更新的程序和系统补丁出现，因此，为了加强系统安全，一定要经常更新系统内核。

Kernel 是 Linux 操作系统的核心，它常驻内存，用于加载操作系统的其他部分，并实现操作系统的基本功能。由于 Kernel 控制计算机和网络的各种功能，因此，它的安全性对整个系统安全至关重要。早期的 Kernel 版本存在许多众所周知的安全漏洞，而且也不太稳定，只有 2.0.x 以上的版本才比较稳定和安全（一般说来，内核版本号为偶数的相对稳定，而为奇数的则一般为测试版本，用户们使用时要多留意），新版本的运行效率也有很大改观。在设定 Kernel 的功能时，只选择必要的功能，千万不要所有功能照单全收，否则会使 Kernel 变得很大，既占用系统资源，也给黑客留下可乘之机。

在 Internet 上常常有最新的安全修补程序，Linux 系统管理员应该消息灵通，经常光顾安全新闻组，查阅新的修补程序。一般情况下，用户可以随时保持对 Red Hat 门户网站（www.redhat.com），Debian Linux

门户网站（www.debian.org）、Turbolinux 门户网站（www.turbolinux.com）、SuSE 门户网站（www.suse.com/index_us.html）、Fedora 门户网站（fedora.redhat.com）等优秀 Linux 发行套件网站的关注，即时的更新系统的最新核心以及打伤安全补丁，这样能较好地保证 Linux 系统的安全。

(来源：TechTarget 中国　作者：羽扇纶巾)

本期电子书由 TechTarget 出品