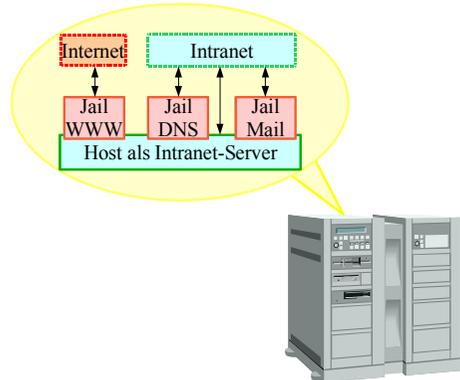


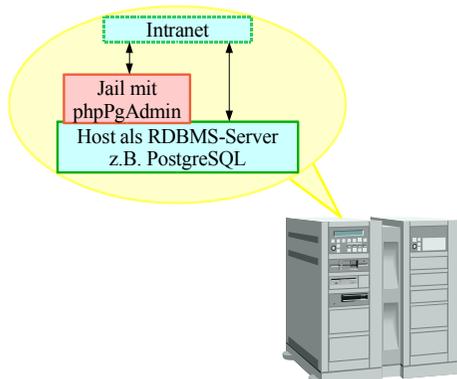
Anwendungsbeispiele

Die folgenden Anwendungsbeispiele zeigen die Leistungsfähigkeit der Jails.

- Server für kleine und mittlere Unternehmen müssen kostengünstig und selbstverständlich auch sicher sein. Ein Betrieb von Diensten wie HTTP, Email und DNS auf jeweils einem Rechner kann die Unterhaltskosten beachtlich steigern. Daher bietet es sich an nur einen Server anzuschaffen und jeden dieser Dienste in einer Jail zu installieren und somit spart das Unternehmen Kosten und gewinnt erheblich an Sicherheit.



- Sicherheit und Kosten von Datenbankservern im KMU-Bereich lassen sich durch Jails ebenfalls optimieren. In einer Jail installiert der Administrator das Web-Frontend der Datenbank und auf dem Hostsystem arbeitet das Datenbanksystem. Die Daten sind dabei vollkommen vor unerlaubten Zugriffen von außen geschützt.
- Um Netzwerkdienste zu testen oder kennenzulernen bieten sich ebenfalls Jails an. Dazu die betreffenden Dienste in einer Jail installieren und schon stehen sie für einen eingehenden Test oder zu Lernzwecken zur Verfügung.



Weitere Informationen

Wir hoffen Ihnen mit dieser kurzen Übersicht eine verständliche Vorstellung der FreeBSD Jails gegeben zu haben.

- Eine ausführliche Anleitung finden Sie unter: <http://www.grunix.de/dokumentationen/jails/>
- Die manual page zu den Jails ist sehr gut geschrieben: man `jail`
- Wenn Sie die Security-Features von FreeBSD interessieren, so erhalten Sie einen Überblick im Flyer: <http://www.allbsd.de/src/Flyer/FreeBSD/PDF/flyer-de-fbsd-security.pdf>
- Weitere Informationen rund um FreeBSD erhalten Sie im exzellenten und ausführlichen FreeBSD-Handbuch: http://www.FreeBSD.org/doc/de_DE.ISO8859-1/books/handbook/index.html
- Die Fachliteratur hat sich des Themas ebenfalls angenommen: FreeX 02/2007, Seite 24ff, Computer und Literatur-Verlag, Böblingen, ISSN 1436-7033
- Deutsches BSD-Forum: <http://www.BSDGroup.de/>

FreeBSD

Jails

Was sind Jails?

Jails umschreibt man am besten mit „FreeBSD in FreeBSD“. Eine Jail geht deutlich weiter als eine chroot-Umgebung. Eine Jail ist ein komplettes eigenständiges FreeBSD-System innerhalb eines FreeBSD Systems, wobei diverse Restriktionen auf einen Prozess und dessen Kindprozesse gesetzt werden. Prozesse in einer Jail können somit nicht auf Prozesse des Hostsystems zugreifen.

Dabei wird keine Hardware emuliert (wie bei vmware) oder ein eigener Kernel (wie bei XEN) genutzt, eine Jail teilt sich die Ressourcen mit dem Hostsystem. Das Hostsystem ist von Veränderungen, die innerhalb einer Jail stattfinden, nicht betroffen. Der Jail-Mechanismus ist dabei anderen virtuellen Realisierungen in Sachen wie Performance (insbesondere, wenn mehrere virtuelle Instanzen genutzt werden sollen) weit überlegen.

Sobald Sie als Administrator sich in der Jail bewegen, sehen Sie ein vollständiges FreeBSD, das sich genauso verwalten läßt wie jedes andere FreeBSD-System.

Die FreeBSD-Jails gehören zum Basissystem und erhöhen die Sicherheit signifikant. Neben einem kompletten System kann aber auch nur ein Prozess, ein Dienst, in eine Jail eingesperrt werden. Eine Jail ist ein System für sich, sprich, es kann dort die gleiche Software installiert werden wie auf einer normalen FreeBSD Installation.



Serverdienste in Jails

Gerade Serverdienste wie DNS, HTTP, FTP, SMTP/POP3/IMAP etc. waren in der Vergangenheit oftmals die Schwachstelle im System, welche ein Hacker für einen erfolgreichen Angriff ausnutzte. Auch wenn man immer up to date bleibt und eine Firewall nutzt, so kann man die Risiken immer nur minimieren.

Um diese auf ein Minimum schrumpfen zu lassen, bietet es sich an, diese Serverdienste in eine Jail zu sperren. Sei es jeden in eine einzelne Jail oder alle Dienste in eine Jail, Ihr Hostsystem wird im Falle eines Einbruchs über eine Schwachstelle eines Serverdienstes nicht beeinträchtigt werden.

Sicherheit durch Jails

Wenn ein Eindringling etwas verändert, wie können Sie dann sicher sein, dass er nicht noch mehr Schaden ange richtet hat, oder gar eine backdoor installiert hat? Ihr System an sich ist sicher? Was ist aber mit den installierten Diensten? Sind Sie sicher, dass hier nicht ein Bug vorhanden ist, den ein Angreifer ausnutzen könnte, um in Ihr System einzudringen? Was ist mit der downtime Ihres Servers bis sie diesen wieder bereinigt oder gar neu auf-gesetzt haben? Zeit ist Geld und mit einer Jail können Sie sich Mühe und vor allem Geld ersparen. Wurde Ihr Webserver, welcher selbstverständlich in einer Jail läuft, kompromittiert, so suchen Sie nicht lange nach veränderten Dateien.

Sie stoppen die Jail, kopieren Ihre Backup-Jail an die Stelle der alten, und starten diese wieder. Dies wird nicht mehr als eine Minute in Anspruch nehmen und Ihre Inter-netpräsenz ist wieder auf dem alten Stand. Danach kön-nen Sie offline die kompromittierte Jail untersuchen und den Fehler ausfindig machen.

Sie sehen, neben der richtigen Wahl des Betriebssystems, der Nutzung einer Firewall, eines IDS (Intrusion Detecti-on System) und weiteren Sicherheitsfeatures, sollten Sie auf die Möglichkeiten einer Jail keines Falls verzichten.

Ihr Server als Fort Knox

Sicher ist, dass nichts wirklich sicher ist, auch wenn Her-steller diverser Firewall-Lösungen oftmals anderes ver-sprechen. Firewalls sorgen für die erste Barriere, das Sys-tem an sich für die zweite, wenn Sie Jails nutzen, errich-ten Sie eine dritte Mauer und mit den FreeBSD security flags steht hinter dieser Mauer noch ein Tresor, der keine Tür hat. Der Einbruch in ein System ist schon schlimm genug, durch die Nutzung einer Jail nehmen Sie schon viel von dem Schrecken. Ärgerlich wird es, wenn der Ein-

dringling Daten verändert, Daten löscht und seine Spuren aus den diversen Logfiles löscht. Er braucht dazu admi-nistrative Rechte, also root-Zugriff, und diesen bekommt er nicht? Was aber wenn doch? Er kann machen was er will, oder doch nicht?

Nein, das kann er nicht, denn durch die Nutzung von security flags können Sie Ihre Verzeichnisse und Dateien so bearbeiten, dass nicht einmal mehr root diese zu verän-dern vermag. Wollen Sie auf diese Möglichkeit der Absi-cherung verzichten? Wohl kaum! Aber damit nicht genug: mit dem securelevel, welchen man während des laufendes Betriebs erhöhen aber nicht herabsetzen kann, verhindert man auch, dass ein Eindringling die Firewallregeln modi-fiziert, Kernelmodule lädt oder entlädt oder Disks zum schreiben öffnet. Mehr zu security flags und securelevel erfahren Sie im entsprechenden Dokument.

Mehrwert durch Jails

Eine Jail dient in erster Linie dazu, das System sicherer zu machen. Die weiteren Vorteile zeigen die Stärken des Konzepts:

- So können mehrere virtuelle Server, mit unterschiedli-chen Diensten, auf einem physikalischen Server betrie-ben werden. So lässt sich eine demilitarisierte Zone (DMZ) nur mit Jails auf einem Server aufbauen.
- Serverdienste sind meist sehr umfangreich und werden daher meist nicht von einem Administrator betreut. Ent-weder es haben mehrere User root-Rechte, oder Sie müssen mit Dingen wie „sudo“ experimentieren. Beides ist aus der Sicht der Sicherheit bedenklich! Lagern Sie daher die Dienste jeweils in eine Jail aus, so kann jeder Administrator seinen Dienst pflegen, ohne dass andere darauf Zugriff haben.
- Sie brauchen eine Testumgebung für Ihren Entwickler oder für bestimmte Dienste? Warum extra einen Server einrichten, ganz zu schweigen von den Kosten der Hardware? Richten Sie Ihrem Entwickler eine Jail ein und er darf in dieser schalten und walten, ohne das Hostsystem zu beeinträchtigen.
- Bei Schulungen müssen nicht alle Teilnehmer auf dem gleichen Server arbeiten. Jeder Teilnehmern kann je-weils eine Jail nutzen, um die gestellten Aufgaben zu er-arbeiten. Wenn die Schulung beendet ist, muss nicht der komplette Server neu aufgesetzt werden, es reicht aus, neue Jails an die Stelle der alten zu kopieren.
- Will man seinen Kunden root-Shells anbieten, wickelt man dies auf einem Server mit vielen Jails für viele Kunden ab. Die Anschaffung neuer Hardware entfällt,

die Einrichtung ist eine Angelegenheit von Minuten (oder wird automatisiert).

- Interessant ist auch der Aspekt, X-Window-Applikatio-nen in einer Jail zu betreiben. So lassen sich Bedienero-berflächen sauber vom z.B. Datenbank-Host trennen

**Die Fakten sprechen für sich:
mehr Sicherheit, mehr Übersicht, weniger Kosten,
bessere Ausnutzung der vorhandenen Hardware!**

Technisches

Die Implementierung von Jails ist nicht auf das Userland begrenzt. So gibt es im Kernel an zahlreichen sicherheits-relevanten Stellen Abfragen, ob ein Prozess auf dem Hostsystem oder in einer Jail ausgeführt wird. Dies wird vom Prozess-Management anhand eines Flags in der Pro-zesstabelle strengstens überwacht!

Eine Jail braucht in etwa 140 MB auf Speicherplatz auf der Festplatte, je nachdem, was in einer Jail installiert wird steigt auch der Speicherbedarf.

Aus Sicht der Administration wird eine Jail wie jedes an-dere System behandelt. So kann man sich beispielsweise via SSH in diese einloggen und hat hiernach keinen Zu-griff auf das Hostsystem. Aus Sicht des Hostsystems ist eine Jail ein Verzeichnis mit einer kompletten minimalen FreeBSD Installation, auf welche Sie vom Hostsystem Zugriff haben.

Um mit der Jail zu arbeiten, muss man sich aber nicht im-mer in diese selbst einloggen. Mit „jexec“ lassen sich Dienste und Programme vom Hostsystem aus in einer Jail starten und stoppen, die Prozessliste in der Jail oder die in der Jail angemeldeten User anzeigen.

Am Hostsystem ist jeder Jailprozess in der Prozessliste mit einem „J“ gekennzeichnet. So ist für den Betreiber ei-ner Jail auch ohne login in die Jail sofort ersichtlich, wel-che Dienste auf dem Server ausgeführt werden.

Eine Jail unterliegt auch gewissen Restriktionen, die man zum Teil über das Hostsystem und Kernel-Variablen (sysctl-MIBs) verändern kann. So sind keine „raw sockets“ (wie ping und traceroute) erlaubt, können aber über das Hostsystem ebenso wie das Setzen des Hostna-mes explizit für eine bestimmte oder alle Jails erlaubt werden. Ab FreeBSD 6.2 ist dürfen einer Jail auch mehre-re IP-Adressen zugeordnet werden. Genauso ist es mög-lich, IPv6-Adressen zu verwenden.

Der große Erfolg der Jails hat dazu geführt, dass diese mittlerweile auch als „Sysjail“ auf andere BSD-Derivate portiert werden. Ab Solaris Version 10 wurden sie als „Solaris-Zones“ implementiert.