Aufgabenstellung

Short Description

Nowadays any system connected to the Internet will be attacked by attackers anywhere in the world. Systems may be compormised due to suboptimal configurations (e.g. weaknesses in applications, services and in the operation system). To decrease this risk of succesful attacks system administrators should „harden“ their systems. There exist specific checklists and strategies to harden a given operating system. Argus Systems provides a so called Secure Application Environment for Linux Systems, named PitBull LX. This Software protects against application security flaws by isolating applications in separate security compartments.

The result of this thesis should lead to a lab, in which the hardening principles are outlined. The attendees of the lab should get hands on experience investigating various systems (a standard server, a hardened server).

Tasks

- Description of the Hardening Principles (properties of the servers, hardening concepts)
- Find examples which show the differences between a standard and a hardened system
- Installation and configuration of the three systems on one machine (VMWare)
- Set up a lab in which students can experience the hardening principles

Technologies

In this thesis, the students will focus on the following technologies and subjects

- Server hardening principles
- Linux as server
- VMWare
- Labs for students

Unterschrift .....

Abstract

Härten eines Systems bedeutet, die Sicherheit des Systems zu erhöhen. Erreicht wird dies durch geeignete Konfiguration der Programme und durch Abschalten der nicht benötigten Dienste. Im Rahmen dieser Arbeit wurden die verschiedenen Aspekte des Härtens eines Linux Systems gezeigt. Daraus resultierte eine Liste von Vorschlägen, die beim Härten eines Systems beachtet werden sollten. Diese Liste umfasst folgende Ebenen eines Systems: Physisch, Lokal, Datei- und Dateisysteme, Passwörter und Verschlüsselung, Betriebssystem-Kern und Netzwerk. Besonders ausführlich wurde PAM (Pluggable Authentication Module) beschrieben, da dieses Verfahren sehr verbreitet ist und PAM viele Möglichkeiten bietet, die Sicherheit der Authentifizierung zu erhöhen. Alle Angaben, die das Härten des Systems betreffen, wurden Distributionsunabhängig gemacht.

Weiter wurde eine Übung erarbeitet, die Fragen und deren Lösungen zu den Themen Härten eines Systems, PAM, ARP (Address Resolution Protocol) und rlogin (remote login) beinhaltet. Für diese Übung steht ein gehärterter und ein nicht gehärteter Red Hat Linux Server, fertig konfiguriert, auf einer CD-Rom bereit. Diese Server können direkt in eine VMware-Umgebung geladen und gestartet werden. Um die Übungsserver starten zu können braucht man eine VMware Version (auch als Trial-Version erhältlich) für das entsprechende Betriebssystem. Dadurch ist die Übungsdurchführung plattformunabhängig.

Management Summary

Einleitung

Für Unternehmen entstehen jährlich immense Kosten durch Angriffe über das Internet. Nach einer Schätzung von PricewaterhouseCoopers sind im Jahr 2000 Kosten in der Höhe von $\$1.6 * 10^{12}$ durch Viren entstanden. Aber nicht nur die verursachten Kosten, sondern auch der Schutz der Firmeninternen Daten beschäftigt die Unternehmen. Das sind Gründe, warum Sicherheit für viele Unternehmen immer wichtiger wurde. Sie wollen sich vor Angriffen schützen, um die damit verbundenen Kosten und Schäden zu minimieren. Jeder, der die Sicherheit seiner Systeme vernachlässigt, muss mit zusätzlichen Kosten und Schäden rechnen.

Zum Glück sind nicht nur die Angriffe ausgereifter geworden, sondern auch die Methoden, sich davor zu schützen. In den letzten Jahren sind neue Verfahren entwickelt worden, die es erlauben, sich auch gegen professionelle Angriffe zu schützen. Diese Arbeit beschäftigt sich mit dem sogenannten Härten eines Systems. Durch Härten eines Systems wird versucht, dieses System sicherer zu machen.

Ausgangslage

Der Auftrag dieser Arbeit kam von Prof. Dr. P. Heinzmann. Das Ziel war es, eine Übung zu erstellen. In dieser Übung sollte es möglich sein, verschiedene Prinzipien kennen zu lernen und zu erkennen, wie sie ein System sicherer machen. Einer dieser Vergleiche sollte zwischen einer manuellen Konfiguration und dem Produkt *PitBull LX* der Firma Argus [3] sein.

Ein weiteres Ziel war es, die theoretischen Grundlagen des Härtens zu erarbeiten. Trotz dieser Ziele war zu Beginn nicht klar, ob diese Ziele überhaupt erreicht werden können, da niemand genau wusste, was der Begriff Härten alles umfasst.

Vorgehen

Zuerst wurde versucht so viel wie möglich über das Härten von Systemen in Erfahrung zu bringen. Dazu wurde in verschiedenen Bibliothekskatalogen, Datenbanken mit Artikelverzeichnissen aus Fachzeitschriften und Wissenschaftsmagazinen, wie auch Internetforen und Webseiten gesucht. Parallel dazu fand die Einarbeitung in die Konfiguration eines Linuxservers, Anwendung von PitBull LX und VMware statt. Zwecks Qualitätskontrolle und Steigerung der Arbeitseffizienz wurde ein LaTeXGuide erstellt, der alle LaTeX-Befehle erläutert, die innerhalb des Berichtes verwendet werden würden.

Nach einer ersten Phase des Einarbeitens in die Thematik wurde versucht das Thema einzugrenzen und einen Übungsaufbau zu definieren. Für die Übung zeigte sich, dass erstellen von verschiedenen Attackensczenarios am aufschlussreichsten für das vergleichen verschiedener Serverkonfigurationen. In der Folge wurde versucht verschiedene Attacken zu inszenieren. Nach einer ausgiebigen jedoch erfolglosen Recherche, wurde entschieden den Übungsablauf zu ändern. Anstatt mit Attacken werden nun die Unterschiede mit sicherheitskritischen Konfigurationen gezeigt. Während den Recherchen wurden parallel praktische Versuche mit einem Linuxserver gemacht und das Vorgehen und die Erkenntnisse in LaTeX dokumentiert.

Ergebnisse

Im Rahmen dieser Arbeit entstand eine theoretische Abhandlung der Prinzipien des Härtens. Es wurden verschiedene Aspekte des Härtens erarbeitet: Physisch, Lokal, Datei- und Dateisysteme, Passwörter und Verschlüsselung, Betriebssystem-Kern und Netzwerk. Besonders ausführlich wurde das Verfahren PAM (Pluggable Authentication Module) beschrieben, da dieses sehr verbreitet ist und viele Möglichkeiten bietet, die Sicherheit der Authentifizierung zu erhöhen. Alle Angaben, die das Härten des Systems betreffen, wurden Distributionsunabhängig gemacht. Es wurde ein Übungs/-auf/-bau realisiert, der es ermöglicht die Prinzipien des Härtens anzuwenden und somit besser zu verstehen. Die Abbildung 1 zeigt eine Übersicht der Übungsinstallation. Für die Übungen stehen zwei komplett konfigurierten Server zur Verfügung: Ein nicht gehärteter und ein gehärteter Server (Server 1 und Server 2). Diese Server werden innerhalb von VMware gestartet. VMware simuliert die komplette Hardware eines Systems. Da VMware für die verbreitesten Betriebssysteme erhältlich ist, ist die Übung plattformunabhängig. Die Übung besteht aus verschiedenen Aufgaben bezüglich der Konfiguration eines System und den Diensten `rlogin` und `finger`. Der Dienst `rlogin` kann verwendet werden, um eine Verbindung zu einem Server

herzustellen. Mit dem Befehl `finger` können verschiedene Informationen über ein System gewonnen werden. Während der Übung können die Konfigurationseinstellungen überprüft werden, indem zwischen Server 1 und Server 2 die Dienste angewandt werden. Ebenfalls denkbar, ist das ausprobieren der Dienste auf dem simulierten Server eines anderen Übungsteilnehmers. Somit könnte die Übung auch zu zweit durchgeführt werden.

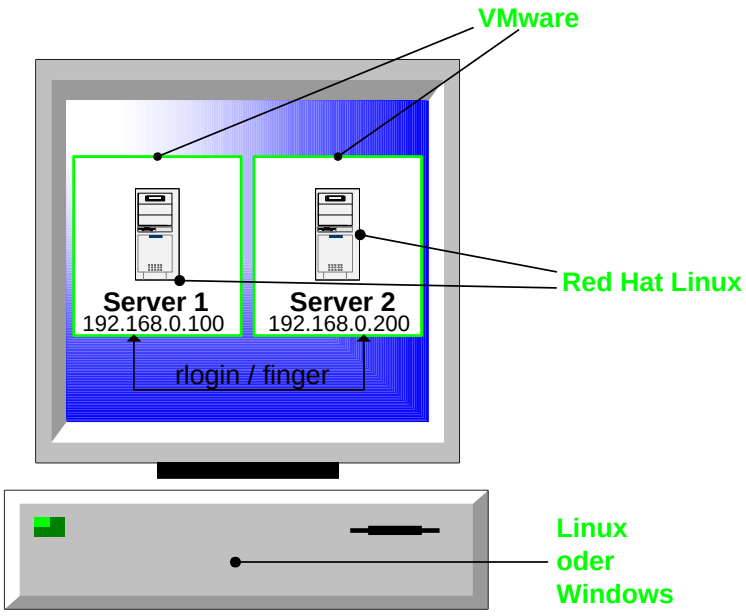

Abbildung 1: Übungsumgebung

Ausblick

Mit dem Härten kann ein Systems sehr sicher gemacht werden. Dennoch gibt es andere Techniken, welche die Sicherheit ebenfalls verbessern. Diese Techniken sind keine Konkurrenz zum Härten, sondern viel mehr eine Ergänzung. Um einen gesamthaften Überblick über die Thematik Sicherheit geben zu können, müsste man diese Techniken ebenfalls berücksichtigen. Zudem umfasst das Härten eines Systems mehr Punkte, als in diesem Dokument beschrieben wurden. Die Auswahl erfolgte nach der Wichtigkeit aus der Sicht der Autoren. Es ist denkbar, diese Punkte mit Fehlenden zu ergänzen. Die Ausführliche Dokumentation der Recherchen für die Übung ge-

eigneter Attacken zeigt, dass eine erfolgreiche Attacke, die dem Angreifer zu den Benutzerrechten eines Administrators verhilft nur mit sehr grossem Aufwand realisiert werden kann. Dies liegt vor allem daran, dass das Funnktionieren der Scripte von verschiedenen Variablen der Systemkonfiguration des anzugreifenden Systems abhängen. Dazu gehören unter anderen die Version des Betriebssystems (oft auch die Version des Kernels) und die Version des anzugreifenden Dienstes. Darum ist es schwieriger für ein konkretes System einen funktionierenden Exploit zu finden, als mit einem Exploit-Script auf dem ganzen Internet ein angreifbares System.

Die entstandene Übung kann beliebig erweitert werden. Die in der erarbeiteten Form bereitgestellten Fragen können je nach Anwendung und Anforderung umformuliert bzw. ergänzt werden.

Inhaltsverzeichnis

Teil I

Einführung

Es werden grundlegende Begriffe erklärt und es wird ein Einblick in die Thematik Sicherheit gegeben.

Kapitel 1

Einleitung

Diese Arbeit beschäftigt sich mit dem Härten eines System. Im Internet und in der Literatur stolpert man oft über diesen Begriff, wenn über Sicherheit geschrieben wird. Da der Begriff und der Vorgang des Härtens nicht im eigentlichen Sinne definiert sind, gibt es sehr unterschiedliche Meinungen darüber, was Härten bedeutet und was es alles umfasst. Die vorliegende Arbeit soll einen Eindruck vermitteln, was unter Härten eines Systems verstanden werden kann. Sie kann keine allgemeingültige Referenz sein, da je nach Einsatzgebiet eines Systems der Begriff unterschiedlich aufgefasst werden kann. Die Arbeit umfasst eine Zusammenstellung von Verfahren, die nach umfassendem Studium der Literatur den Autoren als wichtig erschienen. Welche davon im einzelnen Falle angewendet werden, hängt von der persönlichen Einstellung und der Verwendung ab. Da diese Arbeit nicht den Anspruch hat, jedes Gebiet der Sicherheit zu durchleuchten, beschränkt sie sich auf die folgenden Punkte:

- Härten eines Systems
- Linux als Betriebs- und Serversystem
- VMWare

und lässt folgende Punkte aus:

- Intrusion Prevention und Produkte die dieses Prinzip implementieren (z.B. Stormwatch)
- Secure Application Environment (basierend auf dem Produkt PitBull LX)
- Intrusion Detection

- Windows als Betriebssystem
- PitBull

Im Zusammenhang mit Härten treten in der Literatur verschiedene Begriffe auf, die z.T. verwirrend sein können. Im Englischen wird von „System Hardening“, „OS Hardening“ oder von „Application Hardening“ gesprochen. Mit dieser Unterteilung will man darauf hinweisen, dass es verschiedene Ebenen des Härtens gibt. Diese sind in Tabelle 1.1 zusammengefasst und erklärt. Wenn immer diese Unterteilung nicht von Bedeutung ist, wird der deutsche Begriff *Härten* bzw. *Härten eines Systems* für alle Ebenen verwendet. Die Definition, was darunter zu verstehen ist, wird in Kapitel 3.1 gegeben.

Tabelle 1.1: Unterteilung des Härtens und deren Bedeutung

Ebene	**Beschreibung**
OS Hardening	Diese Ebene umfasst das Härten des Betriebssystems. Das beinhaltet z.B. die richtige Konfiguration des Kernels aber auch die Netzwerkkonfigurationen.
Application Hardening	Darunter versteht man das Härten aller auf einem System laufenden Programme und Dienste.
System Hardening	Dies ist die Vereinigung der beiden oberen Ebenen. Dies bedeutet, System Hardening umfasst den gesamten Bereich, der unter Härten verstanden wird.

Weitere in diesem Dokument verwendeten Begriffe und Abkürzungen sind in einem Glossar auf Seite 69 zusammengefasst.

Das Literaturverzeichnis mit sämtlichen von uns verwendeten Referenzen befindet sich auf Seite 71.

Als Resultat der Arbeit ist eine Übung vorgesehen, die im Fach Internetsicherheit die Möglichkeit gibt, die hier erklärten Prinzipien und Verfahren selbst auszutesten, um sie besser zu verstehen.

Die Texte wurden bewusst nicht immer in rein „technischer Sprache“ verfasst. Die Autoren hoffen, dass die Texte dadurch angenehmer und verständlicher zu lesen ist.

Kapitel 2

Sicherheit

2.1 Was ist Sicherheit

Sicherheit hat sehr viele Aspekte. Diese können grundsätzlich in drei, für diese Arbeit relevanten, Hauptkategorien unterteilt werden:

- Schutz der Vertraulichkeit der Daten (Data confidentiality)
- Datenintegrität (Data integrity)
- Verfügbarkeit des Systems (System availability)

Der erste Punkt, Schutz der Daten, umfasst den Bereich der persönlichen Daten. Das System soll gewährleisten können, dass diese Daten geheim bleiben und nur den vom Besitzer festgelegten Personen zugänglich sind.

Unter Datenintegrität versteht man, dass Daten nur von berechtigten Benutzern verändert werden können bzw. detektiert werden kann, von wem die Änderungen vorgenommen wurden. Verändern heisst dabei auch löschen oder hinzufügen von falschen Daten.

Die Verfügbarkeit eines Systems ist ein sehr wichtiger Punkt im Bereich der Sicherheit. Es geht darum, dass niemand das System so stören kann, dass es unbenutzbar wird. Ein Stichwort hierbei ist „denial of service“. Personen die diese Attacken durchführen versuchen ein System so stark auszulasten, dass die bereitgestellten Dienste nicht mehr genutzt werden können.

Natürlich ist jedes einzelne der oben genannten Gebiete für sich sehr viel komplexer zu verstehen als hier angedeutet. Am technisch schwierigsten zu bekämpfen sind die denial-of-service Attacken. Für die beiden anderen Punkte gibt es bereits gute Lösungen.

Ein weiterer Aspekt, der immer zu heftigen Diskussionen führt, ist die Privatsphäre. Ein System soll die Person, die es benutzt, davor schützen, dass private Informationen über die Person selbst für Dritte einsehbar sind. Das führt schnell zu einer Diskussion über Moral bzw. zu der Frage nach der Legalität. Soll z.B. ein Staat ein Dossier über Computerbenutzer führen dürfen, die sie zur Bekämpfung von Internetkriminalität verwenden? Solche Themen sind sehr spannend, sollen aber nicht im Rahmen dieser Arbeit behandelt werden.

2.2 Warum Sicherheit

Die meisten Benutzer des Internets beachten das Gesetz. Das heisst sie versuchen nicht, andere zu schädigen. Leider sind offensichtlich nicht alle Benutzer so. Für sie gibt es einen englischen Begriff: „intruders“ (Eindringlinge) oder manchmal auch „adversaries“ (Gegner). Diese Personen können grundsätzlich in zwei Kategorien eingeteilt werden. Passive Eindringlinge möchten nur Daten lesen, für die sie keine Berechtigung haben. Aktive Eindringlinge möchten diese Daten auch ändern und sind somit eine grössere Gefahr. Zudem kann noch zwischen Attacken die auf ein Ziel gerichtet sind (targeted) und Attacken die ein zufälliges Ziel aussuchen (opportunistic) unterschieden werden.

Natürlich gibt es nicht nur Personen, die Schaden anrichten, sondern noch andere Arten von Gefahren. Dies kann ein Wurm (worms), ein Virus oder ein Trojanisches Pferd (trojan horse) sein. All das sind kleine Programme mit dem Ziel, Schaden auf einem Rechner anzurichten. Dabei ist es in den meisten Fällen für den Programmierer nicht entscheidend wo der Schaden entsteht. Es sind also zufällige Ziele.

Die Kosten die durch Viren verursacht werden sind beträchtlich. Sie werden für das Jahr 2000 auf $\$1.6 * 10^{12}$ geschätzt [1].

Zusätzliche Informationen und Berichte zu den Auswirkungen gibt es auf der Webseite von Securitystats [10]. Einen Guten Überblick und eine Einführung in das Thema Sicherheit bietet auch das Buch von Andrew. S. Tanenbaum [11].

2.3 Wie sicher ist sicher?

Eines muss schon zu Beginn klargestellt werden: Ein System ist und kann niemals vollkommen sicher sein. Es kann aber einem potenziellen Angreifer möglichst schwierig gemacht werden, in das System einzudringen.

[1] Schätzung von PricewaterhouseCoopers, 2000

Ein weiterer Punkt den es zu beachten gilt ist, dass ein System zwar sehr sicher gemacht werden kann, unter Umständen dadurch aber die Möglichkeiten des Systems eingeschränkt werden. Zum Beispiel kann ein Server ohne Netzwerkverbindung laufen, dadurch wird er aber nahezu unbrauchbar. Somit ist es immer ein Balanceakt zwischen Sicherheit und Flexibilität bzw. Brauchbarkeit. Man sollte sich also immer überlegen, für was das System eingesetzt werden soll. Weiter muss entschieden werden, wie wichtig die Daten sind, die zu schützen sind bzw. wie gross der Schaden ist, falls die Daten verloren gehen oder gestohlen werden. Ebenso wichtig ist es zu entscheiden, wem man vertraut und Zugriff auf das System gewährt. Diese Entscheide bestimmen den Grad der Sicherheit nach denen ein Sicherheitsplan erstellt werden sollte.

2.4 Was kann sicher gemacht werden?

Technisch gesehen gibt es auf einem System verschiedene Stufen, auf denen das System sicherer Gemacht werden kann. Eine sehr ausführliche Erklärung dazu kann im Linux Security-HOWTO [2] gefunden werden. Hier aber dennoch ein Überblick über die verschiedenen Stufen. Auf gewisse Punkte wird in dieser Arbeit in den folgenden Kapiteln genauer eingegangen.

- Physical Security (physikalische Sicherheit)
- Local Securtiy (lokale Sicherheit)
- Files and File system Securtiy (Datei- und Dateisysteme schützen)
- Password Securtiy and Encryption (Passwörter und Verschlüsselung)
- Kernel Securtiy
- Network Security (Netzwerksicherheit)

Dies ist eine technische Unterteilung. Auf Bereiche wie „Social Engineering“ oder „Security Awareness“ wird hier nicht eingegangen.

2.5 Probleme

Das Hauptproblem im Bereich der Sicherheit ist, wie oben schon angesprochen, dass ein System niemals sicher sein kann. Der Grund sind fehlerhafte Applikationen. Dass Applikationen Fehler aufweisen lässt sich aus verschiedenen Gründen nicht vermeiden, z.T. wegen des Zeitdruckes („time-to-market“) oder auch wegen der wachsenden

Komplexität der Programme. Das Einzige was dagegen gemacht werden kann, ist, sobald ein Fehler entdeckt wurde, ihn zu beheben und sofort einen Patch für die Kunden bzw. Benutzer des Programms bereitzustellen. Der Nachteil darin ist, dass der Patch erst gemacht werden kann, wenn das Sicherheitsproblem erkannt wurde (oft durch eine Attacke). Das gleiche gilt für Applikationen, die das System schützen sollen (z.B. ein Virenschutz). Diese Applikationen schützen das System nur vor bekannten Angriffen. Diese Systeme werden reagierende System (reactive Systems) genannt. Das heisst, sie können nur reagieren, was bedeutet, dass der Angreifer bereits in das System eingedrungen ist. Es gibt aber auch einen Ansatz der versucht, Attacken zu erkennen, bevor jemand ins System eindringen kann. Diese werden agierende System (proactive Systems) (Agierende Systeme) genannt. Sie versuchen gewisse Muster, die auf eine Attacke hinweisen, zu erkennen. Wie immer ist agieren die bessere Lösung als reagieren, aber sehr viel schwieriger zu realisieren.

In den folgenden Kapiteln werden verschiedene Ansätze zur Verhinderung von Attacken aufgezeigt und genauer erklären.

Teil II

Sicherheitstechnologien

Es werden verschiedene Technologien vorgestellt, die zur Sicherheit eines Systems beitragen können, bzw. für die Übung verwendet werden. Genauer betrachtet wird in diesem Teil die Technik des Härtens und PAM (Pluggable Authentication Module). Im Softwareteil befindet sich eine kurze Erläuterung der Funktionalität von PitBull, der Kommandos `rlogin`, `finger`, `arp` und eine Beschreibung der Konfiguration von VMware.

Kapitel 3

Härten eines Systems

Es ist die Aufgabe eines Administrators eines Systems, dieses so sicher wie möglich oder so sicher wie nötig zu machen. Um die Sicherheit zu erhöhen gibt es verschiedene, nahezu unendlich viele Möglichkeiten. Aus diesem Grund ist es schwierig, eine „Anleitung“ zu schreiben. Es kann hier lediglich eine Auswahl von Möglichkeiten gegeben werden, die helfen können, die Sicherheit zu erhöhen. Es soll einen Überblick geben und soll gleichzeitig ein Gefühl dafür geben, was alles beachtet werden sollte.

3.1 Was ist Härten eines Systems?

Eine Definition von „Härten eines Systems“ ist wie folgt:

> “System hardening is a step by step process of securely configuring a system to protect it against unauthorized access, while also taking steps to make the system more reliable. Generally anything that is done in the name of system hardening ensures the system is both secure and reliable.”[1]

Härten eines Systems bedeutet damit, ein System so zu konfigurieren, dass es sicher und zuverlässig ist. Dies umfasst nicht nur, bekannte Sicherheitsprobleme zu lösen bzw. zu umgehen, sondern es umfasst auch die Frage, welche Services auf dem System laufen müssen.

Oft wird eine Unterteilung gemäss Tabelle 3.1 gemacht. In dieser Arbeit wird diese Unterteilung nicht gemacht, da es oft schwierig ist, einzelne Punkte in eine Kategorie einzuteilen. Ausserdem ist es nicht entscheidend, in welche Kategorie diese Punkte fallen, wichtig ist nur, dass sie Sicherheitsrelevant sind und deshalb auch

[1] Definition von `http://www.itcoach.com/unsafe/System-Hardening.htm`

beachtet werden. Aus diesem Grund ist der Prozess des Härtens nicht in Kategorien unterteilt, sondern wird anhand eines Ablaufes dargestellt, der alle Kategorien enthält.

Tabelle 3.1: Mögliche Unterteilung des Härtens in verschieden Kategorien

Kategorie	**Beschreibung**
Physisch	Der physische Zugang zum System wird verhindert bzw. geregelt (z.B. wird der Computer in einen Schrank eingeschlossen).
Lokal	Die Rechte der lokalen Benutzer (user mit einem account) werden gesetzt.
Dateien und Dateisysteme	Es wird ein geeignetes Dateisystem gewählt und Dateiattribute werden gesetzt.
Passwörter und Verschlüsselung	Passwörter werden überprüft und es wird wenn immer möglich Verschlüsselung eingesetzt.
Betriebssystem-Kern	Kernel Optionen werden in Hinblick auf die Sicherheit entsprechend gesetzt bzw. verändert.
Netzwerk	Der Zugriff über ein Netzwerk wird so sicher wie möglich gemacht.

3.2 Prinzipien des Härtens

Es gibt verschiedene Ansätze, ein System zu Härten. Zum einen können alle Einstellungen manuell gemacht werden. Das heisst, die Konfigurationen der Programme, die auf dem System laufen, werden von Hand angepasst um deren Sicherheit zu erhöhen. Eine weitere Möglichkeit ist, Linux-Tools zu Hilfe zu nehmen. Diese versuchen die Konfiguration der Programme nach den Aspekten der Sicherheit vorzunehmen. Die dritte Möglichkeit ist schliesslich, ein Produkt einzusetzen, welches das Prinzip des Härtens implementiert. Die zwei letzten Varianten unterscheiden sich grundsätzlich nicht durch die Art und Weise, wie sie das Problem angehen. Meist unterstützen aber kommerzielle Produkte nicht nur das Härten, sondern haben noch andere sicherheitsrelevanten Teile implementiert. Die verschiedenen Varianten sind in Tabelle 3.2 zusammengefasst:

Tabelle 3.2: Verschiedene Varianten des Härtens

Variante	**Beschreibung**
Manuell	Die Konfiguration wird von Hand vorgenommen. Der Administrator ist für die Einstellungen Verantwortlich
Linux-Tools	Das Tool übernimmt die Konfiguration der Programme.
Kommerzielle Produkte	Diese Produkte übernehmen die Konfiguration der Programme, übernehmen aber oft noch zusätzliche Funktionen wie Intrusion Detection, etc.

3.3 Manuelles Härten

Bei der manuellen Konfiguration der Programme, kann das System sehr spezifisch eingerichtet werden. Das ist ein grosser Vorteil gegenüber der automatischen Anpassung der Konfigurationen, bedeutet aber auch, dass der Systemadministrator mehr Wissen über die auf dem Server eingesetzte Software haben muss. Grundsätzlich sollte man sich dabei folgende Punkte überlegen:

- Welche Dienste werden auf dem System wirklich benötigt?
- Welcher Benutzer hat welche Berechtigungen auf dem System?
- Wie ist das System zu Härten, damit bei einer Kompromittierung nicht das ganze System betroffen ist?
- Welche Programme sind geeignet, um unberechtigte Zugriffe zu erkennen und aufzeichnen zu können?

Diese Punkte sind entscheidend und gelten nicht nur für die manuelle Konfiguration, sondern für alle Vorgehensweisen. Am meisten Gedanken wird man sich aber bei der manuellen Konfiguration machen müssen. In der vorliegenden Arbeit geht es nicht um das Erkennen von unberechtigten Zugriffen, trotzdem sollte man wissen, dass es auch zum Härten eines Systems gehört.

3.3.1 Ablauf vor und während der Installation

Folgendes ist ein Vorschlag, ein System zu Härten. Die Liste ist nicht abschliessend, sie soll viel mehr einen Überblick geben, worum es in dem Prozess geht und dabei soll verständlich gemacht werden, was Härten eines Systems bedeutet.

1. Wählen Sie ein BIOS Passwort und stellen Sie sicher, dass das System nur von der Festplatte gestartet werden kann (booten mit Diskette, etc. abstellen). Damit verhindern Sie, dass jemand, der physischen Zugriff zum Rechner hat, diesen mit einer Diskette oder CD Rom booten kann.

2. Wählen Sie ein geeignetes Dateisystem und partitionieren Sie die Festplatte. Was eine sinnvolle Partitionierung bedeutet hängt vom Einsatz des Systems ab. Zum Beispiel sollte ein System, das als Mailserver eingesetzt wird, eine eigene Partition für den Mail Spooler[2] haben (`/var/mail` und/oder `/var/mail/spool`). Damit wird verhindert, dass ein Benutzer den Mail Spool und damit die ganze Festplatte füllen kann. Folgende Punkte sind Vorschläge, die beachtet werden sollten.

   - Es sollten sich alle Verzeichnisse, auf die ein Benutzer Schreibrecht hat, auf einer eigenen Partition befinden. Dies umfasst zum Beispiel `/home` und `/tmp`. Das reduziert das Risiko einer „user DoS“[3] Attacke.
   - Partitionen die dynamisch ihren Inhalt ändern, wie z.B. `/var` sollten ebenfalls auf eine eigene Partition, da sie sonst in kurzer Zeit überfüllt werden könnten.
   - Partitionen, auf denen Software installiert wird, die nicht zur Distribution gehört, sollten auf einer eigenen Partitionen sein. Damit wird die Software bei einer allfälligen Neuinstallation nicht gelöscht. Das hat nicht direkt mit dem Härten eines Systems zu tun, hilft aber im Falle einer Neuinstallation.
   - Daten, die statisch sind, sollten auf einer eigenen Partition sein. Diese kann dann read-only gemountet werden.

   Das gewählte Dateisystem kann im Fall eines Absturzes Auswirkungen haben. Das Standarddateisystem ist `ext2`. Es wird aber empfohlen, ein journalling Dateisystem wie z.B. `ext3` zu wählen. Die Wahrscheinlichkeit Daten zu verlieren ist damit kleiner (zudem ist `ext3` im Betrieb schneller).

3. Wählen Sie ein sicheres root Passwort[4]. Das ist ein grundlegender Schritt um ein sicheres System zu bekommen. Am Besten verwenden Sie dazu einen Pass-

[2] Ein Mail Spooler ist für die Versendung und das Verteilen von Mails zuständig

[3] Das Bedeutet, ein Benutzer könnte die root Partition (`/`) mit Daten auffüllen und damit das System unbenutzbar machen.

[4] ein sicheres Passwort ist mind. 6 Zeichen lang, hat mind. 2 alphabetische Zeichen und 1 numerisches oder 1 spezial Zeichen. Das Passwort muss vom user login verschieden sein und darf auch nicht davon abgeleitet werden können. Es darf zudem nicht von einem Wort oder einer Wortkombination abgeleitet werden können.

wortgenerator. Diese generieren je nach Programm sehr sichere Passwörter (z.B. *makepasswd* unter Linux).

4. Schalten sie „Shadow Password“ ein. Shadow Password bedeutet, das Passwort wird in `/etc/shadow` gespeichert und kann nur noch vom user `root` und von der Gruppe `shadow` gelesen werden. Damit wird verhindert, dass ein Angreifer eine Kopie der Passwortdatei bekommen kann und einen Passwort Cracker darauf ansetzt.

5. Finden Sie heraus, welche Dienste auf dem System laufen. Installieren und starten Sie nur die benötigten Dienste. Dienste sind Programme wie z.B. ein ftp-Server (z.B. *Proftp*) oder ein Webserver (*Apache*). Überlegen Sie sich, was sie auf Ihrem System wirklich anbieten wollen und schalten Sie alle anderen Dienste aus. Der Grund liegt darin, dass Netzwerkdienste auf eine Verbindung von ausserhalb warten. Diese Dienste können (auch unbekannte) Sicherheitslücken aufweisen und sind damit eine Gefahr. Stellen Sie weiter sicher, dass beim Aufstarten kein Dienst, den sie nicht brauchen, versucht wird vom Betriebssystem zu starten.

6. Überprüfen und berichtigen Sie die Zugriffsberechtigungen auf die Dienste. Es sollen nur Benutzer Zugriff auf einen Dienst erhalten, die ihn wirklich benötigen. Oft ist das nur der user root.

7. Installieren Sie nur die Pakete bzw. Programme, die Sie wirklich auf Ihrem System benötigen. Jede übliche Linux Distribution (z.B. SuSE, RedHat, Debian, etc.) beinhaltet tausende von Programmen. Wählen Sie aus denen nur die benötigten aus, damit Sie nicht Programme installiert haben, die es einem Angreifer erleichtern können eine Attacke durchzuführen. Erleichtert wird die Auswahl bei den meisten Distributionen dadurch, dass die Programme in Kategorien unterteilt sind, die nach Bedarf komplett ausgeschaltet werden können. Achten Sie bei der Auswahl der Programme darauf, dass Sie als sicher geltende Programme installieren.

8. Updaten aller installierten Pakete, um jedes Paket auf den aktuellsten Stand zu setzen.

9. Gewünschte „user accounts“ anlegen. Dabei soll beachtet werden, dass sichere Passwörter verwendet werden. Dabei gelten die gleichen Regeln wie beim Anlegen eines root Passwortes.

3.3.2 Ablauf nach der Installation

Wenn die im letzten Abschnitt beschriebenen Schritte durchgeführt sind, ist das System installiert und lauffähig. Trotzdem kann noch mehr getan werden, um das System sicher zu machen. Dies ist wieder eine Auswahl, die nicht abschliessend ist.

1. Setzen Sie ein Bootmanager (*Lilo* oder *Grub*) Passwort. Haben Sie kein Bootmanager Passwort, kann jemand, der root Berechtigung auf dem System erlangt hat, das root Passwort ändern und das System neu starten. In diesem Fall haben Sie keinen Zugriff mehr auf Ihr System, da sie das Passwort nicht mehr kennen. Die einzige Lösung ist dann eine komplette Neuinstallation. Mit einem Bootmanager Passwort verhindern Sie den Neustart ohne Passwort. Das gewählte Passwort sollte natürlich verschieden vom root Passwort sein.

2. Schränken Sie den Zugriff auf das System über das Netz ein. Das bedeutet, ein Benutzer muss sich zuerst mit einen Benutzernamen/Passwort einloggen, um dann mit `su` oder `sudo` root Rechte zu bekommen. Damit wird verhindert, dass sich jemand direkt als root einloggen kann und hat zudem den Vorteil, dass ein direkter Angriff auf das root Passwort sinnlos wird, da zuerst ein Benutzerpasswort gefunden werden muss.

3. Falls Personen physischen Zugriff zum System haben, erlauben Sie nur bestimmten Personen, beim System einen Neustart durchzuführen. Wenn Sie die Standardeinstellung beibehalten kann *jeder* das System neu starten (mit der Tastenkombination `Ctrl-Alt-Del`).

4. Achten Sie beim mounten von Partitionen darauf, dass Sie auf bestimmten Partitionen keine Ausführung von Programmen erlauben (z.B. auf der Partition `/tmp`). Überprüfen Sie die Mountingtabelle (`/etc/fstab`) und setzen Sie die Optionen entsprechend[5].

5. Informieren Sie sich über Sicherheitsupdates die Ihre Software betrifft. Am besten in einer Security Announce Mailing List.

6. Mit PAM (Pluggable Authentication Module) können Sie festlegen, wie Benutzer von den Programmen authentifiziert werden. Das heisst, der Zugriff auf Programme kann geregelt werden. Da die Authentifizierung ein sehr zentraler Punkt ist, ist dieser ein eigener Abschnitt gewidmet. Siehe 3.5.

7. Limitieren Sie die Resourcen, die ein Benutzer belegen kann. Ohne diese Massnahme kann *jeder* Benutzer so viel CPU Rechenleistung und Speicher belegen

[5]Details zu den Optionen erhalten Sie in der Manual Page des mount Befehls (`man mount`)

wie er will. Dies bedeutet, jeder kann das System auslasten bzw. überlasten. Sie können mit PAM (s. oben) einstellen, wieviel Rechenleistung, Speicher, etc., Sie jedem Benutzer zuteilen wollen. Genau genommen können mit PAM alle Resourcen des Systems eingeteilt werden. Damit ein Benutzer nur einen Teil des Festplattenspeichers belegen kann, können Sie pro Benutzer eine quota anlegen. Detaillierte Informationen dazu können unter `http://seifried.org/lasg/users/` gefunden werden.

8. Wenn sich ein Benutzer mit einem falschen Benutzernamen und/oder Passwort einloggt, muss er eine kurze Zeit warten, um einen neuen login promt zu erhalten. Erhöhen Sie diese Zeit dynamisch, das heisst, bei jedem Fehlversuch wird die Zeit erhöht. Damit erschweren Sie einem potentiellen Angreifer einen „Brute Force“[6] Angriff auf das System, da es sehr zeitaufwändig wird. Sie können den Benutzer auch nach 3 Fehlversuchen sperren. Sie sollten die Login-Versuche in einer Logdatei festhalten.

9. Falls es nötig ist, dass Benutzer auf dem System root Rechte bekommen, sollten Sie statt dem Befehl `su` den Befehl `sudo` verwenden. Dieser bietet mehr Möglichkeiten, z.B. kann in der Konfigurationsdatei bestimmt werden, welche Befehle von welchem Benutzer ausgeführt werden dürfen. Zusätzlich gibt es bei der Eingabe eines falschen Passwortes einen Eintrag in der Logdatei und diese wird dem Administrator als Mail zugesandt.

10. Limitieren Sie die Rechte des Benutzers und zeichnen Sie dessen Aktivitäten auf. Zum Beispiel sollten die Rechte der Dateien angepasst werden, damit der Benutzer nur die Dateien lesen kann, die nötig sind. Es können auch die Standardvorgaben für das Erstellen von Dateien geändert werden, damit diese nur noch vom Benutzer selbst lesbar sind (`umask`). Im Bereich der Aufzeichnungen gibt es viele Möglichkeiten, angefangen vom Aufzeichnen des Anmeldens bis hin zum Aufzeichnen jedes einzelnen Zeichens, das der Benutzer eingibt. Hier ist ein sinnvolles Mittelmass zu finden (*syslog*).

11. Überprüfen Sie die Passwörter, die von den Benutzern gewählt wurden, oder noch besser, lassen Sie nur sichere Passwörter zu, in dem Sie bereits beim Erstellen Wortlisten verwenden und dem Benutzer nur erlauben, Passwörter zu benutzen, die mit keinem Wort in den Wortlisten in Verbindung gebracht werden können. Am Besten verwenden Sie für die Überprüfung *cracklib* mit

[6] Dabei werden alle möglichen Paare von Benutzername/Passwort systematisch durchprobiert bis ein gültiges Paar gefunden wurde

einer der Wortliste *cracklib_dict*. Für die Passwörter gelten die Anforderungen, die in der Fussnote 4 beschrieben sind.

Um nachträglich zu überprüfen ob die gewählten Passwörter sicher sind, gehen sie genauso vor wie ein Angreifer. Verwenden Sie einen Passwort cracker wie z.B. *john*[7] zusammen mit Wortlisten. Wenn Sie ein Passwort knacken können, haben Sie eine Sicherheitslücke entdeckt und können entsprechende Massnahmen ergreifen.

12. Es ist wichtig zu entscheiden, wer Zugriff zu den Logdateien hat, da diese oft sehr genaue Informationen über ein System liefern. Zudem werden sie nutzlos, wenn ein Angreifer die Logdateien nach einem erfolgreichen Angriff ändern kann.

13. Sie können Ihr System sicherheitstechnisch in vielen Hinsichten verbessern, wenn Sie den Kernel optimieren. Viele Distributionen bieten „Kernel Patches“ an, die die Sicherheit verbessern. Viele Änderungen können zur Laufzeit angepasst und verändert werden. Dazu dient der Befehl `sysctl`. Er erlaubt zum Beispiel das Ignorieren von icmp requests, aber natürlich noch vieles mehr. Informationen dazu finden Sie ebenfalls in der Manpage (`man sysctl`).

14. Sobald Ihr System fertig konfiguriert ist, machen Sie ein Snapshot von dem System. Dies beinhaltet ein `md5sum` der wichtigsten Verzeichnisse (z.B. boot). Damit können Sie jederzeit überprüfen, ob ihr System noch im originalen Zustand ist.

Das ist der Ablauf, um ein System zu Härten. Die konkreten Einstellungen die gemacht werden müssen, sind in diesem Ablauf nicht beschrieben, da sie Distributionsabhängig sind. Für Debian können die Einstellungen im Securing Debian Manual [9] nachgelesen werden. Für andere Distributionen gibt es jeweils sehr umfangreiche Dokumentationen, die auf den Webseiten gefunden werden können.

Beispiel ssh

Da im obigen Ablauf nur die Schritte, nicht aber die konkreten Einstellungen zum Härten eines Systems beschrieben wurden, soll hier stellvertretend für andere Dienste der *ssh* Dienst genauer betrachtet werden.

Falls Sie anstatt *ssh* immer noch *telnet* verwenden, sollten Sie *telnet* abschalten und stattdessen als externe Zugriffsmöglichkeit nur noch *ssh* verwenden. Anschliessend sollten Sie folgende Einstellungen in der *OpenSSH* Konfigurationsdatei machen:

[7] Dieses Programm kann von `http://www.openwall.com/john/` heruntergeladen werden.

- `ListenAddress IPx`
  ssh soll nur auf einem „Interface“ auf Verbindungen warten. Diese Einstellung ist nötig, falls Sie mehrere Netzwerkkarten haben, damit Sie die Zugriffe besser kontrollieren können. Ausserdem können Sie den Zugriff einschränken, indem Sie z.B. den Zugriff nur von innerhalb Ihres Netzes zulassen, von Ausserhalb jedoch sperren.

- `PermitRootLogin No`
  Es darf sich niemand direkt als root einloggen. Will jemand root Rechte via *ssh* bekommen, ist zweimaliges Einloggen erforderlich. Das heisst ein „Brute Force“ Angriff auf das root Passwort wird sinnlos, da auch ein gültiges root Passwort nicht akzeptiert wird.

- `Listen Port xy`
  Ändern Sie den port (Standardport für *ssh* ist Port 22), auf den *ssh* auf Verbindungen wartet. Damit kann niemand ganz sicher sein, ob der *ssh* Dienst wirklich läuft. *Wichtig:* Diese Option ist „security by obscurity“ (Sicherheit durch Verwirrung) und dient dazu, unerfahrene Angreifer zu verwirren.

- `PermitEmptyPasswords no`
  Leere Passwörter dürfen niemals erlaubt werden.

- `AllowUsers name ref user@host`
  Es wird nur bestimmten Benutzern erlaubt, sich via *ssh* einzuloggen.

- `AllowGroups wheel admin`
  Nur Mitglieder der angegebenen Gruppen (in diesem Beispiel die Gruppe `wheel` und `admin`) dürfen sich via *ssh* einloggen.

- `PasswordAuthentication yes`
  Benutzer dürfen sich mit einem Passwort anmelden. Die bessere Lösung ist, die Option auf `no` zu setzen, um nur den Zugriff via einem ssh-key zuzulassen. Ein Login via ssh-key basiert auf dem Prinzip der asymmetrischen Verschlüsselung. Das heisst, der Benutzer hat einen „private key“ den sonst niemand kennt. Dieser wird zur Authentifizierung benutzt und kann damit nicht durch einen „Brute Force“ Angriff herausgefunden werden, was grundsätzlich bei einer Authentifizierung mit einem Passwort möglich ist.

- `Protocol 2`
  Es soll nur die Protokoll Version 2 erlaubt sein. Version 1 hat einige Sicherheitslücken.

3.4 Automatisches Härten

Im vorangegangenen Abschnitt wurde beschrieben, wie ein System von Hand gehärtet werden kann. Das sieht nach sehr viel Arbeit aus und tatsächlich ist es das auch. Man könnte sich die Frage stellen, ob es nicht ein Programm gibt, das einem die ganzen Einstellungsarbeiten abnimmt und den Vorgang des Härtens automatisiert. Tatsächlich gibt es verschiedene Ansätze, die genau das versuchen. Trotzdem muss gesagt werden, dass ein Programm nicht die ganze Arbeit ausführen kann. Sicherheit ist ein Prozess, der nicht bei der Konfiguration aufhört. Es gibt dauernd neue Programme, neue Angriffe, neue Gefahren. Ein Programm, dass ein System automatisch härtet, kann diesen Anforderungen nur in einem gewissen Masse nachkommen. Das bedeutet, auch beim Einsatz solcher Tools muss der Administrator seinen Teil leisten und muss mit dem System und mit den Gefahren vertraut sein. Es ist ein Irrtum, jeder könne mit dem geeigneten Programm ein sicheres System haben.

Im Folgenden werden zwei Produkte vorgestellt, die das Härten automatisieren. Das eine ist das Paket *harden*, das andere ist *Bastille Linux*.

3.4.1 Harden

Dieses Paket versucht es dem Administrator zu erleichtern, ein System zu installieren, das sicher und leichter zu administrieren ist. Es bringt eine schnelle Hilfe bei der Installation. Es deinstalliert Pakete mit bekannten Sicherheitsproblemen oder, soweit als möglich, auch Programme, die Klartext über ein Netz übermitteln, etc. Es installiert auch Tools, die die Sicherheit des Systems erhöhen sollen (Intrusion Detection Tools, Analysetools, etc.). Konkret wird folgendes installiert:

- `harden-doc`
  Dokumentationen
- `harden-tools`
  Tools um die Sicherheit des Systems zu erhöhen, wie z.B. kernel patches, intrusion detection, etc.
- `harden-environment`
  Hilft eine Umgebung zu konfigurieren.
- `harden-servers`
  Entfernt Server, die als nicht sicher gelten (z.B. *telnetd*).
- `harden-clients`
  Entfernt Clients, die als nicht sicher gelten (z.B. *telnet*).

- `harden-remoteflaws`
  Entfernt Programme mit Sicherheitslücken, die es einem entfernten Angreifer erlauben, das System zu kompromittieren (Welche Programme das sind, hängt von den Programmen und deren Versionen ab).

- `harden-localflaws`
  Entfernt Programme mit Sicherheitslücken die es einem lokalen Angreifer erlauben, das System zu kompromittieren (Welche Programme das sind, hängt von den Programmen und deren Versionen ab).

- `harden-remoteaudit`
  Tools um ein System von einem entfernten Rechner aus zu überwachen.

Das *harden* Paket wird ständig erweitert und es ist deshalb möglich, dass diese Aufzählung nicht mehr aktuell ist. Die aktuelle Version kann von `http://ftp.debian.org/debian/pool/main/h/harden/` heruntergeladen werden.

3.4.2 Bastille Linux

Bastille Linux ist ein Tool, das ein System automatisch härtet. Ursprünglich wurde es für *Red Hat* und *Mandrake* entwickelt, funktioniert mittlerweilen aber auch für andere Distributionen. Bei *Bastille Linux* gibt es verschiedene Modi, aus denen einer gewählt werden kann:

- *Bastille Linux* versucht anhand von Antworten zu Sicherheitsfragen das System zu konfigurieren. Diese Variante nennt sich *InteractiveBastille*.

- Sicherheitsstufen (locker, moderat, paranoid) können gewählt werden. Je nach der gewünschten Stufe wird das System konfiguriert. Diese Variante nennt sich *BastilleChooser*

- Es wird eine bereits definierte Konfigurationsdatei genommen (z.B. von *Bastille Linux*). Anhand dieser Datei wird ein Sicherheitskonzept auf dem System implementiert. Diese Variante nennt sich *AutomatedBastille*.

Weitere Informationen zu diesem Projekt können auf `http://www.bastille-linux.org` gefunden werden.

3.5 Authentifizierung

Ein wichter Aspekt der Sicherheit ist die Authentifizierung der Benutzer durch das System. Immer, wenn sich ein Benutzer einloggt, wird überprüft, ob er Zugriff zum

System hat und welche Rechte ihm gegeben werden sollen. In den Anfängen des Computerzeitalters hat es gereicht, einen Benutzer anhand des Namens und eines Passwortes zu identifizieren. Man glaubte jedem, der ein korrektes Passwort eingegeben hat, dass er auch wirklich derjenige Benutzer sei. Seit dem die Computer mehr und mehr vernetz wurden, kam der Anspruch dazu, dass man kontrollieren kann, woher sich ein Benutzer einloggt und was er auf dem System machen darf. Daraus sind komplexere Authentifizierungsmechanismen entstanden, die im Folgenden erklärt werden. Dies soll helfen zu verstehen, was genau passiert, wenn man sich auf einem System über ein Netzwerk oder lokal einloggt.

3.5.1 Prozesse

Um verstehen zu können, was bei einem login passiert wird hier ein kleiner Überblick der Prozesse unter Linux gegeben. Beim Starten des Systems wird zuerst der Kernel in den Hauptspeicher geladen. Dieser lädt die Treiber für die Hardware und initialisiert sie. Anschliessend wird die root Partition (`/`) gemountet und der erste Prozess gestartet. Dies ist der *init* Prozess. Dieser ist der Vater aller Prozesse. Von ihm werden alle anderen Prozesse abgeleitet. Dies kann man gut nachvollziehen, wenn man den Befehl `pstree` in der Konsole eingibt. Eine mögliche Ausgabe könnte folgendermassen aussehen:

```
se@mandy: pstree -u
init-+-6*[agetty]
     |-bdflush
     |-cron
     |-devfsd
     |-dhcpcd
     |-eth0
     |-kapmd
     |-keventd
     |-klogd
     |-kreiserfsd
     |-ksoftirqd_CPU0
     |-kswapd
     |-kupdated
     |-lockd
     |-master-+-pickup
     |        `-qmgr
     |-portmap
```

```
|-rpciod
|-sshd---sshd---sshd---bash(se)---pstree
`-syslogd
```

Alternativ können die Prozesse und deren Hirarchie auch mit dem `ps` Befehl angeschaut werden. Dann sieht die Ausgabe bei gleichen Prozessen folgendermassen aus:

```
se@mandy: ps ax -H -o user,pid,tty,stat,command
USER       PID TT       STAT COMMAND
root         1 ?        S    init [3]
root         2 ?        SW     [keventd]
root         3 ?        SW     [kapmd]
root         4 ?        SWN    [ksoftirqd_CPU0]
root         5 ?        SW     [kswapd]
root         6 ?        SW     [bdflush]
root         7 ?        SW     [kupdated]
root         8 ?        SW     [kreiserfsd]
root        26 ?        S      /sbin/devfsd /dev
root       858 ?        SW     [eth0]
root       860 ?        S      /sbin/dhcpcd -t 30 -h mandy eth0
bin        930 ?        S      /sbin/portmap
root       934 ?        SW     [rpciod]
root       935 ?        SW     [lockd]
root      1014 ?        S      /usr/lib/postfix/master
postfix   1038 ?        S        pickup -l -t fifo -u
postfix   1039 ?        S        qmgr -l -t fifo -u
root      1051 ?        S      /usr/sbin/sshd
root      1134 ?        S        /usr/sbin/sshd
se        1136 ?        S          /usr/sbin/sshd
se        1137 pts/0    S            -bash
root      1110 ?        S      /usr/sbin/syslogd -m 0
root      1113 ?        S      /usr/sbin/klogd -c 3 -2
root      1116 ?        S      /usr/sbin/cron
root      1128 tty1     S      /sbin/agetty 38400 tty1 linux
root      1129 tty2     S      /sbin/agetty 38400 tty2 linux
root      1130 tty3     S      /sbin/agetty 38400 tty3 linux
root      1131 tty4     S      /sbin/agetty 38400 tty4 linux
root      1132 tty5     S      /sbin/agetty 38400 tty5 linux
root      1133 tty6     S      /sbin/agetty 38400 tty6 linux
```

Wie man sieht, werden beim Start verschiedene Prozesse gestartet. Welche das sind, hängt von der Konfiguration und der Verwendung des Systems ab. Zum Beispiel wird hier noch der ssh Deamon (*sshd*) gestartet. Was aber immer vom System bereit gestellt wird, sind mehrere tty. Ein tty ist ein virtuelles Terminal auf dem man sich einloggen kann. Der Begriff tty (TeleTYpe) stammt aus den Anfängen der Computergeschichte, als Computer noch riesige Anlagen im Keller waren und man sich mittels eines Terminals (meist nur ein Bildschirm, eine Tastatur und eine serielle Schnittstelle zum Computer) einloggen musste. Heute geschieht dies lokal, das Prinzip ist aber dasselbe geblieben. In Abbildung 3.1 sieht man den Vorgang beim Start des Systems. Üblicherweise werden nicht wie in der Abbildung nur 3 tty gestartet, sondern 6 (tty1 bis tty6).

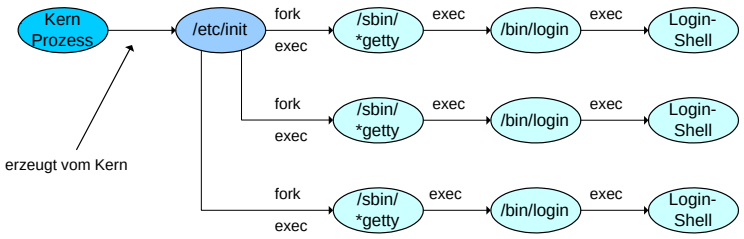

Abbildung 3.1: Prozessvergabelung beim Start des Systems.

Man sieht, dass der init Prozess `/sbin/*getty` startet, was zu einer Reihe von *getty Prozessen führt. Jeder *getty Prozess stellt ein tty zu Verfügung, das als erstes eine Begrüssungsmeldung auf den Bildschirm schreibt (`/etc/issue`) und anschliessend das login Programm (`/bin/login`) startet. Dabei entsteht kein neuer Prozess. Das login Programm fordert den Benutzer auf, seinen Benutzernamen und das Passwort einzugeben. Dieser Ablauf wird im Folgenden noch genauer erklärt. Sobald das erfolgreich abgeschlossen ist, wird eine shell gestartet, die dem Benutzer weitere Eingaben ermöglicht.

3.5.2 PAM - Pluggable Authentication Module

Was passiert nun genau, wenn das login Programm gestartet wurde und sich ein Benutzer anmeldet? Klassischerweise passiert folgendes:

- Das login Programm stellt durch Abfrage von Benutzernamen und Passwort sicher, dass der Benutzer überhaupt exisiert und auch derjenige ist, der er

vorgibt zu sein. Ferner werden im Erfolgsfalle dem Benutzer Gruppenrechte oder andere Privilegien gestattet.

- Es wird überprüft, ob das Benutzerkonto überhaupt gültig ist.
- Der Inhalt von `/etc/motd` wird auf dem Bildschirm ausgegeben. Danach wird ein Dienst zur Verfügung gestellt. Dies ist in den meisten Fällen eine shell, kann aber auch ein anderer Dienst sein.

Damit ist eigentlich alles in Ordnung und man fragt sich, warum kompliziertere Authentifizierungsmechanismen wie PAM überhaupt gebraucht werden.

Mit der klassichen Methode müssen die Authentisierungsmethoden in jedem Programm separat implementiert werden. Was macht man nun, wenn man die Anmeldung bei *allen* Diensten durch ein anderes Verfahren austauschen möchte? Man müsste das Anmeldeverfahren bei allen Programmen neu implementieren und diese neu installieren. PAM (Pluggable Authentication Module) dient nun dazu, diese Authentisierungsmethoden zu vereinheitlichen. Das heisst, anstatt irgendein Anmeldeverfahren in einen Dienst-Server einzubauen, gibt dieser Dienst-Server die Aufgabe an die PAM-Bibliothek weiter. Diese wiederum liest die Konfigurationsdatei des Dienstes, und entscheidet dann, welche Verfahren in welcher Reihenfolge für die Anmeldung des Benutzers angewendet werden. Die verschiedenen Verfahren sind in den Modulen implementiert, die in der Konfigurationsdatei angegeben werden müssen. Dabei wird an die aufrufende Anwendung nur das Gesamtergebnis zurückgegeben. Dieses Prinzip ist in Abbildung 3.2 illustriert. Damit kann ein Administrator für jedes Programm (vom Netzwerk-Serverdienst über das Login bis hin zum Passwortgeschützten Bildschirmschoner) festlegen, welches Modul in welcher Reihenfolge abgearbeitet wird und was für einen Einfluss Erfolg oder Misserfolg eines Moduls auf den Anmeldevorgang haben.

Es gibt für jedes Programm, dass PAM unterstützt im Verzeichnis `/etc/pam.d/` eine eigene Konfigurationsdatei, in der genau das Anmeldeverfahren für dieses Programm festgelegt werden kann. Hier ein Beispiel einer Konfigurationsdatei für das login Programm (`/etc/pam.d/login`):

```
#%PAM-1.0
auth     required   /lib/security/pam_securetty.so
auth     required   /lib/security/pam_stack.so service=system-auth
auth     required   /lib/security/pam_nologin.so
account  required   /lib/security/pam_stack.so service=system-auth
password required   /lib/security/pam_stack.so service=system-auth
session  required   /lib/security/pam_stack.so service=system-auth
session  optional   /lib/security/pam_console.so
```

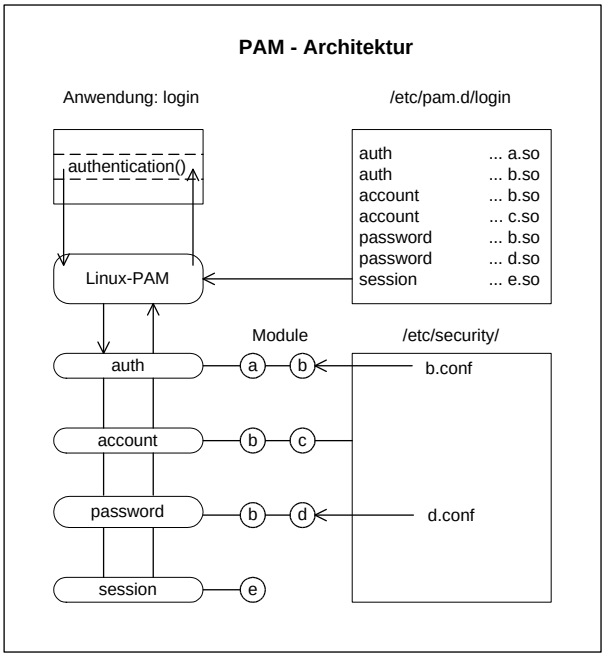

Abbildung 3.2: Architektur des Pluggable Authentication Module

Das Format ist immer wie folgt:

`PAM-Dienst Wichtigkeit Modul Argumente`

PAM-Dienst Der Dienst beschreibt, welche Aufgabe das Modul im Anmeldevorgang übernimmt.

auth übernimmt bei der Benutzeranmeldung zwei Aufgaben: Erstens wird der Benutzer überprüft und zweitens werden Gruppenrechte (unabhängig von `/etc/group`) oder andere Rechte vergeben.

account begrenzt den Zugang nach Folgenden Kriterien

- Uhrzeit
- verfügbare Systemressourcen (CPU, Speicher, etc.)
- Ort, von dem aus die Anmeldung statt findet.

session übernimmt Aufgaben, die bevor oder nachdem der Benutzer den Dienst in Anspruch nimmt, hat ausgeführt werden sollen (z.B. log-Datei schreiben, Variablen setzen, etc.).

password wird aufgerufen, wenn ein Benutzer seinen Zugangscode ändern will. Ein Zuganscode kann z.B. ein Passwort, ein Fingerabdruck oder Ähnliches sein.

Wichtigkeit bestimmt, was für eine Bedeutung Erfolg bzw. Misserfolg der Anmeldung haben.

required heisst, das Modul ist Notwendig. Schlägt dieses fehl, schlägt die gesamte Anmeldung fehl. Die folgenden Module werden aber dennoch abgearbeitet, um möglichst viel über einen potentiellen Angreifer zu erfahren und ihm möglichst wenig Anhaltspunkte über den Grund des Scheiterns zu geben.

requisite wie required, jedoch bricht der Anmeldevorgang sofort ab.

sufficient bedeutet, dass der Erfolg dieses Moduls für eine erfolgreiche Anmeldung ausreicht, falls vorher kein required-Modul fehlgeschlagen hat.

optional hat keinen Einfluss auf den Anmeldevorgang, ausser, wenn *nur* optional-Module in der Datei stehen.

Modul und Argumente beschreiben, welche Module für welche Aufgaben in welcher Reihenfolge verwendet werden.

Wie man sieht, ist PAM ein sehr mächtiges Werkzeug und ein zentraler Mechanismus der Sicherheit. Die gegebene Beschreibung ist eine Einführung. Es gibt sehr detaillierte Dokumentationen zu PAM. Diese können vom Web [8] heruntergeladen werden.

Im den vorhergehenden Abschnitten war immer die Rede vom *login* Programm. PAM ist aber keineswegs auf dieses Programm beschränkt. Wie schon erwähnt, kann PAM in Verbindung mit jedem Programm, das PAM unterstützt, verwendet werden.

Kapitel 4

Software

In diesem Kapitel wird der theoretische Hintergrund von Einstelleungen und Optionen von den Softwareprodukten erläutert, die für den Versuch benötigt werden. Dabei werden nur die Optionen erklärt, denen eine besondere Bedeutung im Zusammenhang mit den Übungen zukommt.

4.1 PitBull

PitBull eine Software von der Firma Argus Systems Group ermöglicht eine zusätzliche Sicherung des Systems. Herausragendster Vorteil der zusätzlichen Sicherung eines Systems durch PitBull ist der Schutz vor Sicherheitslöchern in installierten Softwarekomponenten. Zum Beispiel würde sie, wenn der Angreifer mit einer Bufferoverlow Attacke die Rechte des angegriffenen Dienstes erhält verhindern, dass er weiter ins System eindringen kann.

In den Übungsaufgaben kommt PitBull nicht zum Einsatz, da in der zur Verfügung stehenden Zeit nicht genügend aussagekräftige Beispiele zur Demonstration der Eigenheiten eines mit PitBull gesicherten Systems gefunden werden konnten. Mehr zu diesem Entscheid ist im Anhang A „Evaluation des Übungsablaufs“ auf Seite 52 zu lesen.

4.2 Rlogin

Mit rlogin ist es möglich sich bei einem System anzumelden (ähnlich wie mit telnet). Dazu ist es notwendig, dass auf dem System der Dienst rlogind gestartet wurde. Die Syntax für den Befehl lautet:

```
rlogin -l {Benutzername} {Hostname oder IP-Adresse}
```

Befindet sich auf dem System, auf welchem der Dienst gestartet im Verzeichnis eines Benutzers eine Datei namens `.rhosts`, so kann sich dieser Benutzer von den Systemen, die in dieser Datei eingetragen sind ohne Passwort anmelden. Möglicher Inhalt der Datei `.rhosts`:

```
192.168.0.1 master
192.168.0.3 master
```

Befindet sich diese Datei im `home`-Verzeichnis des Benutzers `master` kann dieser Benutzer von den IP-Adressen 192.168.0.1 und 192.168.0.3 ohne Passwort auf das System zugreifen. Das Deaktivieren oder Aktivieren dieses Dienstes geschieht unter Red Hat durch Editieren der Datei `\etc\xinetd.d\rlogin`. Dies ist ein Beispiel für den Inhalt der Datei:

```
# default: on
# description: rlogind is the server for the rlogin(1) program. \
# The server \
# provides a remote login facility with authentication based on \
# privileged port numbers from trusted hosts.
service login
{
socket_type = stream
wait = no
user = root
log_on_success += USERID
log_on_failure  += USERID
server = /usr/sbin/in.rlogind
disable = no
}
```

In der abgebildeten Konfigurationsdatei ist der `rlogin`-Dienst eingeschaltet. Dies ist zu sehen an der Zeile `disabled=no`. Wird die Zeile durch den Eintrag `disabled=yes` ersetzt, ist der Dienst ausgeschaltet.

4.3 Arp

Der Befehl `arp` erlaubt es den Arp-Cache anzuzeigen oder zu konfigurieren. Mit dem Parameter -a kann der Arp-Cache angezeigt werden. Mit der Syntax
`arp -s {IP-Adresse} {MAC-Adresse}`
kann für eine IP-Adresse eine statische MAC-Adresse gesetzt werden. Dies ist eine

Möglichkeit, um zu verhindern, dass dem System eine falsche MAC-Adresse untergeschoben wird.

4.4 Finger

Über den Befehl `finger` können Informationen über die Benutzer eines Systems abgefragt werden, falls auf dem System der entsprechende Dienst gestartet wurde. Um die momentan an einem System angemeldeten Benutzer anzuzeigen benutzt man den Befehl mit der Syntax `finger @{Hostname oder IP-Adresse}`. Um Detailinformationen über einen Benutzer des Systems zu erhalten kann vor dem @-Zeichen der Benutzer angegeben werden. Die Deaktivierung oder Aktivierung dieses Dienstes erfolgt analog zum Dienst `rlogin` aus dem Abschnitt 4.2. Die Konfigurationsdatei des `finger`-Dienstes lautet `\etc\xinetd.d\finger`. Ist der Dienst aktiviert sieht sie folgendermassen aus:

```
# default: on
# description: The finger server answers finger requests. Finger is \
# a protocol that allows remote users to see information such \
# as login name and last login time for local users.
service finger
{
socket_type = stream
wait = no
user = nobody
server = /usr/sbin/in.fingerd
disable = no
}
```

4.5 VMware

In diesem Kapitel geht es um ausgewählte Themen zur Benutzung und Installation eines VMware Systems. Im Besonderen werden die Themen genauer erläutert, die im Ramen des Einrichtens der Übungsumgebung besonderer Aufmerksamkeit bedürfen. Eine konkrete Anleitung zur Installation ist im Kapitel 5 auf Seite 37 zu finden.

4.5.1 Begriffe

Im Folgenden wird der Begriff *VMware Host* oder *Host System* verwendet werden. Damit wird das System bezeichnet auf welchem VMware installiert wird (Im Ramen

dieser Arbeit ist dies ein Linux System mit Red Hat 7.3).

Unter dem *virtuellen System* (auch *Virtual Machine*) wird das System verstanden, das innerhalb der VMware Umgebung läuft. Dem virtuellen System werden Hardwarekomponenten vorgespielt. Die VMware Umgebung leitet die Anfragen auf diese simulierten Hardwarekomponenten weiter auf reale Hardwarekomponenten des Host Systems. Es gibt verschiedene Möglichkeiten das Betriebssystem für das virtuelle System aufzusetzen. Entweder es existiert bereits eine reale Harddisk mit einem bereits vollfunktionsfähigen Betriebssystem oder es wird von VMware eine virtuelle Disk erzeugt, auf die, nach dem Booten des virtuellen Systems das gewünschte Betriebssystem installiert werden kann (Option *virtual disk*). Im ersten Fall ist nur der Ort an dem sich das bereits lauffähige Betriebssystem befindet anzugeben. Vollständigkeitshalber sollte hier erwähnt werden, dass es noch eine weitere Möglichkeit gibt ein virtuelles System mit einem Betriebssystem auszurüsten. Die Firma VMware stellt sogenannte VMware Guest OS Kit's zur Verfügung, dies sind für VMware vorkonfigurierte Betriebssysteme.

4.5.2 Diskmodi von Virtuellen Disks

Einem virtuellen System kann Speicherplatz in Form von virtuellen Disks zugeordnet werden. Für diese Disks existieren drei verschiedene Modi: *Persistent*, *Undoable* und *Nonpersistent*. Diese Modi bestimmen, wie die Daten der Disks gespeichert werden. Einem virtuellen System können mehrere virtuelle Disks zur Verfügung gestellt werden.

Persistent

Die virtuelle Disk verhält sich wie eine physische Harddisk: Daten werden nachdem sie geändert wurden sofort gespeichert. Der Zustand der Disks bleibt nach einem Neustart des virtuellen Systems erhalten.

Undoable

Änderungen auf der virtuellen Disk werden in einer Datei (mit der Namenserweiterung `.REDO`) protokolliert und können beim Herunterfahren des virtuellen Systems rückgängig oder definitiv gemacht werden.

Nonpersistent

Die Datei der virtuellen Disk wird nur gelesen. Änderungen der Daten werden wie bei *undoable* protokolliert. Die Änderungen auf der virtuellen Disk werden beim

Herunterfahren automatisch rückgängig gemacht.

4.5.3 Duplizieren eines virtuellen Systems

Hier wird davon ausgegangen, dass beim Erstellen des virtuellen Systems mit der Option *virtual disk* gearbeitet wurde (Abschnitt 4.5.1 auf Seite 28). Für jedes virtuelle System wird ein eigenes Verzeichnis erstellt. In diesem Verzeichnis befinden sich alle relevanten Daten. Im VMWare Handbuch [6] wird empfohlen vor dem Verändern der Dateien eines virtuellen Systems vom gesamten Verzeichnis ein Backup anzulegen. Um ein virtuelles System zu duplizieren braucht man bloss dieses Verzeichnis zu kopieren. Dazu muss das virtuelle System heruntergefahren werden. Falls zusätzliche Dateien ausserhalb des Verzeichnisses erstellt wurden, die das virtuelle System benutzt, müssen diese vom neuen Verzeichnis aus über die gleichen relativen Pfade erreichbar sein. Nun kann über den Menüpunkt `File > Open` die Konfigurationsdatei des kopierten virtuellen Systems geöffnet werden, womit das soeben kopierte System geladen wird.

Die Portierung eines virtuellen Systems von einem Linux Rechner auf einen Windows Rechner ist laut VMware Handbuch [6] auch möglich.

4.5.4 Netzwerkkonfiguration

Beim Einrichten eines neuen virtuellen Systems kann man zwischen drei verschiedenen Netzwerkintegrationen wählen: *Bridged*, *Host-only* oder *NAT*. Bei der Wahl eines Modus ist darauf zu achten, dass bei der Installation von VMware die Frage nach der Unterstützung des entsprechenden Modus bestätigt wurde. Sobald die Netzwerkunterstützung von VMware aktiviert wird, erstellt es virtuelle Switches. Diese sind im Host System als Netzwerkkomponenten zu erkennen. Es können bis zu zehn solche Switches erzeugt werden. Einige von ihnen tragen spezielle Bezeichnungen wie *Bridged* (auch VMnet0), *Host-only* (auch VMnet1) oder *NAT* (auch VMnet8). Diese speziellen Switches werden in den entsprechenden Netzwerkkonfigurationen verwendet.
Ein virtuelles System kann an einen beliebigen, virtuellen Switch angehängt werden. An einen virtuellen Switch können auch mehrere virtuelle Systeme angehängt werden. Im Folgenden werden die verschiedenen Netzwerkintegrations Modi erklärt.

Bridged Modus

Dieser Modus verbindet das virtuelle System mit dem LAN, an welchem das Host System angeschlossen ist. Falls das Host System über mehrere Netzwerkkarten verfügt, können für diese auch mehrere Bridges definiert werden.

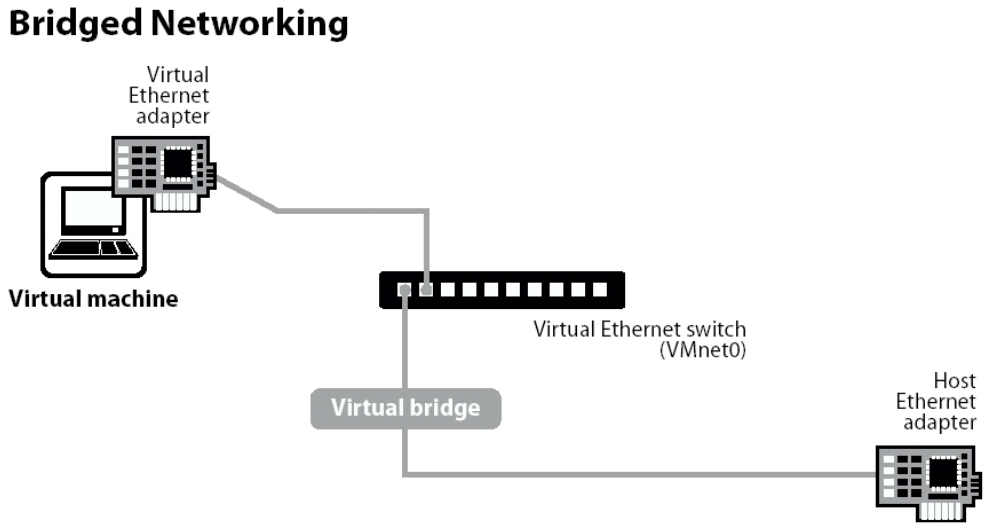

Abbildung 4.1: Verbindung des virtuellen Systems über den Ethernetadapter des Hosts mit dem Netzwerk im Bridged Modus

Abbildung 4.1 (entnommen aus dem VMware Handbuch [6]) zeigt die Komponenten, die in diesem Modus simuliert werden und wie sie mit dem realen Host verbunden sind. Die einzige reale Komponente, die auf der Abbildung zu sehen ist, ist der Ethernetadapter des Host Systems (bezeichnet mit „Host Ethernet adapter“). Ein virtueller Netzwerkadapter („Virtual Ethernet adapter“) und ein virtueller Switch („Virtual Ethernet switch“) werden von VMware simuliert. Logisch ist das Host System über den virtuellen Switch mit dem virtuellen System verbunden. Diese Verbindung ist in der Abbildung mit „Virtual bridge“ bezeichnet. Das System ist vergleichbar mit zwei realen Systemen, die an dem gleichen Switch angeschlossen sind. Andere Systeme, die mit dem Host System verbunden sind, sind logisch ebenfalls mit dem virtuellen Switch verbunden. Die Wahl der IP-Adressen ist unabhängig von der VMware-Umgebung. Die IP-Adressen können so gewählt werden, wie sie auch für reale Netzwerkkomponenten gewählt würden. Zum Beispiel könnte die IP-Adresse des virtuellen Systems 192.168.0.101 und die des Hostsystems 192.168.0.1 lauten, falls sich die Systeme im gleichen Subnetz befinden sollen.

Host-only Modus

In diesem Modus simuliert das virtuelle System dem Host System ein Interface mit der Bezeichnung Host-Only (VMware Linux) oder einen Ethernet Adapter (VMware Windows). Dieser Modus wird benutzt, falls das virtuelle System keine Verbindung zu einem realen Netzwerk, mit dem das Host System möglicherweise verbunden ist, aufweisen soll.

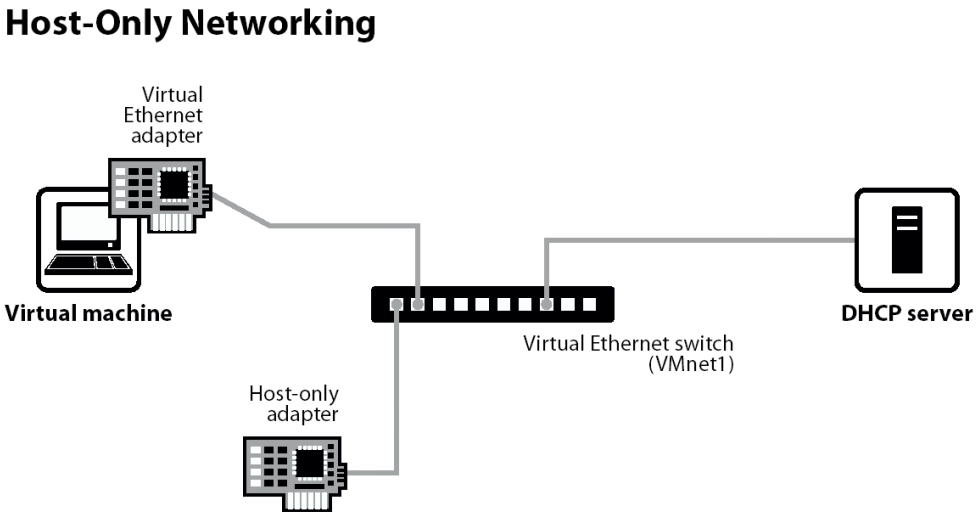

Abbildung 4.2: Im Host-only Modus simulierte Netzwerkkomponenten

Abbildung 4.2 zeigt die Komponenten, die in diesem Modus simuliert werden: Ein virtueller Netzwerkadapter („Virtual Ethernet adapter“), ein virtueller Switch („Virtual Ethernet switch“) und ein DHCP Server. Der DHCP Server ordnet dem virtuellen System automatisch eine IP-Adresse zu. Das virtuelle System erhält keinen Zugang zum Netzwerk das mit dem Ethernetadabter des Host Systems verbunden ist, ausser es wird auf dem Host System eine entsprechende Software installiert (zum Beispiel ein Proxy Server). Ein Proxy Server könnte dann die Anfragen des virtuellen Systems, die er über den virtuellen Netzwerkadapter („Host-only adapter“) erhält über den realen Hostadapter des Host Systems in ein angeschlossenes Netzwerk weiterleiten. Das bedeutet auch, dass andere Rechner der gleichen Broadcastdomain wie das Host System nichts von dem Virtuellen System sehen, falls kein Proxy installiert ist.

Network Address Translation (NAT) Modus

Die Installation des NAT Modus ist notwendig, falls für das virtuelle System keine eigene IP-Adresse zur Verfügung steht. Falls das Host System per *Wireless LAN* oder *Dial-Up Verbindung* mit einem anderen Netzwerk verbunden ist, stellt der NAT Modus eine Möglichgkeit dar, wie das virtuelle System Zugriff zum Netzwerk erhalten kann, mit welchem das Host System verbunden ist. Das virtuelle System ist über die gleiche IP-Adresse wie das Host System erreichbar.

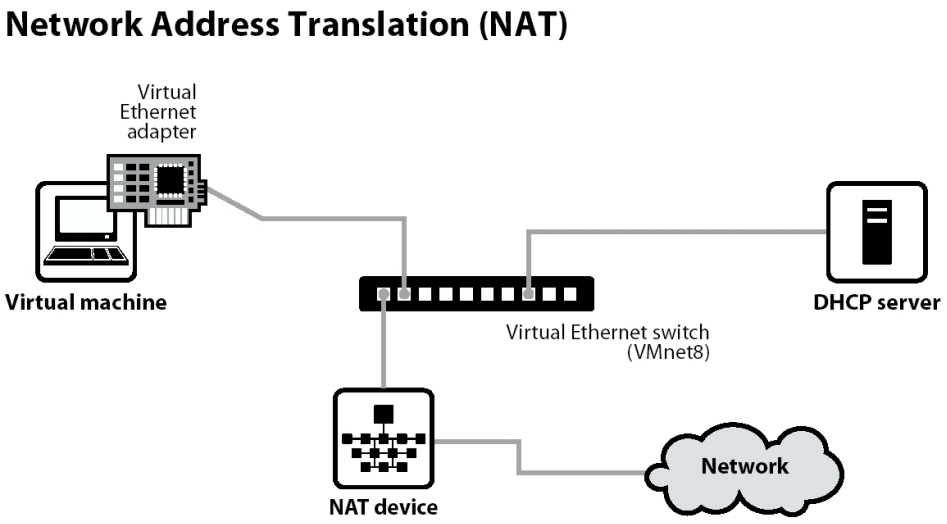

Abbildung 4.3: Verbindung des virtuellen Systems über den Ethernetadapter des Hosts mit dem Netzwerk im NAT Modus

In Abbildung 4.3 sind die simulierten Komponenten zu sehen: Ein virtueller Netzwerkadapter („Virtual Ethernet adapter“), ein virtueller Switch („Virtual Ethernet switch“), ein DHCP Server und ein sogenanntes *NAT Device*. Das reale Netzwerk, an dem auch das Host System angeschlossen ist, ist mit „Network“ gekennzeichnet. Logisch ist es direkt mit dem NAT Device verbunden. Das NAT Device ermöglicht es einem realen System, das an dem gleichen Netzwerk angeschlossen ist wie das Host System, mit dem virtuellen System zu kommunizieren. Die Angabe einer Portnummer ermöglicht es dem NAT Device zu entscheiden, ob mit der Adresse das Host System oder das virtuelle System gemeint ist. Der DHCP Server ordnet dem virtuellen System, auf dem nach aussen nicht sichtbaren Netz, automatisch eine IP-Adresse zu.

4.5.5 Tips und Tricks

Wechseln zwischen VMwarehost und virtuellem System

Läuft ein virtuelles System, erhält man per `Ctrl + Alt` die Kontrolle über das Hostsystem. Diese spezielle Tastenkombination ist nötig, weil alle anderen Eingaben von der virtuellen Konsole interpretiert werden. Um wieder das virtuelle System zu steuern braucht man nur in das Fenster der VMware Applikation zu klicken.

Installation eines virtuellen RedHat Systems

Bei der Installation von RedHat auf einem virtuellen System ist es empfehlenswert die Installation im textbasierten Modus durchzuführen. Die Bedienung der grafikbasierten Installation ist aus Gründen der Geschwindigkeit etwas mühsam.

Hinweis bezüglich VMware Tools Package

Beim Start des virtuellen Systems wird ein Hinweis angezeigt, dass *VMware Tools* installiert werden müsse. Dies ist für die Installation von virtuellen Servern jedoch nicht notwendig, da keine speziellen Grafikfunktionen verwendet werden. *VMware Tools* würde es beispielsweise ermöglichen eine höhere Auflösung als 640x480 mit 16 Farben zu verwenden.

CD Wechsel bei der Installation des virtuellen Servers

```
The CD-ROM drive failed to eject the disc with the error:
Device or resource busy.
```

Um dieses Problem zu beheben hilft es das CD-Rom Laufwerk ausserhalb von VMware zu unmounten und wieder zu mounten. Der Fehler und die Behebung konnte aber nicht repliziert werden.

4.5.6 Erfahrungen

Kopieren einer Virtual Disk

Das gesamte Verzeichnis eines virtuellen Systems wurde kopiert. Nach dem Starten des kopierten Systems sollte auf dem virtuellen System Pitbull isntalliert werden. Nach der Initierung des Installationsskriptes von Pitbull gab es eine Fehlermeldung begleitet von Steuerzeichen, die auf Probleme der Festplatte hinwiesen. Die Meldung wurde in der Konsole angezeigt, in welcher gerade das Installationsskript ausgeführt wurde. Da das Skript gerade auf eine Eingabe wartete ist es wahrscheinlich, dass das

Betriebssystem die Meldung erzeugte und dass es ein Problem mit dem Zugriff auf den virtuellen Disk gab. Merkürdigerweise schien dieser Zugriffsfehler das VMware Hostsystem beeinträchtigt zu haben.

Das Home-Verzeichnis des VMware Benutzers wies anschliessend ungültige Verzeichnis- und Dateinamen auf (siehe Abbildung 4.4).

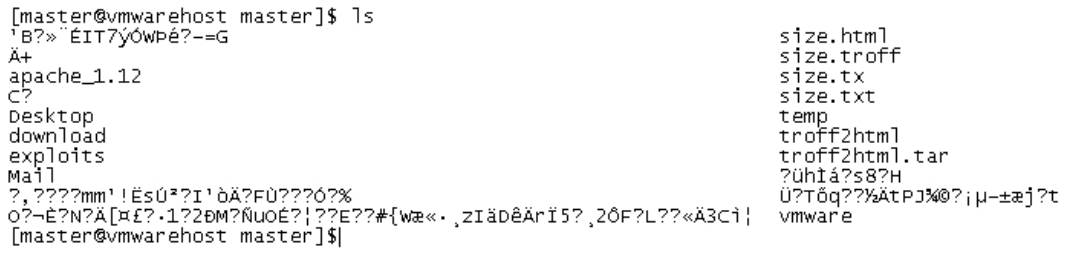

Abbildung 4.4: Ausgabe des ls-Commandos mit defekten Dateieinträgen

Das System war immer noch funktionstüchtig. Die erstellten Dateien haben keinerlei Inhalt und können ohne Beeinträchtigung des Systems gelöscht werden.

Teil III

Lab

Der folgende Teil enthält eine Beschreibung zur Installation der Übungsumgebung, sowie die Übungsaufgaben und deren Lösungen.

Kapitel 5

Installation der Übungsumgebung

Das folgende Kapitel beschreibt, wie die Übungsumgebung aufgebaut werden kann. Die schnellste Methode um eine lauffähige Übungsumgebung zu erhalten ist, das Installieren der Software VMware und dem Kopieren und Starten der virtuellen Disks (siehe „Dublizieren eines virtuellen Systems“ in Abschnitt 4.5.3 auf Seite 30). Auf den beiliegenden CDs befinden sich die Dateien der virtuellen Disks der Server 1 und 2 (siehe Anhang D).

5.1 VMware Hostsystem

Das VMware Hostsystem, ist das System auf dem VMware gestartet wird. Für die Installation des VMware Hostsystems wurde Red Hat 7.3 verwendet[1]. Es wurde eine Standard Installation eines Desktop-Systems mit Netzwerkzugang durchgeführt. Für die graphische Oberfläche wurde X-Windows mit dem Window Manager KDE, in den Versionen die mit diesem Release mitgeliefert werden installiert. Während der Installation wurden folgende Einstellungen in der angegebenen Reihenfolge vorgenommen. Bei nicht angegebenen Optionen wurde die Standardeinstellung beibehalten.

- textbasierte Installation durch die Eingabe von `text` wählen
- Sprache für den Installationsvorgang: Deutsch

[1] Da die Software VMware auch für das Betriebssystem Windows erhältlich ist, könnte für das VMware Hostsystem der Übungsumgebung auch ein System mit Windows benutzt werden.

- Tastaturtyp: sg-latin1
- Maustyp: Microsoft IntelliMouse (PS/2), ohne die Drei-Tastentyp-Emulation
- **Installationstyp: Desktop**
- Für die Partionierung wurde der *Disk Druid* gewählt.
- Initialisierung und Löschen des Disks bestätigen (es handelt sich hierbei, um die virtuelle Harddisk)
- **Partitionen: 47MB (ext3) /boot ,604 MB (swap), / (ext3) 5498MB** (hier kann die Grösse auf 1 MB eingestellt und die Option „Den gesamten verfügbaren Platz ausfüllen“ selektiert werden.
- Bootloader GRUB auswählen
- Bootloader im Masterboot record installieren lassen
- Keine speziellen Boot-Optionen eintragen
- kein GRUB Passwort verwenden
- Die Optionen „bootp“und „dhcp“ deaktivieren, IP-Adresse **(z.B. 192.168.0.2)** und Netzmaske (z.B. 255.255.255.0) eintragen[2]
- als Rechnername zum Beispiel „**vmwarehost**“
- keine Firewall selektieren
- „German Switzerland“ selektieren (auch als standard Sprache)
- Zeitzone Europa/Zürich
- Root **Passwort : hard2go**
- neuer User erstellt, **User: master**, **Passwort: behave**
- keine zusätzlichen Pakete selektiert

[2]Falls das Hostsystem mit den Virtuellen Servern kommunizieren möchte, muss seine IP-Adresse aus dem gleichen Subnetz, wie die IP-Adresse des virtuellen Servers(siehe Abschnitt 5.2 „Virtuelles System“) stammen. Ansonsten kann die IP-Adresse frei gewählt werden.

Der benötigte Speicherplatz für diese Installation beläuft sich auf 915MB. Ein vollständiges Protokoll der Installation ist unter `/root/install.log` zu finden. Erklärungen zu den einzelnen Optionen und weitere Installationsdetails können den Online Handbüchern von RedHat entnommen werden (Installation von Red Hat 7.3 [5] oder für die aktuelleste Version [4]).

5.2 Virtuelles System

Wie in der Evaluation des Übungsablaufs im Anhang A auf Seite 52 erläutert werden für die Übung mit VMware zwei Server simuliert: Server 1 und Server 2. In den folgenden Abschnitten wird erklärt, wie in VMware die zu simulierenden Hardwarekomponenten eingestellt werden und anschliessend die beiden Server innerhalb der VMware-Umgebung aufgesetzt werden.

5.2.1 VMware Konfiguration

Die Installation eines virtuellen Servers erfolgt nach dem Starten von VMware am einfachsten über den „Configuration Wizard“. Die einstellbaren Optionen sind gut beschrieben. Um ein neues Virtuelles System aufzusetzten sind folgende Einstellungen in der hier aufgezählten Reihenfolge zu machen.

- Create standard virtual machine
- Linux
- *Display name* Bsp. „Standard Server“
- „Full path of the virtual machine directory“
  Bsp. `/home/master/vmware/standard_server/`
- Create a new virtual disk
- „Virtual disk size (in megabytes)“ Bsp. 1300
- CD-ROM: Standard-Einstellungen belassen
- Floppy: Standard-Einstellungen belassen
- Bridged networking (*Bridged Modus* in Kapitel 4.5.4 auf Seite 30)

Bemerkung: Der Hauptspeicher wird standardmässig mit 32 MB eingerichtet. Falls mehr Hauptspeicher zur Verfügung steht, kann dies nach Beenden des „Wizards“ auf folgende Weise eingestellt werden:

1. Über `Settings>Configuration Editor` den Konfigurationseditor öffnen
2. In der Auflistung von Systemkomponenten „Memory“ auswählen
3. Den Wert von „Guest Size (MB)“ auf `128` einstellen

Nun ist die virtuelle Hardware des zu simulierenden Servers eingstellt. Jetzt kann auf dem virtuellen System ein Betriebssystem installiert werden.

5.2.2 Virtuelle Server

Um auf dem virtuellen System Linux zu installieren, muss die erste CD der Distribution eingelegt und das virtuelle System gestartet werden. Dies geschieht über die „Power On“ Schaltfläche. Das virtuelle System wird direkt von der realen CD gestartet. Alternativ könnte das Betriebssystem auch von ISO Image-Dateien der CDs erstellt werden (siehe VMWare Handbuch [6]).

Für die Installation der beiden virtuellen Server wurde Red Hat 7.3 verwendet. Der Installationsvorgang der beiden Server unterscheidet sich nur darin, das bei Server 1 während der Installation die Option „Shadow-Passwörter verwenden“ deaktiviert wird. Im folgenden wird in den einzelnen Schritten des Installationsvorgangs darauf hingewiesen, wenn für die Server eine individuelle Einstellung notwendig ist. Auf den Servern wird keine graphische Oberfläche installiert. Während der Installation werden folgende Einstellungen in der angegebenen Reihenfolge vorgenommen. Bei nicht angegebenen Optionen ist die Standardeinstellung beizubehalten.

- textbasierte Installation durch die Eingabe von `text` wählen
- Sprache für den Installationsvorgang: Deutsch
- Tastaturtyp: sg-latin1
- Maustyp: Microsoft IntelliMouse (PS/2), ohne die Drei-Tastentyp-Emulation
- Server 1: **Installationstyp: Benutzerdefinierte Installation**[3]
  Server 2: **Installationstyp: Server**
- Für die Partionierung wurde der *Disk Druid* gewählt.
- Initialisierung und Löschen des Disks bestätigen (es handelt sich hierbei, um die virtuelle Harddisk)

[3] Standardmässig würde sonst Password-Shadowing installiert werden.

- **Partitionen: 47MB (ext3) /boot ,200 MB (swap), / (ext3) 1052MB** (hier kann die Grösse auf 1 MB eingestellt und die Option „Den gesamten verfügbaren Platz ausfüllen“ selektiert werden.
- Bootloader GRUB auswählen
- Bootloader im Masterboot record installieren lassen
- Keine speziellen Boot-Optionen eintragen
- kein GRUB Passwort verwenden
- Die Optionen „bootp“und „dhcp“ deaktivieren[4]
  Server 1: **IP-Adresse 192.168.0.100**[5]
  Server 2: **IP-Adresse 192.168.0.200**
  Netzmaske 255.255.255.0 eintragen
- als Rechnername zum Beispiel „server1“, beziehungsweise „server2“
- keine Firewall selektieren
- „German Switzerland“ selektieren (auch als standard Sprache)
- Zeitzone Europa/Zürich
- Root **Passwort : hard2go**
- neuer User erstellt, **User: master**, **Passwort: behave**
- Konfigurationansicht der Authentifizierung (betrifft nur Server 1): Die Option „Shadow-Passwörter verwenden“ deaktivieren
- alle Pakete deselektiert, Web-Server selektiert

Diese Installation benötigt 940 MB Speicherplatz auf dem virtuellen System. Ein vollständiges Protokoll der Installierten Pakete wird später unter `/root/install.log` zu finden sein. Die Ausgabe des Befehls `ifconfig` auf dem virtuellen System liefert uns die folgende Informationen[6]:

[4] Die IP-Adresse wird manuell gesetzt, darum wird der DHCP-Dienst nicht benötigt.

[5] Der Versuch ist nicht abhängig von den gewählten IP-Adressen. Diese IP-Adressen sind als Vorschläge zu verstehen.

[6] Falls die Ausgabe nicht den Erwartungen entspricht, können die Netzwerkeinstellungen zum Beispiel mit dem Konfigurationsprogramm `/usr/bin/setup` editiert werden

```
eth0      Protokoll:Ethernet  Hardware Adresse 00:50:56:40:00:46
          inet Adresse:192.168.0.100  Bcast:192.168.0.255
          Maske:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:100
          RX bytes:1425 (1.3 Kb)  TX bytes:1354 (1.3 Kb)
          Interrupt:10 Basisadresse:0x10a0

lo        Protokoll:Lokale Schleife
          inet Adresse:127.0.0.1  Maske:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:0
          RX bytes:560 (560.0 b)  TX bytes:560 (560.0 b)
```

Abbildung 5.1: Ausgabe des Befehls ifconfig auf dem virtuellen System

In der Abbildung 5.1 ist zu erkennen, dass das virtuelle System eine IP-Adresse und eine MAC-Adresse erhalten hat. Die Ausgabe ist nicht zu unterscheiden von einem realen System mit einer realen Netzwerkkarte. Nun kann auf einem anderen Rechner des Uebungsnetzes in einem Browser die IP-Adresse (hier 192.168.0.100) eingegeben werden und es wird eine Standard Webseite des Apache-Webservers angezeigt.

Auf dem Server 1 werden nach der Installation die Dienste `rlogin` und `finger` aktiviert. Wie diese Dienste aktiviert werden entnehmen sie bitte dem Abschnitt 4.2 zum Thema `rlogin` auf Seite 26 und dem Abschnitt 4.4 zum Thema `finger` auf Seite 28.

Auf Server 2 sind die Dienste `finger` und `rlogin` deaktiviert. Das Bootmanagerpasswort ist eingschaltet und lautet: **nashorn**. Wie der Bootmanager GRUB konfiguriert werden muss, um das Bootmanagerpasswort zu aktivieren ist in den Lösungen zu den Übungen in Abschnnitt 6.2 auf Seite 45.

Kapitel 6

Übungen

Ziel der Übung ist es, die Prinzipien des Härtens eines Systems zu verstehen und anhand verschiedener Tests praktisch „erleben“ zu können. Der konkrete Übungsaufbau und -ablauf ist im Anhang A im Abschnitt A.3 beschrieben.

In diesem Kapitel sind Wissensfragen und praktische Übungen zusammengestellt. Sie könnten dazu dienen, in einer Übung verwendet zu werden. Das Kapitel ist unterteilt in Übungsaufgaben 6.1 und Lösungen 6.2.

6.1 Aufgaben

1. (a) Es gibt eine einfache Methode, um sich in ein Linux System als root einzuloggen, ohne dass man ein Passwort hat. Versuchen Sie herauszufinden, wie es funktioniert und probieren Sie Ihre Lösung aus.
   Hinweis: Der Bootmanager *Grub* kann im „single user mode“ gestartet werden.

   (b) Versuchen Sie das Problem von zu lösen, indem sie ein Bootmanager Passwort setzen.
   Hinweis: `info grub`

2. Finden Sie mit dem `ps` Befehl heraus, welche Prozesse gerade am Laufen sind. Welcher Prozess startet die shell (hier bash)?

3. Sie wollen, dass sich kein Benutzer ausser root am System anmelden kann, ohne den Rechner im „single user mode“ aufzustarten. Was tun Sie?

4. Sie wollen eine Datei `/etc/loginuser` erstellen, so dass sich nur noch die in dieser Datei aufgelisteten Benutzer am System per *login* anmelden können.

   (a) Konfigurieren Sie PAM so, dass dies möglich wird.
   Hinweis: Verwenden Sie das Modul `pam_listfile.so`

   (b) Testen Sie die Konfiguration mit einem bestimmten Benutzer, der ein gültiges Passwort besitzt (allenfalls einen neuen Benutzer anlegen) und in der Datei eingetragen ist. Versuchen Sie auch, sich unter einem gültigen Benutzer einzuloggen, der *nicht* in der obigen Datei aufgelistet ist.

   (c) Was passiert, wenn Sie sich mit einem gesperrten Benutzer (nicht in der Datei eingetragen) grafisch anmelden und warum?

5. Was versteht man unter dem Begriff „Härten eines Systems“?

6. Was ist der Vorteil eines hardened Servers?

7. Was kann zusätzlich zum Härten gemacht werden, damit die Sicherheit weiter erhöht wird?

8. Was für Produkte gibt es, die das Härten automatisch übernehmen?

9. Wie ist ein sicheres Passwort aufgebaut und wie kann es schon bei der Erstellung überprüft werden?

10. Welche Dateirechte gibt es unter Linux und wie können sie verändert werden?

11. Was bedeutet PAM?

12. In welcher Datei kann man unter PAM das Verhalten von `login` konfigurieren?

13. Mittels ARP-Spoofing kann eine Man-in-the-Middle-Attacke durchgeführt werden. Es soll zu einem Rechner eine FTP-Verbindung hergestellt werden. Die MAC-Adresse des FTP-Servers sei bekannt. Wie kann man verhindern, dass sich jemand dazwischen schaltet? (Hinweis: Den Befehl `arp` verwenden)

14. Mittels `rlogin` ist es möglich zu einem Rechner eine Verbindung aufzubauen, ohne das ein Passwort angegeben werden muss.

   (a) Welche Einstellungen müssen dazu auf dem Server gemacht werden? (Bemerkung: Um Netzwerkdienste mit geänderte Einstellungen neu zu starten muss der Befehl `/etc/init.d/xinetd restart` eingegeben werden.)

   (b) Wie lautet die Syntax des Befehls auf dem Client, um die Verbindung herzustellen?

(c) Auf dem Server wurde die IP-Adresse 192.168.0.50 als *trusted* konfiguriert. Wie melden sie sich mit diesem Wissen ohne Passwort beim Server an?

15. (a) Das Standard-Server verwendet kein Passwort-Shadowing. Wie ist der Inhalt der Datei `\etc\passwd` zu interpretieren?

    (b) Das gesicherte System verwendet Passwort-Shadowing. Vergleichen sie den Inhalt der Datei `\etc\passwd` mit der, des Standard-Servers. Wo sind die Passworte gespeichert?

16. Auf dem Standard-Server ist ein Dienst Namens `finger` installiert.

    (a) Versuchen sie von einem anderen Linux-Rechner aus mit dem Befehl `finger` Informationen über den Benutzer `master` auf dem Standard-Server herauszufinden.

    (b) Wie können Sie diesen Dienst auf dem Standard-Server ausschalten?

6.2 Lösungen

1. (a) Starten Sie das System neu und wählen sie den gewünschten Eintrag im Bootmenu an. Drücken Sie anschliessen `e` um den String zu editieren. Hängen Sie dem String folgendes an:

   ```
   console=tty0 single
   ```

   Damit sieht ein kompletter boot Eintrag z.B. wie folgt aus:

   ```
   kernel /boot/vmlinuz-2.4.20 ro root=/dev/hda1 console=tty0 single
   ```

   Drücken Sie anschliessend `Enter` um zu Bestätigen, gefolgt von `b` um das geänderte Menu zu booten. Wie sie sehen, haben Sie ohne sich einzuloggen root Berechtigung auf dem System.

   Bemerkung: Das Bootmenu wird dabei nur temporär geändert. Das heisst, beim nächsten Start sind wieder die Originaleinstellungen vorhanden.

   (b) Setzen Sie ein Bootmanager Passwort. Damit kann verhindert werden, dass der String, der dem Kernel übergeben wird, verändert werden kann. Es kann sogar verhindert werden, dass ein Bootmenu-Eintrag ohne ein Passwort gestartet werden kann. Das folgende Beispiel erläutert wieder den Vorgang beim *Grub* Bootmanager.

   Das folgende ist ein Auszug aus `/boot/grub/menu.lst` , einer Konfigurationsdatei von *Grub*:

```
#Globale Optionen
default=0
timeout=10

#Menu Eintrag
title Red Hat Linux (2.4.18-3)
  root(hd0,0)
  kernel /vmlinuz-2.4.18-3 ro root=/dev/hda2
  initrd /initrd-2.4.18-3.img
```

Diese Datei kann nun um einen Passwort Eintrag erweitert werden. Dazu ist folgender Eintrag in der globalen Sektion zu machen:

```
password --md5 <verschlüsseltes Passwort>
```

Um das gewünschte Passwort in verschlüsselter Form zu erhalten, kann das Programm *grub-md5-crypt* verwendet werden. Dieses Programm fragt nach dem Passwort und gibt dieses in verschlüsselter Form auf der Konsole aus. Dieses wird anschliessend in die obige Konfigurationsdatei eingefügt.
Soll zusätzlich das Starten des System ohne Passwort verhindert werden, kann das mit der Eingabe von `lock` am Ende des Menu Eintrages erreicht werden. Falls mehrere Menu Einträge vorhanden sind, muss dies für jeden Eintrage eingegeben werden.
Die Konfigurationsdatei könnte mit den vorgenommenen Änderungen z.B. wie folgt aussehen:

```
#Globale Optionen
default=0
timeout=10
password --md5 $1$6sUzf/$6jYUb70dd.ZaYNxOEv7le1

#Menu Eintrag
title Red Hat Linux (2.4.18-3)
  root(hd0,0)
  kernel /vmlinuz-2.4.18-3 ro root=/dev/hda2
  initrd /initrd-2.4.18-3.img
  lock
```

Die Änderungen werden nach dem Speichern der Datei aktiv. Beim nächsten Start muss das Passwort eingegeben werden, um das System starten zu können.

2. Verwenden Sie folgenden Befehl: `ps ax -H -o user,pid,tty,stat,command`. Der Ablauf ist folgender: *init longrightarrow login longrightarrow bash*. Falls Sie sich grafisch angemeldet haben, sieht die Lösung verschieden aus (je nach Windowmanager).

3. Legen Sie die Datei `/etc/nologin` an. Sie können diese Datei noch mit einer Meldung füllen, die dem Benutzer angezeigt wird.
   Lösung von `http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam-6.html#ss6.17`

4. (a) i. `echo test >> /etc/loginusers`
   ii. Editieren von `/etc/pam.d/login`. Einfügen folgender Zeile:
   `auth required /lib/security/pam_listfile.so onerr=succeed item=user sense=allow file=/etc/loginuser`

   (b) Die Benutzer, die in `/etc/loginusers` eingetragen sind, können sich anmelden, die restlichen nicht mehr.

   (c) Die grafische Anmeldung funktioniert noch, da nur `/etc/pam.d/login` verändert wurde. `/etc/pam.d/gdm` (bzw `/etc/pam.d/xdm`, je nachdem, welcher grafische login-Manager verwendet wurde) blieb unverändert.

   Lösung von `http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam-6.html#ss6.13`

5. Das System wird so konfiguriert und eingestellt, dass er nur die Dienste anbietet, die wirklich benötigt werden und nur die Benutzer eingerichtet sind, die tatsächlich ein Login benötigen. Alle weiteren Einstellungen und Pakete werden deaktiviert, bzw. deinstalliert. Die aktivierten Dienste werden so konfiguriert, dass sie maximale Sicherheit bieten.

6. Das Sicherheitsrisiko, dass sich jemand unrechtmässig Zugang verschafft wird verringert. Durch das Entfernen nicht benötigter Dienste gibt es auch weniger potentielle Fehlerfälle.

7. Möglichkeiten sind (nicht abschliessend):

   - Firewall
   - Inrusion Detection System
   - Intrusion Prevention

8. *harden*, *Bastille Linux*

9. Ein sicheres Passwort ist mind. 6 Zeichen lang, hat mind. 2 alphabetische Zeichen und 1 numerisches oder 1 spezial Zeichen. Das Passwort muss vom user login verschieden sein und darf auch nicht davon abgeleitet werden können. Es darf zudem nicht von einem Wort oder einer Wortkombination abgeleitet werden können.

   Passwörter können schon beim erstellen überprüft werden. Dazu verwenden Sie das PAM-Modul *pam_cracklib* in Verbindung mit *cracklib* und *cracklib_dict*

10. Read (lesen), Write (schreiben) und Executable (ausführen). Sie können mit dem Befehl `chmod` verändert werden (`man chmod`).

11. Pluggable Authentication Module

12. `/etc/pam.d/login`

13. Mit folgender Syntax `arp -s {IP-Adresse} {MAC-Adresse}`, wird die MAC-Adresse fix einer IP-Adresse zugeteilt. Für die Adressen sind die Werte des FTP-Servers einzugeben. Somit ist es einem Angreiffer nicht mehr möglich auf diese IP-Adresse ein ARP-Spoofing durchzuführen.

14. (a) In der Datei `.rhost` muss die IP-Adresse des Systems eingetragen werden, dem vertraut werden soll.
    (b) `rlogin -m {user} {Host}`
    (c) IP-Adresse ändern auf 192.168.0.50 und sich mit dem `rlogin` mit der Syntax aus der letzten Teilaufgabe einloggen.

15. (a) user hasch usw.
    (b) Die Hashwerte der Passworte sind nicht mehr in der Datei `\etc\passwd` sondern in einer anderen Datei Namens `\etc\shadow` gespeichert.

16. (a) `finger master@{IP-Adresse des Standard Servers}`
    (b) Befindet sich diese Datei beispielsweise im `home`-Verzeichnis des Benutzers `bravo` kann dieser Benutzer von den IP-Adressen 192.168.0.1 und 192.168.0.3 ohne Passwort auf das System zugreifen. Das Deaktivieren des Dienstes geschieht durch Editieren der Datei `\etc\xinetd.d\finger` (Siehe Abschnitt 4.4). In der Konfigurationsdatei ist die Zeile `disabled=no`. zu ersetzten durch `disabled = yes`. Hiermit wird der Dienst deaktiviert, sobald das Netzwerk mit dem Befehl `ixinetd.d restart` neu gestartet wurde. Dieser Befehl kann nur mit root-Rechten ausgeführt werden.

Teil IV

Schlussfolgerungen

Dieser Teil umfasst ein Fazit und einen Ausblick der Arbeit.

Kapitel 7

Schlussfolgerungen

7.1 Fazit

Es ist schwierig, Härten eines Systems einzugrenzen und festzulegen, was alles dazugehört. Diese Arbeit zeigt verschiedene Aspekte auf, die beim Härten eines Systems beachtet werden sollten. Welche Aspekte dabei wichtiger bzw. weniger wichtig sind, hängt von der Verwendung des Systems ab und muss von Fall zu Fall neu entschieden werden. Trotzdem ist es gelungen, einen Überblick zu geben, der erlaubt, die Prinzipien zu verstehen und auch das Gefühl dafür gibt, was mit Härten alles erreicht werden kann. Da es immer neue Programme gibt, wird es auch immer neue Anforderungen an die Sicherheit geben, wenn man aber verstanden hat, worauf man achten sollte, wird man keine grosse Schwierigkeiten haben, neue Sicherheitsanforderungen zu verstehen und umzusetzen.

Die erarbeitete Übung bietet eine Möglichkeit, das gewonnene Wissen in der Praxis umzusetzen. Dabei wurde darauf geachtet, dass die Übung nicht aus Fragen besteht, die einfach aus einem Text abgeschrieben und somit schnell beantwortet werden können, sondern, Fragen zu stellen, zu deren Beantwortung das Verständnis für das entsprechende Gebiet vorhanden sein muss. Damit wurde versucht, die Aufgaben möglichst spannend zu gestalten. Sie geben darüber hinaus Aufschluss über Sicherheitsrisiken, deren Vorhandensein kaum jemandem bewusst ist, sich aber dennoch viele aussetzen. Damit soll ein „Aha-Effekt“ entstehen.

7.2 Ausblick

Die in dieser Arbeit erzielten Resultate entsprechen nur einem Ausschnitt aus dem Gebiet der Sicherheit. Wie immer, ist damit nur die „halbe Wahrheit“ gesprochen

bzw. geschrieben worden. Es gibt noch sehr viele Richtungen, aus der die Thematik Sicherheit angegangen werden kann. Wollte man die „ganze Wahrheit“ wissen, wäre dies schier ein Ding der Unmöglichkeit. Wobei dazu noch zu sagen ist, dass dies auch nicht unbedingt nötig ist. Sicherheit ist ein dehnbarer Begriff, der je nach Anwendung, je nach persönlicher Ansicht und je nach Risiko unterschiedlich ausgelegt werden kann. Nebst dem Härten eines Systems gibt es noch weitere Gebiete, die ebenso wichtig sind. Namentlich sind das Intrusion Prevention, Intrusion Detection, Firewalling und andere. Jedes dieser Gebiete würde für sich eine eigene Arbeit ergeben. Damit soll nur gezeigt werden, dass diese Arbeit einen Ausschnitt der grossen Welt der Sicherheit bietet und nicht als Vorlage genommen werden kann, ein System komplett sicher zu machen.

Ebenso wie das Härten eines Systems nur ein Ausschnitt ist, sind die in dieser Arbeit beschriebenen Vorgehensweisen und Vorschläge nur ein Ausschnitt aus allen Möglichkeiten, die das Härten bietet. Die Auswahl erfolgte, nach Studium aller gefundenen Möglichkeiten, nach der Wichtigkeit aus Sicht der Autoren. Man könnte sich durchaus vorstellen, die Arbeit in diesem Sinne zu erweitern und noch mehr Möglichkeiten des Härtens aufzuzeigen. Vor allem aber ist es wichtig, sich selbst auf einem aktuellen Stand zu halten, da sich im Bereich der Sicherheit sehr viel in kurzer Zeit ändert.

Die Übung lässt sich fast beliebig ausbauen. Das Ziel war bei der Erstellung, mögliche Fragen und deren Antworten bereitzustellen. Aufbauend auf diesen Fragen können neue, andere Fragen entstehen, die je nach Ziel und Anforderungen der Übung verschieden ausfallen können.

Anhang A

Evaluation des Übungsablaufs

A.1 Idee

Das Prinzip des Härtens soll dem Student anhand eines direkten Vergleichs der Eigenschaften und des Verhaltens eines gehärteten und eines nicht gehärteten Servers gezeigt werden. Um verschiedene Ansätze des Härtens zu zeigen wird ein weiterer Vergleich mit einem Server durchgeführt, der mit *Argus Pitbull* gesichert wurde. Es werden drei Server mit verschiedenen Konfigurationen verglichen:

- Ein Standard Server ohne speziell angepasste Konfiguration
- Ein manuell gehärteter Server
- Ein Server auf dem Argus Pitbull installiert wurde

A.2 Übung Variante 1

A.2.1 Uebungsaufbau

In einer ersten Variante waren verschiedene Attacken geplant, um die Auswirkungen der unterschiedlichen Konfigurationen der Server zu zeigen.

Tabelle A.1: Varianten der Serverinstallationen für die Übung

Kategorie	**Attacke**	**Server 1**	**Server 2**	**Server 3**
OS	Passwörter cracken	shadow Passwort ausgeschaltet	shadow Passwort eingeschaltet	Auf diesem Server ist PitBull installiert. Sonst entspricht die Installation derjenigen von Server 1.
	ARP-Attacke	default Einstellungen	statisches ARP Cache, log Einträge, arpwatch	
	Trojanisches Pferd	-	Einschränkungen des Benutzers, Ausführbarkeit im System einschränken	
Applikation	Attacke gegen ssh1	ssh als Dienst verwenden und Erklärungen dazu machen	ssh1 updaten auf ssh2, oder sicher konfigurieren	
	ftp (buffer overflow)	default Einstellungen	ftp Dienst sicherer konfigurieren	
	Analysetools	Ungenutzte Dienste aktiviert (Bsp. Finger)	Ungenutzte Dienste deaktivieren	

In der Tabelle A.1 sind die, für die einzelnen Server typischen Konfigurationsdetails aufgelistet. In der ersten Spalte ist die Kategorie, beziehungsweise der speziell zu härtende Bereich aufgeführt. Die Kategorie „OS“ bezeichnet Sicherheitsaspekte, die mit Betriebssystemeinstellungen zu tun haben. Die Kategorie „Applikation“ umfasst Sicherheitsaspekte gewisser Dienste oder den Einsatz von Analysetools zur Gewinnung von Informationen über ein System. Die Spalte „Attacken“ sind mögliche Angriffe auf einen der Server 1 bis 3, um die Auswirkungen der unterschiedlichen Konfigurationen zu demonstrieren.

Für den Übungsablauf waren verschiedene Varianten denkbar. Im Folgenden werden die Varianten beschrieben und die Gründe, warum welche Variante gewählt wurde, erläutert.

A.2.2 Übungsverlauf Variante 1

Auf einem Rechner werden innerhalb der VMware Umgebung drei Server simuliert. Die Studenten versuchen mittels Tools und bekannten Sicherheitsschwächen Unterschiede der Zustände und Konfigurationen herauszufinden.

A.2.3 Übungsverlauf Variante 2

Die Klasse wird in Gruppen zu jeweils vier Studenten aufgeteilt. Jede Gruppe arbeitet an zwei Computern. Zwei der Studenten (hier als Gruppe A bezeichnet) werden sich in einem ersten Durchgang mit dem Härten eines Servers befassen. Die anderen zwei Studenten (hier als Gruppe B bezeichnet) machen sich mit möglichen Angriffen vertraut. Anschliessend wird Gruppe B Angriffe auf den nicht gehärteten, den gehärteten, wie auch den mit Pitbull gesicherten Server durchführen. Gruppe A wird die Auswirkungen der Angriffe überprüfen.

Wie in Abbiludng A.2 zu sehen ist werden die Phasen 1 und 2 werden zweimal durchgeführt. Wobei im zweiten Durchgang die Gruppen ihre Rollen tauschen. Damit die Gruppen im zweiten Durchgang neue Aktionen durchführen können, ist der Fokus der Durchgänge auf verschiedene Themen des Härtens gerichtet. Im dargestellten Beispielablauf wurden die Themen *OS Hardening* und *Application Hardening* gewählt.

Tabelle A.2: Aufteilung der Gruppen und deren Aufgaben

		Gruppe A	**Gruppe B**
1. Durchgang *OS Hardening*	Phase 1	Härten eines Servers	Studieren von Beispielattacken
	Phase 2	Überwachen und interpretieren der Auswirkungen	Attackieren eines nicht gesicherten Standard Servers und des von Gruppe A konfigurierten Servers
2. Durchgang *Application Hardening*	Phase 1	Studieren von Beispielattacken	Härten eines Servers
	Phase 2	Attackieren eines nicht gesicherten Standard Servers und des von Gruppe B konfigurierten Servers	Überwachen und interpretieren der Auswirkungen

A.2.4 Entscheid

Variante 2 ist vom Lernaspekt und der technischen Realisierung her der Variante 1 vorzuziehen. Sie bietet sowohl die Möglichkeit das Prinzip des Härtens aus der Sicht eines Administrators zu sehen, als auch deren Auswirkungen aus der Sicht eines Angreifers. Zusammengefasst bietet Variante 2 folgende Qualitäten und Vorteile gegenüber Variante 1:

- Gruppendynamik durch Interaktion einer Zweiergruppe mit einer anderen Zweiergruppe
- Ansporn durch eine Gruppe als „Gegenspieler“
- Keine Serverüberlastung, da jede Vierergruppe nur mit zwei Computern interagiert
- Auswirkungen der einzelnen Attacken können besser überwacht werden, da jede Zweiergruppe ein anderes System angreift
- Auswirkungen des Härtens eines Systems sind konkreter ersichtlich

A.3 Übung Variante 2

A.3.1 Übungsaufbau

Tabelle A.3: Variante 2 der Serverinstallationen für die Übung

Kategorie	**Server 1**	**Server 2**
OS	Shadow-Passwörter ausgeschaltet	Shadow-Passwörter eingeschaltet
	kein Bootmanager Passwort	Bootmanager Passwort eingeschaltet
Applikation	Dienst `rlogin` aktiviert	Dienst `rlogin` deaktiviert
	Dienst `finger` aktiviert	Dienst `finger` deaktiviert
	Webserver installiert[1]	Webserver installiert

A.3.2 Übungsverlauf

Die Übung kann einzelnen oder zu zweit durchgeführt werden. Im Wesentlichen geht es darum, Dienste und Betriebssystemeinstellungen zu konfigurieren. Dabei werden die Aufgaben aus dem Kapitel 6 auf Seite 43 durchgearbeitet. In Abbildung A.1 ist die Übungsumgebung dargestellt. Während der Übung wird Server 1 konfiguriert. Um die Einstellungen zu vergleichen, können beide virtuellen Systeme (Server 1 und 2) nebeneinander in zwei VMware Instanzen ausgeführt werden. Durch das Aufrufen der Dienste des Server 1 vom Server 2 aus können die Konfigurationen überprüft werden. Falls zu zweit gearbeitet wird, kann auch von einem Rechner auf den anderen eine Verbindung hergestellt werden, um zum Beispiel das passwortfreie Einloggen mittels `rlogin` auszuprobieren. Grundsätzlich gibt es viel Spielraum in den Übungsaufgaben bezüglich dem Testen der Konfigurationen. Dies soll die Experimentierfreude anspornen und somit den Lerneffekt erhöhen.

[1] Der Webserver wird im Übungsverlauf nicht explizit genutzt, ist aber bereits installiert, falls die Übung erweitert werden sollte.

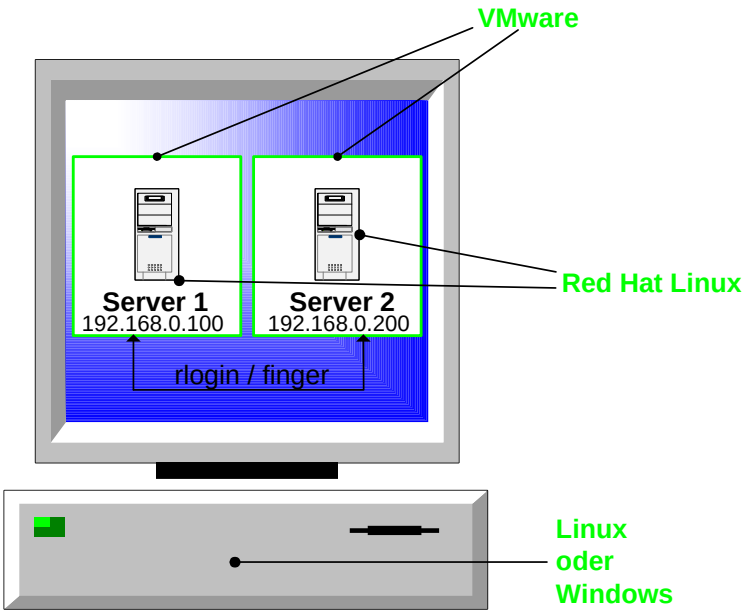

Abbildung A.1: Uebungsumgebung

A.4 Entscheid

Die Übungsvariante 1 ist davon abhängig, wieviele erfolgreiche Attacken inszeniert werden können. Solche funktionierenden Attacken für ein bestimmtes System zu finden ist sehr aufwendig. Attacken selber zu programmieren überstieg das zur Verfügung stehende Zeitbudget bei weitem (siehe Anhang B auf Seite 58). Die Übungsvariante 2 bietet wie Variante 1 die Möglichkeit Techniken des Härtens anzuwenden. Anstatt Attacken werden nun nicht deaktivierte, sicherheitskritische Dienste zum Aufzeigen von Systemschwächen genutzt.

Anhang B

Evaluation einer Attacke

Um den Vorteil eines mit PitBull gesicherten Servers am Besten zeigen zu können, ist eine der mächtigsten Attacken auf einen Server nötig. Dazu gehört das direkte erlangen von Root-Rechten über einen Remote Zugang. Dies bedeutet, dass von einem, über das Netzwerk erreichbaren Rechner jemand die Möglichkeit erlangt, auf dem angegriffenen System beliebige Befehle auszuführen. Einem solchen Angriff entspricht die *Bufferovwerflow Attacke* (genauer erklärt im folgenden Abschnitt B.1).

Als Erstes werden neue Sicherheitslücken auf dem Internet publiziert. Darum wurde für die Suche nach Informationen über eine solche Attacke vor allem dieses Medium verwendet.

B.1 Bufferoverflow Attacke

Diese Attacke nutzt Schwachstellen in der Implementation von Programmen aus, denen Parameter oder andere zu verarbeitende Daten übergeben werden können. Die Schwäche besteht darin, dass bei der Verarbeitung von übergebenen Daten, diese Daten in den Hauptspeicher kopiert werden, ohne dass die Grösse des dafür reservierten Speicherbereichs überprüft wird. Wird Beispielsweise als Parameter ein String von einer gewissen Länge erwartet, doch ein weitaus längerer String übergeben, wird trotzdem der ganze String in den Speicher geschrieben. Dabei werden andere Daten, die sich ebenfalls im Speicher befinden überschrieben. Falls sich das Programm in diesem Moment gerade innerhalb einer Subroutine befindet, befindet sich auch die Rücksprungadresse ins Hauptprogramm in diesem Speicherbereich. Ziel der Attacke ist es, diese Rücksprungadresse zu überschreiben und dabei eine Adresse zu setzten, die in den Bereich des Speichers springt, an dem sich der übergebenen String befindet. Gelingt dies, wird der Prozessor nach dem durcharbeiten der Subrouti-

ne versuchen, den übergebenen String als Programmcode zu interpretieren. Falls sich in dem String also Maschinencode in der Form von Binärwerten (sogenannter *Payload*) befindet, der über einen Methodenaufruf des Betriebssystes eine Shell ausführt, erhält der fremde Rechner Zugang zu einer Shell mit den Benutzerrechten des fehlerhaften Programmes.

B.2 Programmieren der Attacke

Um eine solche Attacke durchzuführen wurden zuerst Hintergrundinformationen über das Prinzip der Attacke gesucht. Als beste Quellen erwiesen sich hierfür zur Einführung in die Thematik der Artikel „Das Sicherheitsloch“ aus dem Magazin c't [7] und zur Vertiefung der Artikel „Smashing the Stack for Fun“ aus dem Phrack Magazin [1].

Durch das Studium der Prinzipien wurde schnell klar, dass das Erstellen eines eigenen Exploits dieser Art dem Umfang einer ganzen Studienarbeit entsprechen würde (Details dazu sind in den oben genannten Quellen zu finden):

- Analysieren von Quellcode eines Dienstes auf verletzliche Speicherzugriffe, beziehungsweise Aufrufe von bekannten Funktionen, die beim Speicherzugriff keine Bereichsprüfung der Variable durchführen

- Zusammenstellen des sogenannten Payloads der Attacke (dies sind die notwendigen Befehle in Maschinencode, um einen Betriebssystemaufruf zu tätigen, der beispielsweise eine Shell öffnet)

- Das anzugreifende Programm in einem Debugger ausführen, um eine bestimmte Speicheradresse (im Stack) zur Laufzeit des Programmes herauszufinden

- Ein Programm schreiben, das die Verbindung aufbaut und dem Dienst die nötigen Daten übermittelt

B.3 Entscheid

Um eine solche Attacke selber zu programmieren stand nicht genügend Zeit zur Verfügung. Aus diesem Grund wurde entschieden, auf dem Internet nach bereits vorhandenen Attacken für den Standard Server der Übungsinstallation aus Abschnitt 5.2.2 zu suchen.

B.4 Suche nach einem Exploit

Zuerst wird der Verlauf der Suche beschrieben und die genutzten Informationsquellen genannt. Anschliessend werden weitere Anregungen für eine Suche genannt.

Bei der Suche nach einem Exploit wurden verschiedene Strategien verfolgt. Zuerst wurde eine sehr eingegrenzte Suche durchgeführt, anschliessend wurden die Kriterien des Exploits immmer mehr gelockert. Grob könnte die Suche und in folgende Phasen eingeteilt werden:

1. Suche nach Exploits für die installierten Softwarepakete des aufgesetzten Standard Servers mit Red Hat 7.3
2. Suche nach Exploits für ältere Versionen der installierten Softwarepakete
3. Suche nach Exploits einer beliebigen lauffähigen Software unter Red Hat 7.3

Im Folgenden sind die Ergebnisse der einzelnen Phasen zusammengefasst. Eine Auswahl der gefunden Scripte sind auf der beigelegten CD zu finden (D auf Seite 67).:

1. Es wurden zwei Skripte für den *Apache Server* gefunden, die aber nicht die erwarteten Reaktionen zeigten, bzw. nicht funktionierten.
2. Für eine ältere Version des *Apache Servers* wurden verschiedene Scripte für die gleiche Schwachstelle gefunden. Die Apacheversion auf dem Server wurde durch eine ältere Version ersetzt. Dies galt auch für alle abhängigen Pakete. Dennoch erwies sich kein Script als tauglich.
3. Es wurde auf verschiedenen Webseiten auf eine Schwachstelle einer älteren Version des FTP-Serverprogrammes *ProFTP* hingewiesen. Nach der Installation des Programmes wurde ein FTP-Exploit-Script ausgeführt, jedoch ohne Erfolg.

B.5 Informationskanäle

Eine Auswahl der besuchten Internetseiten, die nach Informationen durchsucht wurden:

online.securityfocus.com Sehr Umfangreiche Datenbank bezüglich Computer Sicherheit

packetstormsecurity.nl Sehr umfangreiche Datenbank mit diversen Exploit-Beschreibungen inklusive Quellcode. Das Motto der Organisation *Packet Storm* ist „Know your enemy“.

www.phrack.org Das Hacker und Phreak Magazin überhaupt. Es bietetet neue kreative Ideen für Hacking Ansätze.

www.cultdeadcow.com Seite einer Hackergruppe. Sie bieten ein paar wenige Tools und News an.

www.atstake.com Eine Netzwerksicherheit-Beratungs Firma in Amerika. Sie bietet verschiedene ausgeklügelte Netzwerksicherheit-Tools an. Unter anderem auch solche, die Applikationen auf potentielle Bufferoverflow-Fehler prüfen.

Alternativ befragten wir Mitglieder der LUGS (Linux User Group) ein Linux-Verein an der Hochschule Rapperswil, die uns einzelne der oben genannten Webseiten verwiesen.

B.6 Ungenutzte Informationskanäle

Der Internet Relay Chat (IRC) bietet eine intensiv genutzte Plattform zum Informations- und Programmaustausch. Unter anderen auch zu Themen die sich in Grauzonen der Legalität befinden. Um diese Quellen allerdings gezielt und effizient nutzen zu können, muss man sich während einer gewissen Zeit (Schätzungsweise ein bis zwei Monaten) in den verschiedenen Chaträumen und auf den verschiedenen Servern „bewegen“, beziehungsweise Kontakte knüpfen und Informationen sammeln. Über diese Vorarbeit erhält man Wissen über nicht offizielle und versteckte Chaträume, wie auch Aufenthalt und Commandos für sogenannte Bots[1]. Die auf IRC gehandelten Informationen sind oft brisanter, als die auf Webseiten publizierten, da Benutzer miteinander kommunizieren können, ohne dass sie ihre IP-Adressen preisgeben müssen[2].

Die Suche nach Informationen in verschiedenen Newsgroups wäre ebenfalls eine Alternative, benötigt aber Erfahrungen bezüglich der Qualität der einzelnen Gruppen.

Das Anmelden bei Mailinglisten zu den entpsrechenden Themen könnte sich auch als hilfreich erweisen bei der Suche nach Exploits, kann sich aber durch das Verarbeiten von ganzen Mailfluten als sehr zeitaufwendig entpuppen.

[1] Bots sind Dienste, die sich wie reale Chatpersonen in Chaträumen aufhalten und auf eingegebenen Text reagieren (Zum Beispiel Scripte ausführen oder Dateitransfers einleiten).

[2] IRC Server versenden die Nachrichten an die bei ihnen angemeldeten Clients.

B.7 Fazit der Exploit-Suche

Die Suche nach einem Exploit, der für eine spezifische Systemkonfiguration geschrieben wurde, ist sehr aufwendig. Es gibt verschiedene Variablen der Systemkonfiguration, die mit dem Exploit übereinstimmen müssen. Dazu gehören die Version des Betriebssystems (oft auch die Version des Kernels) und die Version des anzugreifenden Dienstes. Beim *Apache Server* sind auch die Versionen der installierten Module relevant. Bei einem Angriff auf den *Apache Server* gibt dies bereits mindestens vier Variablen die mit dem Exploit übereinstimmen müssen. Falls nur eine Handvoll Exploits für den Apache gefunden werden, ist es unwahrscheinlich, dass einer der Exploits funktioniert. Zudem ist der Sourcecode von gefundenen Exploits meistens von sehr schlechter Qualität. Unter schlechter Qualität wird folgendes verstanden:

- Schlecht bis gar nicht kommentiert (geschweige den Dokumentiert)
- kaum aussagekräftige Fehlermeldungen oder gar keine Fehlermeldungen implementiert
- fehlerhafter Code (Typendeklaration falsch, Methodenaufrufe mit falscher Anzahl Argumenten, usw.)
- Variablen- und Methodennamen nicht in Englisch (gefunden wurden Scripte in Spanisch)

Dies hat verschiedene Gründe: Der Code war nicht für die Weitergabe gedacht, nur für den Eigenbedarfentwickelt und/oder Codeteile wurden zusammenkopiert aus „in the wild“ entdeckten, bösartigen Programmen (z.B. aus einem Wurm). Um ein gefundenes Script benutzen zu können, sind oft folgende Massnahmen erforderlich:

- Fehler finden und korrigieren
- Programmlogik genau analysieren und gegebenfalls anpassen
- in Machinencode integrierter Payload (Siehe Abschnitt B.1 „Bufferoverlow Attacke“) für das anzugreifende System selber schreiben, kompilieren und in das Script einfügen, damit die Betriebssystemversion sicher stimmt.

Es folgt ein Auszug aus der Datei `apache-scalp.c` mit einem Kommentar der Entwickler zur Illustration des Aufwandes die mit einem funktionierenden Exploit-Script verbunden sind.

```
* The "experts" have already concurred that this bug...
*      -       Can not be exploited on 32-bit *nix variants
```

```
*      -       Is only exploitable on win32 platforms
*      -       Is only exploitable on certain 64-bit systems
*
* However, contrary to what ISS would have you believe, we have
* successfully exploited this hole on the following operating systems:
*
*      Sun Solaris 6-8 (sparc/x86)
*      FreeBSD 4.3-4.5 (x86)
*      OpenBSD 2.6-3.1 (x86)
*      Linux (GNU) 2.4 (x86)
*
* Don't get discouraged too quickly in your own research. It took us
* close to two months to be able to exploit each of the above operating
* systems. There is a peculiarity to be found for each operating system
* that makes the exploitation possible.
```

Dieser Kommentar sagt nicht nur aus, dass es Monate dauern kann bis ein spezieller Exploit funktioniert, sondern sich Sicherheitsexperten oft selber nicht einig sind, für welche Systeme ein Exploit jetzt funktioniert oder nicht. Es scheint schwieriger zu sein für ein konkretes System einen funktionierenden Exploit zu finden, als mit einem Exploit-Script auf dem ganzen Internet ein angreifbares System.

B.8 Entscheid

Die Exploit-Suche zeigte sich trotz verschiedenen Ansätzen und Nachforschungen ohne Erfolg. Da ein solcher Exploit die Basis für die Übung mit PitBull darstellte, wurde der Übungsablauf überarbeit. Der neue Übungsablauf war nun, die Hardening Theorie anzuwenden und nicht mehr drei verschieden Server zu vergleichen (Siehe A.3 auf Seite 56).

Anhang C

Hardware Evaluation

Im Verlauf der Arbeit standen nacheinander zwei verschiedene Server mit unterschiedlich leistungsfähigen Komponenten für Testzwecke zur Verfügung (im Folgenden mit „Alpha Lab“ und „Beta Lab“ bezeichnet). Auf beiden Rechnern wurde Red Hat 7.3 und VMware Workstation Version 3.2 installiert. Im Folgenden werden die Hardwarekonfigurationen der beiden Systeme und das Verhalten der Systeme während dem Einrichten und den ersten Funktionstests geschildert, um einen Eindruck über die Anforderungen zu vermitteln.

C.1 Alpha Lab

Das zur Verfügung stehende Hostsystem hat folgende Hardwaredaten:

- 3.2 GB Festplatte
- Hauptspeicher 128 MB
- Intel Pentium II, 266 MHz
- ATI Mach64, 4 MB RAM
- Microsoft Maus (PS/2), zwei Knöpfe
- SVGA fähiger Bildschirm
- Standard 105 Tasten Keyboard

Es konnten maximal zwei virtuelle Systeme gleichzeitig gestartet werden. Pro Systemen stand maximal 32 MB Hauptspeicher zur Verfügung. Die Anzahl parallel

aufgesetzter Systeme wurde auch durch die zur Verfügung stehende Festplattenkapazität stark eingeschränkt. Pro Server sollte mindestens 1 GB zur Verfügung stehen. Davon wird 1 GB bereits vom Hostsystem selbst verwendet. Die Systemreaktionen der virtuellen Systeme waren sehr träge. Die Konfigurationen aus den Konsolen der simulierten Server waren noch gut möglich. Der Bildaufbau dauerte teilweise ein bis zwei Sekunden. Bei Webzugriffen von einem beliebigen System im Uebungsnetz war nicht ersichtlich, dass es sich nur um einen simulierten Server handelte. Um zwei Webserver, ohne zusätzliche Dienste für eine Übung zu simulieren, würden diese Hardwarevoraussetzungen reichen. Das System wäre aber sehr verletzlich gegnüber einer *Denial of Service Attacke*.

C.2 Beta Lab

Das zur Verfügung stehende Hostsystem hat folgende Hardwaredaten:

- 6.4 GB Festplatte
- Hauptspeicher 512 MB SDRAM
- Intel Pentium III, 550 MHz
- ATI Mach64, 16 MB RAM
- Microsoft Maus (PS/2), zwei Knöpfe
- SVGA fähiger Bildschirm
- Standard 105 Tasten Keyboard

Das starten von drei virtuellen Systemen gleichzeitig stellt kein Problem dar. Jedem virtuellen System stehen 128 MB Hauptspeicher zur Verfügung. Das Arbeiten mit den virtuellen Systemen war ohne grössere, bemerkbare Geschwindigekitseinbussen möglich.

C.3 Erkenntnis

Die Systemvoraussetzungen des Beta Labs waren gegnüber dem Alpha Lab wesentlich besser. Die Grösse der Harddisk reicht zwar aus, lässt aber für die Serverinstallationen nicht allzuviel Spielraum für das Installieren von zusätzlichen Paketen. Es stehen pro System (dem Host System inklusive) 1.6 GB Plattenspeicher zur Verfügung.

Um möglichen Speicherplatzproblemen vorzubeugen, sollte das Host System mindestens 8 GB Festplattenspeicher besitzen, damit den drei virtuellen Systemen je 2 GB zur Verfügung stehen. Ansonsten genügen die Hardwarevoraussetzungen des Beta Labs den Übungsanforderungen.

Anhang D

Beiliegende CDs

Dem Bericht liegen zwei CDs bei. Auf einer CD befindet sich der Bericht, Skripte und Informationen bezüglich dem Projekt, auf der anderen befinden sich die in VMware ausführbaren konfigurierten Server 1 und 2. Dieser Anhang beschreibt den Inhalt der beigelegten CDs.

D.1 CD 1

- Bericht in LaTeX
- Bericht im PDF-Format
- Diagramme aus dem Bericht
- Exploitcode Beispiele
- Webseite des Projekts (inkl. Zeitauswertung und Projektplan)
- VMware Desktop 3.2 für Linux (inkl. Lizenz)
- VMware Desktop 3.2 für Windows (ohne Lizenz)

D.2 CD 2

- VMware Dateien des Server 1
- VMware Dateien des Server 2

Anhang E

Projektmanagement Dokumente

E.1 Zeitauswertung

Dieser Auszug aus der Datenbank zeigt die Arbeitszeiten geordnet nach Person und Bereich.

Michael Egli			162.00
Dokumentation	92.00		
Einarbeiten	32.00		
IT Arbeitsumgebung	15.00		
IT Projektbezogen	1.00		
Meeting	11.25		
Projektmanagement	10.75		
René Herrmann			155.75
Dokumentation	63.50		
Einarbeiten	26.25		
IT Arbeitsumgebung	18.50		
IT Projektbezogen	24.75		
Meeting	10.25		
Projektmanagement	12.50		
		Projekt Total:	317.75

E.2 Projektplan

Der Projektplan befindet sich auf der folgenden Seite.

Anhang F

Glossar

A

ARP Address Resolution Protocol. Das Protokoll verbindet die IP-Adresse mit der physikalischen MAC-Adresse der jeweiligen Ethernet-Karte.

Authentifizierung Bezeugen der Echtheit

Authentisierung Beglaubigung, Rechtsgültigmachung

D

Dienst Ein Programm, dass Funktionen oder Ressourcen zur Verfügung stellt (Beispielsweise *sshd*).

H

Härten eines Systems Vorgehen, um ein System sicherer zu machen.

Host System System auf welchem VMware installiert ist.

M

MAC Media Access Control. Dieser Layer bietet die Möglichkeit des individuellen Zugriffs bezüglich der verwendeten Netzarchitektur.

N

Non-Repudiation Eindeutigkeit des Ursprungs.

O

OS Hardening Prozess um die Sicherheit eines IT-Systems zu erhöhen.

P

PAM Pluggable Authentication Module. PAM dient der einheitlichen Authentifizierung der Benutzer durch verschieden Programme.

partitionieren Die Festplatte in verschiedene Bereiche aufteilen, die untereinander unabhängig sind.

V

Virtuelles System Ein innerhalb von VMware installiertes System, das einen kompletten Rechner simuliert. Die Hardwarekonfiguration des Rechners sind simulierte VMware Komponenten.

Literaturverzeichnis

[1] Smashing the Stuck for fun. Website: http://www.phrack.org/show.php?p=498&a=14.

[2] Kevin Fenzi. Linux Security-HOWTO. Website: http://tldp.org/HOWTO/Security-HOWTO/.

[3] Argus System Group. Website: http://www.argus-systems.com.

[4] Red Hat Inc. Red Hat Manuals. Website: http://www.redhat.com/docs/manuals/.

[5] Red Hat Inc. The Official Red Hat Linux x86 Installation Guide, RedHat Release 7.3. Website: http://ftp.ind.net/pub/mirrors/ftp.redhat.com/redhat/linux/7.3/de/doc/RH-DOCS/rhl-ig-x86-en-7.3/index.html.

[6] VMware Inc. Vmware manual (online verfügbar). Website: http://www.vmware.com/support/.

[7] Stephan Kallnik, Daniel Pape, Daniel Schröter, and Stefan Strobel. Das Sicherheitsloch, c't 23/2001,s.216. Website: http://www.heise.de/ct/01/23/216/.

[8] Andrew G. Morgan. PAM Dokumentation. Website: http://www.kernel.org/pub/linux/libs/pam/.

[9] Alexander Reelsen and Javier Fernández-Sanguino Peña. Securing Debian Manual. Website: http://www.linuxsecurity.com/docs/harden-doc/html/secuting-debian-howto/.

[10] Securitystats. Website: http://www.securitystats.com/.

[11] Andrew S. Tanenbaum. *Modern Operating Systems*. Prentice Hall, NJ 07458, 2 edition, 2001. ISBN 0-13-031358-0. 951 pp.

Index