Studienprogramm Sachbearbeiter:in Digitale Forensik
Praktikum Digitale Forensik Grundlagen
Dozenten: Tim Wetterau | Leander Hossfeld
wetterau@hs-mittweida.de | hossfeld@hs-mittweida.de
Stand: 25.03.2024

HOCHSCHULE MITTWEIDA
University of Applied Sciences

Digitale Forensik

Teil 1 – Grundlagen Linux

In diesem Praktikum lernen Sie den grundlegenden Umgang mit dem offenen Betriebssystem Linux. Als Beispiel ziehen wir eine aktuelle Version von Linux Mint mit der Oberfläche Cinnamon heran, an dem die gezeigten Beispiele getestet werden sollen. Zu dessen Download sowie Installation wird auf die online verfügbare Installationsanleitung mit Oracle VirtualBox verwiesen. Bei Fragen und Problemen sprechen Sie uns bitte direkt in den Praktikumssitzungen an, um schnelle Hilfe zu erhalten.

Gegenstand dieses Praktikums sind folgende Punkte:

- Nutzung des Terminals
- Nutzer- und Gruppenverwaltung
- Paketierung
- Arbeiten mit dem Dateisystem

Kurze Erläuterung syntaktischer Notationen:

- <...> sind durch konkrete Angaben zu ersetzen (mit < und > ersetzen)
- [...] optionale Angaben bei einem Befehl (mit [und] ersetzen)
- $ Anfang einer Befehlszeile im Terminal
- # Kommentare (werden von Bash nicht interpretiert und müssen nicht eingegeben werden
- *kursiv* sind alle Aufgaben geschrieben, die von Ihnen zu lösen sind

Vorbereitung

Zur Durchführung des Praktikums ist vorausgesetzt, dass die durch die Praktikumsanleiter zur Verfügung gestellte virtuelle Maschine Linux-Mint-BKA oder eine andere funktionierende Linux-Distribution auf ihrem Rechner erfolgreich installiert ist.

Hinweis für Fortgeschrittene und erfahrene Linux-Nutzer:

➔ Sollte Sie sich in der Lage fühlen schon komplexere Aufgaben im Bereich der Shellprogrammierung umzusetzen, dann probieren Sie sich gern direkt an Praktikum 4 – Shell-Programmierung

➔ Wenn Sie diese Aufgaben problemlos lösen konnten, können Sie gern das Praktikum vorzeitig verlassen
  - Dennoch empfehlen wir das Praktikum 3 – Praxis-Hashing durchzuarbeiten
  - Dort werden Ihnen praktische Inhalte vermittelt, die im forensischen Bereich Anwendung finden

➔ Ansonsten arbeiten Sie bitte ruhig die Praktika mit Ihren Komillitonen durch und stellen Sie bei Unklarheiten Fragen

Nutzung des Terminals

Im Gegensatz zu anderen Betriebssystemen basieren viele Linux-Installationen auf reinen Textkommandozeilen (Konsole) und bieten dem Anwender keine graphische Oberfläche (GUI – engl. Graphical User Interface). Anwendung findet das oft im Cloud- und Serverbereich, wobei durch das Einsparen der GUI mehr Rechenleistung für die Serverdienste zur Verfügung steht. Ebenso wird per Remotezugriff (SSH – Secure Shell) rein textuell auf die entfernte Maschine zugegriffen. Aufgrund dessen ist die Arbeit mit einer Linux-Shell essenziell im Umgang mit dem Betriebssystem.

Zu Beginn starten Sie Oracle VirtualBox, welches Sie bereits installiert haben und fahren die Linux-Mint-BKA VM hoch. Melden Sie sich mit folgenden Zugangsdaten an:

- **Nutzername:** praktikus
- **Passwort:** praktikus-0815

Nach einer kurzen Dauer sehen Sie einen Willkommens-Bildschirm. Sie können sich gern kurz die Zeit nehmen und das System auf ihre Bedürfnisse anpassen, dies ist aber nicht notwendig für die zu erledigenden Aufgaben.

Abbildung 1: Terminal-Symbol in der Taskleiste von Linux Mint

Starten Sie nun ihr (vielleicht) erstes Terminal. Dazu können Sie das Terminal-Icon in der Taskleiste der VM auswählen oder die Tastenkombination **Strg + Alt + T** drücken. Sie sehen nun einen Command-Prompt:

Abbildung 2: Command-Prompt im Terminal

Dieser Command-Prompt verlangt immer eine Eingabe des Nutzers. Das System wartet somit auf Befehle. In Abbildung 2 sehen Sie in blau die erste Besonderheit in Linux. „**~**“ steht immer für das Home-Verzeichnis des Nutzers. Dieses ist standardmäßig unter dem Pfad „**/home/<username>**“ zu finden. In Ihrem Fall „**/home/praktikus**“. *Zur Überprüfung der Angabe geben Sie doch Ihren ersten Befehl nach dem Command-Prompt ein:*

```
$ pwd
```

pwd ist die Abkürzung für „**Print Working Directory**“ (zeige akuelles Arbeitsverzeichnis), also das Verzeichnis, in dem Sie sich gerade befinden. Durch die Bestätigung mit Enter/Eingabe erhalten Sie das aktuelle Arbeitsverzeichnis in der nächsten Zeile als Ausgabe.

Die Arbeit in der Kommandozeile beruht ausschließlich auf Kommandos/Befehlen. Diese haben meist einen ähnlichen Aufbau. Dieser soll durch folgenden (etwas komplexeren) Befehl dargelegt werden:

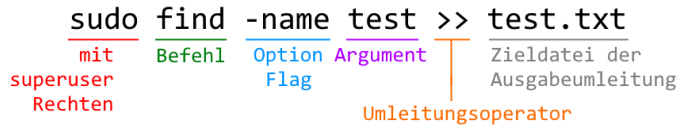

Abbildung 3: beispielhafter Befehl im Terminal

Die meisten Befehle können Optionen erhalten. Diese sind meist mit einem vorangestellten Minus (**-**) gekennzeichnet und einem folgenden Buchstaben. Zusätzlich verlangen wiederrum viele Optionen zusätzliche Argumente. Im Beispiel in Abbildung 3 nutzen wir **find**. **find** kann eine Datei im aktuellen Verzeichnis rekursiv finden anhand des Namens. Dazu geben wir zu find die Option „**-name**“ an. Als Argument erhält die Option „**test**“. Folglich suchen wir mit dem Befehl die Datei mit dem Namen test. Anstatt dieses nun auf der Kommandozeile wieder auszugeben, leiten wir die Ausgabe in eine Datei um. Dabei ist es nicht wichtig, ob diese bereits besteht. Mit „**>>**“ leiten wir die Ausgabe in die Datei und hängen sie an den bereits bestehenden Inhalt an. Die Datei für die Ausgabe ist im Beispiel „**test.txt**“.

Woher weiß man denn nun aber, welche Optionen und Argumente von einem Befehl unterstützt werden? Dafür gibt es auch einen Befehl! Mit „**man <befehl>**“ kann man das „Manual“, also die Bedienungsanleitung eines Befehls aufrufen. *Versuchen Sie das einmal und geben sie den folgenden Befehl ein:*

```
$ man echo
```

Sie sehen nun die Bedienungsanleitung einer der einfachsten Befehle in Linux. Sie können sich hier bewegen mit den Pfeiltasten nach oben und unten oder mit den Bildlauftasten. Brauchen Sie mehr Hilfe drücken Sie **h**. *Lesen Sie sich die man-Page von echo ruhig einmal durch und machen sie sich ein klein wenig vertraut* mit dem Aufbau des Befehls und dessen Möglichkeiten. *Welche Optionen unterstützt der Befehl? Brauchen sie weitere Argumente?* Wenn Sie damit fertig drücken Sie **q**, um die Man-Page zu schließen. *Als erste Aufgabe versuchen Sie nun die Ausgabe „**`Hello World`**“ im Terminal zu erzeugen!*

```
$ echo hello world
```

Das Terminal bietet sich für die reguläre Arbeit mit Dateien, als auch zum Automatisieren von repetitiven Arbeitsschritten an. Wir wollen zu diesen Komplexen nun in den kommenden Abschnitten einen kleinen Einstieg finden.

Arbeiten mit dem Dateisystem

Die Arbeit mit dem Dateisystem ist in Linux per Console vielleicht etwas gewöhnungsbedürftig, aber wenn man sie verstanden hat, wird sie schnell intuitiv und eindeutig. Dazu bekommen Sie zuerst einige Grundlagen. Das native Dateisystem in Linux ist ext3 oder ext4. Allerdings kann Linux mit fast allen anderen Dateisystemen ebenso interagieren.

Linux-Dateisysteme sind in einer Baumstruktur aufgebaut, wie in Abbildung 4 dargstellt. Dabei ist das Wurzelverzeichnis, also das höchste im Baum, das Verzeichnis „/“.

Tabelle 1: Kürzel im Dateisystem

Kürzel	Bedeutung
/	Wurzelverzeichnis (root)
.	aktuelles Verzeichnis
..	nächstes übergeordnetes Verzeichnis
~	Home-Verzeichnis des aktuellen Nutzers

Pfadangaben können entweder absolut oder relativ erfolgen. Absolute Pfadangaben beginnen immer mit der Angabe des Wurzelverzeichnisses „/“ und sind unabhängig vom aktuellen Verzeichnis. Relative Pfadangaben sind es aber. Sie gehen immer vom aktuellen Verzeichnis aus und beschreiben nur die relative Position der gesuchten Dateien im Dateisystem.

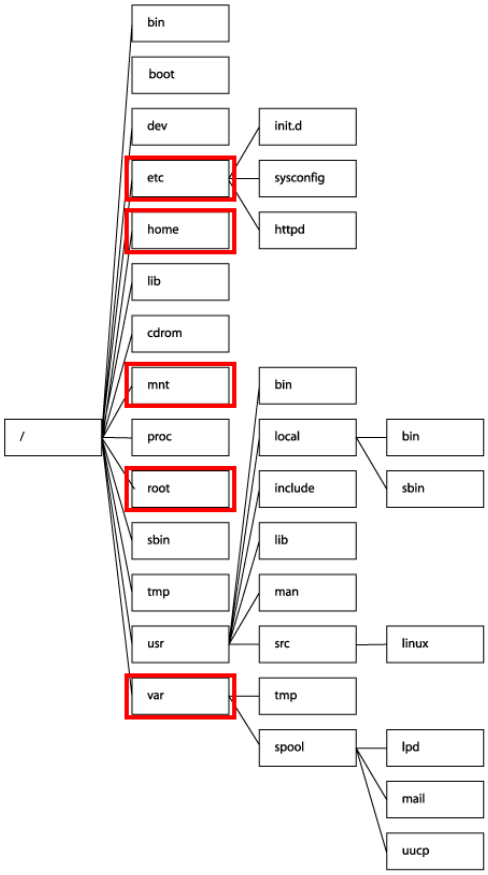

Abbildung 4: Dateiverzeichnisbaum unter Linux. Die wichtigsten sind in rot markiert.
Quelle: https://www.selflinux.org/selflinux/html/verzeichnisse_unter_linux01.html

Beispiel:
Wir gehen davon aus, wir befinden uns im Vezeichnis `/home/alice` und wollen die Datei `test.txt` im Home-Verzeichnis des Nutzers praktikus adressieren.

- **Absolut:** `/home/praktikus/test.txt`
- **Relativ:** `../praktikus/test.txt`

In der relativen Pfadangaben geben wir zuerst „`..`“ an und gehen damit ins nächsthöhere Verzeichnis „`home`“. Von dort aus gehen wir wie in der absoluten Angabe über `praktikus` nach `test.txt`. Weitere Informationen über den Inhalt der wichtigen Verzeichnisse erhalten sie an der Stelle nicht. Dafür sei auf das Modul Betriebssysteme verwiesen.

Nun wollen wir einige Befehle zur Navigation und Arbeit im Dateisystem betrachten. Der erste ist der folgende:

```
$   cd <Zielverzeichnis>
```

cd steht für change directory, also dem Wechsel des Arbeitsverzeichnisses. Durch die Angabe des Zielverzeichnisses, egal ob absolut oder relativ, gelangen Sie in das angegebene Zielverzeichnis. *Versuchen sie in folgende Verzeichnisse zu wechseln und verifizieren Sie das Arbeitsverzeichnis mit **pwd**. In welchen Verzeichnissen befinden Sie sich nach den entsprechenden Kommandos? Stellen Sie etwas fest? Warum ist das so?*

- /home/alice
- ~
- ..
- /root

Bei dem Wechseln in das Verzeichnis /root und /home/alice kommt die Fehlermeldung „keine Berechtigung“. Das liegt an den fehlenden Root-Rechten, um dieses Verzeichnis einzusehen.

Eine wichtige Operation, die regelmäßig gebraucht wird ist das Kopieren. Dabei kann, wie bei Windows Systemen, unterschieden werden zwischen dem Verschieben oder einfachem Kopieren. Auch Linux unterscheidet hier. Zum Kopieren nutzen wir den ersten und zum Verschieben (Kopieren und Löschen im Quellverzeichnis) den zweiten der folgenden Befehle. Deren Syntax ist sehr ähnlich. An dieser Stelle sei noch einmal darauf verwiesen, dass wir empfehlen, die Manpages der einzelnen Befehle stets zu konsultieren.

```
$   cp <Quellverzeichnis/Datei> <(Zielverzeichnis)/Datei> [-option]
$   mv <Quellverzeichnis/Datei> <(Zielverzeichnis)/Datei> [-option]
```

cp steht an der stelle für „copy“ und mv für „move“. Beim Kopieren oder Verschieben von Ordnern muss ggf. die Option „**-r**“ für recursive angegeben werden. Es sei außerdem erwähnt, dass **mv** auch dazu geeignet ist, Dateien umzubenennen, indem die Quelldatei der alte und die Zieldatei der neue Name der Datei ist. *Kopieren Sie nun die Datei **file4alice** in ihrem Home-Verzeichnis in das Home-Verzeichnis von alice.*

```
$   cp file4alice /home/alice/
```

Um neue Dateien anzulegen, können viele Möglichkeiten eingesetzt werden. Um eine einfache, leere Datei im Dateisystem anzulegen, gibt es den Befehl touch. touch ist ein Befehl, um die letzte Zugriffs- und Änderungszeit zu manipulieren. Demnach legt touch eine Datei an, wenn der angegebene Dateiname noch nicht existiert, ansonsten werden einfach die genannten Eigenschaften der Datei manipuliert. In ihrem Home-Verzeichnis finden Sie eine Datei touch.M3. *Ändern Sie bitte die Zugriff- und Änderungszeit der Datei ab! Erstellen Sie eine Datei mit einem noch nicht vergebenen Namen und überprüfen Sie deren Existenz. **Zusatz:** Verändern Sie **ausschließlich** die Änderungszeit der Datei touchModTime! Überprüfen Sie den Erfolg ihrer Eingabe!*

```
$   Touch touch.M3
$   Touch neueDatei
$   Touch -m touchModTime
```

Dateien können auch außerdem mit dem „>“-Operator erstellt werden. Dabei kann man einen beliebigen Befehl verwenden, der eine textuelle Ausgabe erzeugt und diese in die neue Datei umleitet. Beispiel:

```
$   echo „das ist ein Test“ > something.txt
```

Damit wird eine neue Datei „**something.txt**“ angelegt, welche den Inhalt „das ist ein Test“ enthält. *Schreiben Sie einen beliebigen Satz in die Datei echoedIn.file.*

```
$   echo „Satz“ > echoedIn.file
```

Ein neues Verzeichnis kann über den folgenden Befehl erstellt werden, **mkdir** (make directory). Auch hier muss lediglich der Befehl angegeben werden mit dem zu erstellenden Verzeichnis. Entnehmen Sie mögliche Optionen den Manpages.

```
$   mkdir <Verzeichnisname>
```

Erstellen Sie ein Verzeichnis unter dem Pfad „`/home/praktikus/dir1/dir2/targetdir`*“! Welche Möglichkeiten bestehen, um das Verzeichnis* `targetdir` *und den Pfad* `/home/praktikus/dir1/dir2/` *zu erstellen? Konsultieren Sie die Manpages!*

```
$   cd ~
$   mkdir -p dir1/dir2/targetdir
```

Man kann das entweder so machen oder man erstellt die Verzeichnisse einzeln und wechselt dann immer in das jeweils tiefere nach dem Erstellen.

Ebenfalls können Dateien und Verzeichnisse gelöscht werden. Dazu können die folgenden Befehle verwendet werden:

```
$   rm <Dateiname>              # Löschen einer Datei
$   rm -r <Verzeichnisname>     # Löschen mit -r von Verzeichnissen
$   rmdir <Verzeichnisname>     # Oder Verzeichnis löschen mit remove directory
```

rm ist ein Befehl zum Löschen von Dateien. Mit der Angabe der Option „**-r**“ können aber auch rekursiv Verzeichnisse gelöscht werden. Optional kann auch das Gegenstück von **mkdir** verwendet werden, **rmdir**. Auch hier wird lediglich der Name des zu löschenden Verzeichnisses als Argument mitgegeben.

Ein weiterer deutlich komplexerer Befehl (wenn man möchte) ist **find**. Er wird benutzt, um Dateien zu finden. Eine einfache Anwendung soll an der Stelle ausreichen. Wir gehen davon aus, dass wir wissen, wie die Datei heißt, die wir suchen oder zmd. wissen, welchem Aufbau deren Name folgt. Damit können wir folgende Befehle schreiben:

```
$   find -name <Dateiname>
$   find -regextype grep -regex <grep Ausdruck>
```

find sucht bei den angegebenen Befehlen nur im aktuellen und den darunterliegenden Verzeichnissen. Per „**-name**“ geben wir den ganzen Namen an und mit „**-regex**“ einen regulären Ausdruck, der auf den richtigen Dateinamen passt. Dabei kann per „-regextype“ die Art des zu verwendeten Regex-Schemas angegeben werden. In Linux arbeiten wir oft mit dem Befehl grep, weshalb dessen Schema angenommen wird. Weitere Schemata finden Sie mit dem Befehl:

```
$   find -regextype help
```

Zu regulären Ausdrücken erfahren Sie später noch mehr, da sie ein wichtiges Tool in vielerlei Anwendungen sind.
In welchem Verzeichnis befindet sich die Datei findM3.here? Gehen Sie von Ihrem Home-Verzeichnis aus!

```
$   find -name findM3.here
$   ./Dokumente/.secret/.moreS3cr3t/findM3.here
```

Nutzer- und Gruppenverwaltung und Umgebungsvariablen

Auch Linux verwaltet Nutzer und entsprechende Nutzergruppen im System. Ein einfacher Befehl, um herauszufinden, wer man selbst ist, ist **whoami** (wer bin ich). *Geben Sie den Befehl in das Terminal ein und lassen sie sich ihren eigenen Namen anzeigen.*

```
$   whoami
```

Um zu sehen, wer gerade noch an dem System angemeldet ist, verwenden wir einfach nur **who**. Damit sehen wir mit welchem Namen, in welchem Terminal und wann sich ein Nutzer am System angemeldet hat. *Testen Sie auch diesen Befehl und schauen Sie sich dessen Ausgabe an.*

```
$   who
```

Einige weitere Infomationen über den Nutzer finden Sie auch mittels **id**. Hier sehen Sie welche **uid** (Nutzer-ID), **guid** (Gruppen-ID) der Nutzer hat und zu welchen Gruppen der Nutzer gehört. Weitere Informationen über die aktuelle Sitzung zeigen Sie mit dem Befehl **env** (Environment – Umgebungsvariablen) an. Hier sehen Sie auch Angaben, wie **USER, HOME, PWD, PATH, TERM, LANG** und die verwendete **SHELL**. *Lassen Sie sich die Umgebungsvariablen im Terminal ausgeben.*

```
$   env
```

Einzelne Variablen (auch benutzerdefinierte) können Sie mit der folgenden Syntax ausgeben:

```
$   echo ${<Variablenname>}
```

Um den Wert einer Variable zurückzugeben muss der Name der Variable immer in geschwungenen Klammern geschrieben werden mit einem **$**-Zeichen (Dollar) davor. *Lassen Sie sich die verwendete Sprache (LANGUAGE) ausgeben.*

```
$   echo ${LANGUAGE}
```

Um zu sehen, welche Nutzer und Gruppen am System eingerichtet sind, können folgende Befehle verwendet werden:

```
$   cut -d: -f1 /etc/passwd
$   cat /etc/group
```

Der erste Befehl cut nutzt die Option „-d:“. Sie zeigt an welches Trennzeichen (delimiter) eine Datei verwendet. „**-f1**“ steuert, dass nur die erste Spalte ausgegeben wird. „**/etc/passwd**“ ist die Datei im Dateisystem, die die Nutzer über angelegte Nutzer enthält. Wenn Sie die gesamte Datei **/etc/passwd** sehen wollen, geben Sie folgenden Befehl ein:

```
$   cat /etc/passwd
```

Mit entsprechenden Root-Rechten können Sie auch neue Nutzer anlegen. Diese können Sie mit dem Befehlszusatz sudo erlangen. Sudo ist die Abkürzung für Superuser do, also das Ausführen eines Befehls mit Superuser-Rechten. Manchmal kommt es vor, dass man Befehle aus einer privilegierten Shell starten muss. Dazu nutzen wir den Befehl:

```
$   sudo -s
```

und starten damit eine neue Shell mit Root-Rechten. Ein einfaches Anlegen eines neuen Nutzers erfolgt mit folgender Syntax:

```
$   sudo adduser <Nutzername>
```

Legen Sie einen neuen Nutzer mit dem Namen alice an. Sie werden dazu aufgefordert ihr Passwort einzugeben. Lesen Sie sich danach die Terminalausgaben aufmerksam durch. Welche Aktionen stellen Sie fest? Was wird für Sie automatisch erledigt? Vergeben Sie alice ein neues Passwort und bestätigen Sie dieses. Alle anderen Angaben können Sie ausfüllen, sind aber nicht notwendig. Bestätigen Sie die Eingabe jeweils mit Enter. Geben Sie zum Schluss ein „J“ ein, um die Richtigkeit der Angaben zu bestätigen. Legen Sie auch noch einen Nutzer bob an.

```
$   sudo adduser alice
$   automatisches Erstellen von Home-Verzeichnis, Gruppe alice (guid)
```

```
$   Automatisches Hinzufügen von alice zur Gruppe alice
$   sudo adduser bob
```

Ebenso lassen sich neue Gruppen mit dem Befehl addgroup erstellen. Auch dazu benötigen Sie Superuser-Rechte. *Erstellen Sie eine neue Gruppe employees.*

```
$   sudo addgroup employees
```

Den Gruppen lassen sich mehrere Nutzer hinzufügen. Für jeden neuen Nutzer wird standardmäßig eine neue Gruppe angelegt. Hinzufügen lässt sich ein Nutzer einer Gruppe mit dem Befehl usermod:

```
$   sudo usermod -aG <Gruppenname> <Nutzername>
```

Fügen Sie die Benutzer praktikus, alice und bob zur erstellten Gruppe employees hinzu. Prüfen Sie korrekte Durchführung der Befehle, indem Sie sich erneut die eingerichteten Gruppen anzeigen lassen.

```
$   sudo usermod -aG employees alice
$   sudo usermod -aG employees bob
$   sudo usermod -aG employees praktikus
```

Nutzer und Gruppen können ebenfalls gelöscht werden mit den folgenden Befehlen:

```
$   sudo deluser <Nutzername>
$   sudo delgroup <Gruppenname>
```

Dabei ist zu beachten, dass das Home-Verzeichnis eines Nutzers bei dessen Löschung nicht automatisch gelöscht wird. Dazu sollte zusätzlich die Option „—remove-home“ angegeben werden. Ebenso sei zu beachten, dass beim Löschen eines Nutzers automatisch dessen automatisch generierte Gruppen ebenfalls gelöscht wird. *Löschen Sie den zuvor angelegten Nutzer bob und dessen Home-Verzeichnis.*

```
$   sudo deluser --remove-home bob
```

Paketierung

Wie auch Windows erhält Linux stetig Updates zu installierten Paketen und Anwendungen, nur sieht das etwas anders aus. Aufgrund dass Linux ein quelloffenes System ist, werden auch offene Paketquellen verwendet, von denen Softwareupdates in Paketen abgerufen werden. Weiterhin werden Aktualisierungen nicht automatisch abgerufen und müssen vom Nutzer initialisiert werden. Einer der meistgenutzten Befehle für die Paketverwaltung ist **apt** (aptitude). Dessen Verwendung soll kurz betrachtet werden.

Um Updates auf einem System zu installieren, brauchen Sie generell Superuser-Berechtigungen. Dazu geben Sie, wie bereits gelernt, „**sudo**“ vor dem Befehl an. Die Konfigurationsdatei für zusätzliche Repositories in apt liegt unter dem Verzeichnis **`/etc/apt/sources.list`**. Die offiziellen Paketquellen finden Sie hingegen unter **`/etc/apt/sources.list.d/official-package-repositories.list`**. Diese Datei sollte von Ihnen nicht verändert werden. *Aktualisieren Sie die bestehenden Pakete mit dem folgenden Befehl:*

```
$   sudo apt update
```

Hierdurch werden die Paketlisten der Repositories gelesen und Sie erhalten Sie eine Übersicht, wie viele Pakete aktualisiert werden können. Bedenken Sie: `update` aktualisiert nicht das System, sondern nur die Paketlisten. Das eigentliche Update der Pakete erfolgt durch den Befehl:

```
$   sudo apt dist-upgrade
```

Durch den Befehl „**`add-apt-repository <Repository>`**“ kann eine neue Paketquelle hinzugefügt werden. Dies wird teilweise bei der Installation von PHP für Webserver verwendet. Dabei kann die Quelle **ppa:ondrej/php** als

neues Repo hinzugefügt werden und folglich darüber PHP installiert werden. *Versuchen Sie diese mit dem erworbenen Wissen hinzuzufügen.* Denken Sie daran, dass Sie Superuser-Rechte benötigen. Konsultieren sie ggf. die Manpage des Befehls.

```
$   sudo add-apt-repository ppa:ondrej/php
```

Aktualisieren Sie folgend erneut die Paketlisten und installieren Sie verfügbaren Updates der Pakete. Als nächstes schauen wir uns an, wie wir neue Pakete bzw. Anwendungen installieren. Dazu nutzen wir ebenso **apt**. Mit der Syntax:

```
$   sudo apt install <Paketname> [-y]
```

können wir das Paket mit dem Namen **<Paketnamen>** installieren. Die folgende Option „**-y**“, bewirkt, dass wir während des Installationsprozesses keine interaktive Bestätigung geben müssen, dass das Paket installiert werden darf. Das passiert mit „**-y**“ automatisch. Im forensischen Umfeld nutzen wir oft ein Tool, welches das klassische forensische Format EWF unterstützt. Wie finden wir heraus, welche Pakete dafür installieren können? *Geben Sie doch mal den folgenden Befehl ein und suchen sie Pakete, welche „ewf“ enthalten:*

```
$   apt search ewf
```

Sie bekommen nun schließlich eine Liste mit Paketen, welche auf das Suchkriterium zutreffen aus den Ihnen zur Verfügung stehenden Paketquellen. *Installieren Sie nun bitte die Bibliothek, welches das Expert Witness Compression Format unterstützt.* Als Gegensatz dazu kann man auch installierte Pakete wieder deinstallieren. Dazu nutzen wir die Syntax:

```
$   apt install -y libewf2
$   apt remove <Paketname>
```

Auf Ihrer virtuellen Maschine ist ein Apache2-Webserver, den Sie an der Stelle hier nicht benötigen. *Deinstallieren Sie diesen bitte. Der Paketname ist **apache2**. Bestätigen Sie bei Aufforderung mit der Eingabe von „Y“.*

```
$   apt remove apache2
```

Verwaltung von Benutzerrechten an Dateien

Jede Datei hat Berechtigungen und einen Eigentümer, bzw. eine Eigentümer Gruppe. Die Berechtigung für eine Datei können mit dem **`ls -l`** Befehl angezeigt werden. Bei Eingabe des Befehls erhält man das Long Listing Format, wie in Abbildung 5 gezeigt.

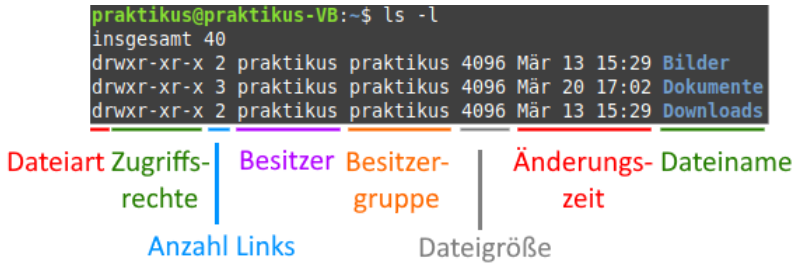

Abbildung 5: Felder des Long Listing Formats durch den Befehl ls -l

Man kann Berechtigungen einer Datei an einen anderen Benutzer abgeben. Dafür nutzen wir den Befehl chown (change file owner and group). Dieser ist mit der folgenden Sytanx zu behandeln:

```
$   [sudo] chown [-option] <Nutzername>:<Gruppenname> <Dateiname>
```

*Überschreiben Sie die Rechte der Datei **`file4alice`**, welches Sie in das Home-Verzeichnis von alice gelegt haben, an alice und der erstellten Gruppe employees.*

```
$   sudo chown alice:employees file4alice
```

Mit den Besitzrechten gehen auch die Zugriffsrechte auf eine Datei einher. Diese haben den folgenden Aufbau:

Tabelle 2: Notation der Zugriffsrechte unter UNIX

Besitzer (user – u)			**Gruppe (group – g)**			**Andere (others – o)**		
r	w	x	r	-	x	r	-	x
Lesen	Schreiben	Ausführen	Lesen	Schreiben	Ausführen	Lesen	Schreiben	Ausführen

Dem Beispiel ist zu entnehmen, dass der Besitzer der Datei alle Rechte hat (Lesen, Schreiben, Ausführen). Die Gruppe der Datei und Andere Benutzer haben hingegen nur das Recht zu lesen und auszuführen. Sie dürfen die Datei nicht schreiben. Die Zugriffsrechte können mit dem Befehl **`chmod`** (change file mode bits) geändert werden. Dafür bieten sich verschiedene Möglichkeiten an:

```
$   chmod [-option] <Oktett> <Datei>
$   chmod [ugo][+-][rwx] <Datei>
```

Im Ersten Kommando muss ein Oktett angegeben werden, welches die neuen Zugriffsrechte darstellt. Dabei kann die Angabe **`rwx`** auch als Folge von drei Bits angesehen werden. Je nachdem welche Berechtigungen gesetzt werden sollen, werden die entsprechenden Bits gesetzt:

Tabelle 3: Oktettnotation für Zugriffsrechte

Kürzel	r	w	x
Bitstelle	2	1	0
Wertigkeit	4	2	1

Beispiel:
Nehmen wir an, wir wollen einer Nutzer Schreib- und Leserechte geben, aber keine Rechte zum Ausführen der Datei. Damit müssen wir Bitstelle 1 und 2 setzen. Um die oktale Angabe zu erhalten, addieren wir die Wertigkeiten der einzelnen Bits zusammen. Es ergibt sich 2 + 4 = 6. Damit muss für das Setzen von r und w der Wert 6 angegeben werden. Für die Angabe bei chmod brauch der Befehl aber die Berechtigungen für alle drei Nutzerarten. Dafür können folgende Beispiele gezeigt werden

- **666** alle Nutzerarten haben die gleichen Rechte (rw-)
- **700** der Besitzer hat alle Rechte (rwx) und dessen Gruppe und Andere haben keine (---)

Alternativ kann das Geben oder Entziehen einzelner Berechtigungen über einen intuitiveren Weg erfolgen. Dafür nehmen die zweite Möglichkeit. Wir geben an für welche Nutzerart (**ugo**) wir welche Berechtigung (**rwx**) hinzufügen (+) oder entziehen wollen (-). Beispielsweise können wir dem Besitzer der Datei die Berechtigung für das Ausführen mit folgendem Befehl geben:

```
$   chmod u+x <Dateiname>
```

Geben Sie dem Nutzer des Verzeichnisses Musik in ihrem Home-Verzeichnis und allen darunterliegenden Dateien alle Rechte und dessen Gruppe sowie anderen Nutzer nur das Rechte das Verzeichnis und die Dateien zu lesen. Nutzen Sie die Manpage!

```
$   cd ~
$   chmod -R 744 Musik
```

Als letztes Thema soll das Ausgeben der Größe einer Datei angeschnitten werden. Dazu nutzen wir den Befehl du (disk usage). Sinnvoll ist die Angabe mit der Option „**-h**“ (human readable), um die Dateigröße „menschenlesbar“ in KB, MB oder GB sehen zu können. Es wir die folgende Syntax verwendet:

- **du -h <Dateiname>**