

# Sicherheit

Team High Society

3. Juli 2006

## Inhaltsverzeichnis

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Einleitung</b>                                | <b>3</b>  |
| <b>2</b> | <b>Sicherheit</b>                                | <b>4</b>  |
| 2.1      | Security through obscurity . . . . .             | 5         |
| <b>3</b> | <b>Verschlüsselung</b>                           | <b>6</b>  |
| 3.1      | Festplattenverschlüsselung . . . . .             | 6         |
| 3.1.1    | Komplette Partition/Container . . . . .          | 6         |
| 3.1.2    | Windows . . . . .                                | 6         |
| 3.1.3    | Linux . . . . .                                  | 7         |
| 3.1.4    | Mac OS X . . . . .                               | 9         |
| 3.2      | IRC-Verschlüsselung . . . . .                    | 10        |
| 3.2.1    | Rechte der IRCOps . . . . .                      | 10        |
| 3.2.2    | Die Wahl des Channelnamen . . . . .              | 11        |
| 3.3      | E-Mail-Verschlüsselung . . . . .                 | 12        |
| 3.3.1    | PGP . . . . .                                    | 12        |
| 3.3.2    | E-Mails sicher empfangen und versenden . . . . . | 12        |
| <b>4</b> | <b>Daten vernichten</b>                          | <b>13</b> |
| 4.1      | Linux . . . . .                                  | 13        |
| 4.2      | Shred/Wipe . . . . .                             | 13        |
| <b>5</b> | <b>Passwörter</b>                                | <b>14</b> |
| 5.1      | Passwörter aufbewahren . . . . .                 | 14        |
| 5.2      | Passwörter selbst erstellen . . . . .            | 15        |
| 5.3      | Passwörter erzeugen . . . . .                    | 16        |

---

|          |  |           |
|----------|--|-----------|
| <b>6</b> | <b>Programme absichern</b>                               | <b>18</b> |
| 6.1      | glftpd . . . . .   | 18        |
| 6.2      | Linux-Server . . . . .                                   | 18        |
| 6.3      | IRC-Clients . . . . .                                    | 18        |
| 6.4      | FlashFXP . . . . .                                       | 19        |
| <b>7</b> | <b>Sicherheit im echten Leben</b>                        | <b>20</b> |
| 7.1      | Kontakt . . . . .  | 20        |
| 7.2      | Keine unnötigen Blicke riskieren... . . . . .            | 20        |
| <b>8</b> | <b>Hausdurchsuchungen</b>                                | <b>21</b> |
| 8.1      | Was darf die Polizei / Welche Rechte habe ich? . . . . . | 21        |
| 8.2      | Wie verhalte ich mich? . . . . .                         | 21        |
| <b>9</b> | <b>Anhang</b>  | <b>23</b> |

# 1 Einleitung

Sicherheit in all ihren Facetten wird ein immer wichtigeres Thema. Die GVV und die Behörden schlafen nicht ewig und selbst wenn es lange gedauert hat: Mittlerweile sind sie recht gut informiert. Die Zeiten, in denen man einfach Kopien von Software tauschen konnte, sind längst vorbei. Durch die immer weiter gehende Verbreitung der Kopien – nicht nur durch Peer-2-Peer-Software – sind die (so falsch genannten) „Raubkopien“ ein ernstzunehmendes Problem für die „Rechteinhaber“.

Dieses Paper dreht sich um das Thema Sicherheit unter dem Aspekt, wann *keine* Sicherheit besteht, wie und wann Scheinsicherheit entsteht und wodurch man größtmögliche Sicherheit erreichen kann.

Die hier abgedeckten Schutzmaßnahmen sind für den Rahmen der Anfertigung und/oder Verbreitung von Kopien gedacht, eindeutig distanzieren – allein schon wegen ethisch/moralischen Bedenken – möchten wir uns von so genannten Sellern, die Kopien gegen Geld verbreiten.

Die Anleitungen in diesem Paper sind so einfach wie möglich gehalten, erfordern aber doch grundlegendes Wissen über Computer, ich unterstelle einfach mal den Lesern, die dieses Wissen nicht besitzen, dass sie ohnehin nicht zu der Zielgruppe dieses Papers gehören.

Außerdem bemühe ich mich, auf alle verbreiteten Betriebssysteme einzugehen. Dazu gehören – meiner Meinung nach – Mac OS X, Linux und Windows.

Auch dieses Paper wird Fehler enthalten und im Laufe der Zeit veralten. Wenn du also Ergänzungen hast, dir ein Fehler aufgefallen ist oder dir eine von mir behandelte Möglichkeit absolut nicht mehr zeitgemäß erscheint, schreib mir bitte eine E-Mail.

## 2 Sicherheit

Was ist Sicherheit überhaupt?

Sicherheit bezeichnet einen Zustand, der frei von unvermeidbaren Risiken der Beeinträchtigung ist oder als gefahrenfrei angesehen wird. (Wikipedia<sup>1</sup>)

Wir werden Sicherheit erreichen, indem wir versuchen, so wenige Spuren unserer Taten zu hinterlassen, wie es möglich ist und die Spuren, die noch existieren, ins Leere laufen lassen. Bedenken wir erst einmal ein paar grundlegende Punkte:

**Internetverbindung:** Die verwendete Internetverbindung ist meistens auf den echten Namen angemeldet und lässt somit schnell eine eindeutige Identifikation über die IP-Adresse zu. Auch bei Dial-Up-Verbindungen ist eine Identifikation möglich, sofern man sich nicht über einen fremden Telefonanschluss einwählt.

Vorschlag: Eventuell hast du eine(n) nette(n) Nachbar(i)n, der/die gerne einen Internetanschluss teilen würde – natürlich nur, um Kosten zu sparen ;-).

**Netzwerk:** Notebookbesitzer möchten auf Wireless-LAN in der Regel nicht mehr verzichten – allerdings ist eine Kette nur so stark wie ihr schwächstes Glied. Es gibt WLAN-Router, die alle Datenpakete in die Luft senden, auch wenn keine Geräte im Netz eingeloggt oder aktiv sind (wie ein Hub also). Achte also auf eine starke und ungeknackte Verschlüsselung (WEP wurde beispielsweise geknackt), wenn du ein WLAN betreibst.

Vorschlag: WPA oder WPA2 mit einem regelmäßig wechselndem, sicheren Schlüssel einrichten.

**Echter Name:** Es ist wohl klar, dass man den eigenen Namen nicht leichtfertig verbreiten sollte. Dass es aber theoretisch auch langt, unter der selben IP bei eBay einen Artikel zu (ver)kaufen und im IRC ein Gespräch über heikle Themen zu führen, daran denken die meisten nicht. Man sollte davon ausgehen, dass ein Provider alle Daten vereinen kann.

Vorschlag: Verschiedene Proxies für die jeweiligen Websites verwenden, auf denen man seinen echten Namen eingeben will/muss. **Privoxy** bietet hierfür zum Beispiel Optionen.

**Vorsicht!** Die meisten SMTP-Server sind so eingestellt, dass sie eine Ident-Anfrage zum Client senden, wenn dieser eine E-Mail verschickt. Dadurch ist dein Ident in vielen E-Mail-Headers vorhanden, wenn du dauerhaft einen Ident-Server laufen

---

<sup>1</sup><http://de.wikipedia.org/wiki/Sicherheit>

hast. Am besten ist es, den Identserver nur dann zu aktivieren, wenn man ihn wirklich braucht (beim Login auf einen FTP zum Beispiel). FlashFXP bietet hierfür eine Einstellung unter „Optionen“, „Einstellungen“, „Verbindung“, „Ident“, „Nur aktivieren während des Verbindungsaufbaus“. Das Verstecken der Identreplies ist wichtig, da IP-Adressen inklusive Identreplies viel eindeutiger zu E-Mailadressen zugeordnet werden können.

## 2.1 Security through obscurity

Mit „Security through obscurity“ (dt.: Sicherheit-durch-Unklarheit) ist das Bestreben nach Sicherheit durch Geheimhaltung gemeint. Damit ist beispielsweise das „sichern“ von Pornofilmen in einem Ordner wie „C:\Windows\System32\Drivers\etc\Hidden“, gemeint – wenn man nicht weiß, wo man suchen soll, findet man die Filme nicht. Auch im echten Leben gibt es dieses Prinzip: Viele Leute verstecken beispielsweise ihren Hausschlüssel unter der Fußmatte oder in einem Blumentopf. Das Problem ist offensichtlich: Wenn man weiß, wo der Schlüssel versteckt ist, kann man die Haustüre öffnen. „Security by obscurity ist also der Versuch, Dinge geheim zu halten, die weite Verbreitung finden.“ (Wikipedia)

„Sicherheit, die nur auf der Geheimhaltung von Informationen beruht, stellt sich sehr oft als ungenügend heraus.“ (Wikipedia)

Das Problem bei Security through obscurity ist, dass sehr subjektiv geschätzt wird, wie wahrscheinlich es ist, dass der Angreifer bestimmte Kenntnisse über das Ziel hat. Meistens wird der Angreifer unterschätzt oder mit Informationen wird zu leichtfertig umgegangen.

Als einfache Merkregel könnte man also festhalten, dass man sich **nie** darauf verlassen sollte, dass der Angreifer den „Schlüssel“ (in diesem Fall nicht das Passwort) zur Lösung nicht kennt/nicht kennen kann.

## 3 Verschlüsselung

### 3.1 Festplattenverschlüsselung

Sollte der Fall eintreten, dass der eigene Rechner oder persönliche Daten in fremde Hände geraten, ist es ratsam, seine Daten durch Verschlüsselung vor fremden Augen sicher zu wissen. Bei den heutigen Prozessoren und PC-Geschwindigkeiten ist das Arbeiten mit verschlüsselten Daten im Normalfall nicht sonderlich langsam, es sei denn es handelt sich um sehr große Daten (Videodateien beispielsweise). Unter den drei meistgenutzten Betriebssystemen ist Verschlüsselung in der jeweils aktuellen Version meist von Haus aus möglich, durch Zusatz-Software jedoch zu verbessern.

#### 3.1.1 Komplette Partition/Container

Bei den meisten Verschlüsselungsprogrammen kann man auswählen, ob man einen Container erstellen möchte oder gleich die ganze Partition verschlüsseln. Wenn man die ganze Partition/Festplatte verschlüsselt, sieht das Betriebssystem nichts auf der Partition/Festplatte, sie wird nicht angezeigt. Erst nachdem man sie gemountet (eingebunden, „angeschlossen“) hat, wird sie vom Betriebssystem erkannt. Ein Container funktioniert genauso, nur befinden sich hier die verschlüsselten Daten auf einer anderen, unverschlüsselten (oder im Einzelfall natürlich auch verschlüsselten, ergibt aber wenig Sinn) Partition. Der Vorteil eines Containers ist, dass er vom Betriebssystem erkannt, aber nicht gelesen werden kann. So kann man einen Container zum Beispiel auf CD brennen oder auf einem USB-Stick transportieren (wie eine normale Datei eben), wohingegen man bei einer komplett verschlüsselten Partition erstmal ein Backup erzeugen müsste. Der Vorteil einer verschlüsselten Partition sind die schnelleren Dateizugriffe, da die Speicherverwaltung bei der Containermethode vom Betriebssystem abhängt und vor allem Windows dazu neigt, Daten zu fragmentieren ;-).

#### 3.1.2 Windows

Auf Windows gibt es recht viele Verschlüsselungsprogramme, das bekannteste ist sicher DriveCrypt. Ich möchte allerdings nicht auf DriveCrypt eingehen und von der Benutzung eher abraten, da es meiner Meinung nach recht umständlich zu bedienen ist und Geld kostet. Es ist ein Closed-source-Programm und keiner weiß, ob die Hersteller nicht eine Hintertür eingebaut haben oder sonst eine Möglichkeit kennen, an die Daten heranzukommen. In den USA ist eine solche Hintertür sogar gesetzlich vorgeschrieben!<sup>2</sup> „Wie jeder andere Software-Hersteller in den USA sind Lotus gezwungen, nur solche Produkte zu exportieren, welche die NSA knacken kann.“ (Heise Telepolis<sup>3</sup>)

DriveCrypt wird gerne dazu verwendet, den kompletten Rechner zu verschlüsseln. Dies ist prinzipiell nicht zu empfehlen, da es erst recht verdächtig ist, vor allem aber die

<sup>2</sup>Auf Heise Telepolis gibt es einen Artikel über Lotus, das eine solche Hintertür für die National Security Agency (NSA) besitzt: <http://www.heise.de/tp/r4/artikel/2/2922/1.html>

<sup>3</sup><http://www.heise.de/tp/r4/artikel/2/2922/1.html>

Ausrede, dass man gar keine verschlüsselten Daten auf dem Rechner hat, schlicht und einfach wegfällt. Ich finde, dass eine MPG-Datei, die nicht abspielbar ist (weil sie in Wirklichkeit ein Crypt-Container ist), um einiges unauffälliger, als einen Rechner, der nicht bootet und einen beim drücken von Enter nach einem Passwort fragt ;-).

Stattdessen gibt es das Open-source-Programm **Truecrypt**<sup>4</sup>, welches verspricht, keine Hintertür eingebaut zu haben: „TrueCrypt does not contain any mechanism or facility that would allow partial or complete recovery of your encrypted data without knowing the correct password or the key used to encrypt the data.“ (TrueCrypt FAQ<sup>5</sup>) – im Gegensatz zu **DriveCrypt** kann man dies jedoch auch nachprüfen, wenn man über Programmierkenntnisse verfügt. **TrueCrypt** kann komplette Partitionen verschlüsseln oder Crypt-Container erstellen. Zur Verfügung stehen die Verschlüsselungsverfahren AES, Blowfish, CAST5, Serpent, Triple DES, Twofish und Kombinationen selbiger. Hilfreich ist außerdem ein Benchmark, mit dem man messen kann, welches Verschlüsselungsverfahren das schnellste bei der eigenen Hardware ist. Empfehlenswert sowohl in Geschwindigkeit als auch in Sicherheit sind AES (bei Schlüssellänge 256 Bit) und Blowfish (variable Schlüssellänge, 256 Bit mindestens empfohlen). **TrueCrypt** hat ein sehr ausführliches, interessantes Handbuch mit weiteren Informationen zu den verwendeten Algorithmen und der Funktionsweise des Programms selbst.

**Tipp:** Eine Verknüpfung zum Unmounten aller Truecrypt-Partitionen/-Laufwerke macht sich in der Schnellstartleiste ganz gut, finde ich. Das Ziel der Verknüpfung muss so lauten:

```
"C:\Programme\TrueCrypt\TrueCrypt.exe" /d /f
```

Die Parameter stehen für „dismount“ und „force“, damit keine Rücksicht auf eventuelle Dateien, die in Benutzung sind, genommen wird – sollte man die Verknüpfung mal brauchen, ist einem das wohl herzlich egal ;-). Der Pfad zu TrueCrypt muss natürlich eventuell auch angepasst werden.

### 3.1.3 Linux

Bei Linux ist ab Kernelversion 2.6.10 das Verschlüsselungsmodul **dmccrypt** fester Bestandteil. Bei manchen Versionen muss es noch via „modprobe dm-crypt && modprobe sha256 && modprobe blowfish && modprobe aes“ nachgeladen werden. Außerdem werden wir das Tool „cryptsetup“ verwenden, es sollte also installiert werden, falls es noch nicht installiert ist.

Die nächsten beiden Abschnitte erklären, wie man eine gesamte Partition beziehungsweise einen Container verschlüsselt, im dritten Paragraph wird gezeigt, wie man ein Dateisystem auf dem Cryptdevice anlegt und es einbindet.

---

<sup>4</sup><http://www.truecrypt.org/>

<sup>5</sup><http://www.truecrypt.org/faq.php>

### Gesamte Partition

```
# Wir verschaffen uns eine Übersicht der Partitionen
fdisk -l /dev/hda

# Nachdem wir die Partition auf der vorher partitionierten
# Festplatte gefunden haben (hier /dev/hda6), erstellen
# wir das cryptdevice
cryptsetup -y -c aes -s 256 -h sha256 create crypted /dev/hda6
```

### Container

```
# Wir laden das loop-modul in den Kernel
modprobe loop

# Anschließend schreiben wir 10x1MB Zufallsdaten
# in die Datei /home/user/container
dd if=/dev/urandom of=/home/user/container bs=1M count=10

# Wir benutzen das loopmodul um den container als
# blockdevice zur Verfügung zu stellen
losetup /dev/loop0 /home/user/container

# Wir erstellen das device "crypted" auf dem loopbackdevice
# /dev/loop0
cryptsetup -y -c aes -s 256 -h sha256 create crypted /dev/loop0
# Man wird nun aufgefordert, ein Passwort einzugeben
```

### Allgemeines

```
# Wir formatieren das neu erstellte cryptdevice
mkfs.ext2 /dev/mapper/crypted

# Wir legen den Mountpunkt an
mkdir /mnt/crypted

# Und mounten es nach /mnt/crypted
mount -t ext2 /dev/mapper/crypted /mnt/crypted
```

### 3.1.4 Mac OS X

#### FileVault

Mac OS X bietet von Haus aus die Möglichkeit, das Homedir (via FileVault<sup>6</sup>) mit AES-128 verschlüsseln. Bei der Verschlüsselung via FileVault wird das Benutzerpasswort benutzt, sodass man das Image automatisch bei jeder Anmeldung benutzt und bei jeder Abmeldung wieder verschließt. Damit mehrere Benutzer gleichzeitig einen Server verwenden können, gibt es die Möglichkeit, den kompletten RAM (und das Swapfile) zu verschlüsseln, sodass die Benutzer keinen Zugriff auf die Daten des jeweils anderen haben. Wichtig bei FileVault, das man übrigens unter „Sicherheit“ in den Systemeinstellungen findet, ist, dass man die Optionen „Beim Beenden des Ruhezustandes oder Bildschirmschoners ein Kennwert verlangen“ und „Automatisches Anmelden deaktivieren“ aktiviert sind.

#### Diskimages

Verschlüsselte Diskimages funktionieren genauso wie Container unter Windows, man sollte jedoch darauf achten, dass man beim Passwort eingeben selbiges nicht versehentlich im Schlüsselbund speichert (außer man möchte mit der Benutzeranmeldung gleichzeitig die verschlüsselten Daten benutzen können, wie bei FileVault).

#### OpenSSL

Auch via OpenSSL, was standardmäßig installiert ist, kann man Dateien verschlüsseln. Dies ist zwar etwas unkomfortabler als mit den von OS X über die GUI bereitgestellten Methoden, aber soll hier trotzdem erwähnt werden.

```
# Verschlüsselt <Quelldatei> mit 128bit Blowfish-  
# Verschlüsselung und schreibt das Ergebnis in <Zieldatei>  
openssl bf -salt -in <Quelldatei> -out <Zieldatei>  
  
# Die Originaldatei sollte man dann löschen  
rm -fP <Quelldatei>  
  
# Entschlüsselung der selben Datei  
openssl bf -d -in <Zieldatei> -out <Quelldatei>
```

<sup>6</sup><http://www.apple.com/de/macosex/features/filevault/>

## 3.2 IRC-Verschlüsselung

Das IRC ist das am häufigsten benutzte Kommunikationsmedium und wurde daher auch mit SSL ausgestattet. Es gibt selbstverständlich noch zahlreiche weitere Methoden, wie zum Beispiel Plugins, die Blowfish verwenden.

SSL und Blowfish setzen an zwei verschiedenen Punkten an. Eine SSL-Verbindung wird vom Client zum Server aufgebaut. In IRC-Netzen gibt es aber mehrere Server (im EFNet ca 50), was SSL nicht unterstützt. Zwischen den einzelnen Servern müssen also auch SSL-Verbindungen aufgebaut werden. Damit der Server nun den eigentlichen Text von Benutzer A auf Server 1 zu Benutzer B auf Server 2 bringen kann, muss er ihn erst ent- und dann wieder verschlüsseln.

Blowfish hingegen verschlüsselt nicht die gesamte Kommunikation zwischen Client und Server sondern nur die einzelnen Nachrichten (PRIVMSGs genannt). Die eigentliche Entschlüsselung des Textes findet also erst auf dem Zielrechner statt.

Benutzt der Anwender also eine SSL-Verbindung in einem Netzwerk wie zum Beispiel dem Linknet und ein Administrator eines Servers bastelt ein bisschen an der IRC-Server-Software herum (gegen Bezahlung eventuell?), sodass die entschlüsselte Nachricht bevor sie an den nächsten Server oder an den Benutzer weitergeleitet wird in einer Datei landet, entsteht die typische Situation der Scheinsicherheit. Der Benutzer denkt, dass seine Verbindung sicher ist und geht daher offener mit sensiblen Informationen um.

Das heißt nicht, dass SSL schlecht ist – im Gegenteil! Eine SSL-Verbindung hilft, vor Attacken oder Überwachung der Internetprovider zu schützen. T-Com, Arcor und die anderen Provider sehen dann nicht mehr, in welchen Channels man sich aufhält und wann man welche Aktion durchführt. Auch ohne das direkte Vorhandensein von sensiblen Informationen könnte ein Channelname oder ein privates Gespräch mit einer bestimmten Person verdächtig wirken.

Wer sich aber auf eine sichere Kommunikation zwischen den Endpunkten verlassen will, sollte auf jeden Fall zu Blowfish greifen.

**Vorsicht!** Sobald der Blowfish-Schlüssel irgendwo im Klartext bekannt wird, ist die komplette Verschlüsselung nutzlos. Blowfish-Schlüssel sollte man also nur per PGP-verschlüsselter E-Mail, automatisch DH1080-verschlüsselten, privaten Gespräch oder über eine SSL-Verbindung via HTTP oder FTP austauschen. Natürlich ist auch eine persönliche Übergabe des Schlüssels möglich, meist aber aufgrund der Entfernung nicht praktisch.

### 3.2.1 Rechte der IRCOps

Über die Rechte der IRCOps gibt es viele falsche Informationen. Was die IRCOps können ist natürlich vom Stand innerhalb des Netzwerks (Local operator, Network operator,

etc...) abhängig und von der verwendeten Server-Software sowie den geladenen Modulen für selbige.

Sehr große Netzwerke wie das EFNet, Linknet, IRCNet oder QuakeNet verwenden meistens eigene Serversoftware, die oft recht spartanisch ausgestattet ist, was die Rechteverwaltung angeht. Dies beruht darauf, dass es die genannten Netze schon lange gibt und Tradition für viele sehr wichtig ist („Früher ging’s doch auch ohne“).

Man kann in jedem Netzwerk davon ausgehen (sofern es nicht in die andere Richtung (= weniger Rechte für IRCOps) modifiziert wurde), dass die IRCOps Topics lesen können und alle Informationen über einen Benutzer haben, die das jeweilige WHOIS hergibt (echte IP-Adresse, Channels, etc).

**Vorschlag:** Eine simple Maßnahme gegen das mitlesen von Topics ist das Verschlüsseln mit Blowfish, was viele Plugins mittlerweile unterstützen.

Für die UnrealIRCd-Serversoftware gab es einige Zeit ein Spy-modul, was es IRCOps erlaubte, unsichtbar Channels zu betreten und den kompletten Text mitzulesen. Dass so etwas für andere Server auch existiert (wenn auch nicht offiziell), halte ich für sehr wahrscheinlich.

### 3.2.2 Die Wahl des Channelnamen

Bei einigen IRC-Servern gibt es dank der deutschen Umlaute Probleme mit Channelnamen. So sind zum Beispiel bei UnrealIRCd ein kleines „ö“ und ein großes „Ö“ verschiedene Buchstaben (IRC ist normalerweise case-insensitive), was laut den Entwicklern auch so gewollt ist.

Außerdem hat der beliebte Windows-IRC-Client mIRC (auch in aktueller Version, 6.16) ein Problem mit dem Eurozeichen (ebenfalls mit UnrealIRCd getestet): Wenn der Channelname nur aus Eurozeichen besteht, wird er in der Channelliste nicht angezeigt. Wenn also von allen IRCOps in einem Netzwerk bekannt ist, dass sie mIRC benutzen, könnte man einen Channel so verstecken – Security through obscurity funktioniert natürlich nicht, aber vielleicht fällt jemandem ja doch ein sinnvoller Einsatzzweck ein? :-)

## 3.3 E-Mail-Verschlüsselung

### 3.3.1 PGP

PGP gibt es für sehr viele Mailer mehr oder weniger komfortabel. Es ist somit als Standard für E-Mail-Verschlüsselung anzusehen.

PGP basiert auf Primzahlen und der Tatsache, dass man große Primzahlen zwar sehr schnell erzeugen, praktisch aber nicht in annehmbarer Zeit wieder zerlegen kann. Ein Teil der zwei Primzahlen, die benutzt wurden, um die dritte zu erzeugen, wird nun öffentlich gemacht (der so genannte Public Key) und der andere bleibt im Besitz des Eigentümers. Wenn nun jemand eine E-Mail an euch senden möchte, so lädt er sich euren Publickey herunter und verschlüsselt seine E-Mail damit. Nun kann sie niemand außer der Besitzer des Privatekey (der andere Teil der Primzahl) wieder entschlüsseln.

Weitere, tiefergehende Informationen dazu findet ihr auf <http://de.wikipedia.org/wiki/PGP>.

Ein Tutorial für Thunderbird (auf Windows/Linux) findet ihr auf <http://www.ch-becker.de/?Emails%20mit%20GnuPG>.

### 3.3.2 E-Mails sicher empfangen und versenden

Die meisten Mailserver und Mailprogramme bieten mittlerweile TLS-Verschlüsselung sowohl bei SMTP als auch bei POP3 an. Allen voran Google Mail, die gar keine unverschlüsselten Verbindungen mehr zulassen.

Man muss allerdings bedenken, dass gesicherte Verbindungen nicht sonderlich viel bringen, da die Mehrheit der Benutzer ihre Mails wohl ungesichert abrufen und/oder versenden (und beim Versenden oftmals die gesamte Mail zitieren) – man beugt aber zumindest bei zwei potentiell abgehörten Stellen (Client und Mailserver) vor.

## 4 Daten vernichten

### 4.1 Linux

Auf Linux funktioniert das bereits mit dem mitgegebenen Tool „dd“ und den Devices „/dev/urandom“ sowie „/dev/zero“. Um eine Partition/komplette Festplatte/Datei sicher zu löschen (erst mit Zufallsdaten, dann mit Nullbytes überschrieben), langen diese beiden Befehle:

```
dd if=/dev/urandom of=/dev/hda6 bs=10240 conv=noerror
dd if=/dev/zero of=/dev/hda6 bs=10240 conv=noerror
```

Hierbei wird die sechste Partition (hda6) der ersten Festplatte (hda) überschrieben. Einen Überblick über die vorhandenen Festplatten und Partitionen kann man sich mithilfe der Befehle „mount“, „df“ und „fdisk“ verschaffen. Der Parameter „bs“ steht für Blocksize, also wieviele Daten pro Zugriff geschrieben/gelesen werden. Mit „conv=noerrors“ werden Fehler unterdrückt, sodass man den Befehl starten kann und sich keine Sorgen machen muss, dass er zu früh abbricht, da ein Fehler aufgetreten ist (defekte Sektoren auf der Festplatte beispielsweise).

### 4.2 Shred/Wipe

Es gibt außerdem mehrere Tools für die Vernichtung von Daten, zum Beispiel „shred“ (bei den GNU coreutils<sup>7</sup> dabei, für das Vernichten kompletter Partitionen/Festplatten, Beispielaufruf: „shred -v /dev/hda6“) und „wipe“ (für das Vernichten einzelner Dateien, Beispielaufruf: „wipe -D /mnt/encrypted/secret.file“). Weitere Tools sind im Anhang auf Seite 23 verlinkt.

---

<sup>7</sup><http://www.gnu.org/software/coreutils>

## 5 Passwörter

Dass so simple Passwörter wie der Name der Freundin/Ehefrau keinen Schutz bieten, hat man spätestens nach den relativ oft auftauchenden Meldungen über Betrug bei eBay durch geknackte Passwörter mitbekommen.

Ein sicheres Passwort sollte – je nach Anwendungszweck und Möglichkeit (Begrenzungen durch das Programm) – mindestens 8, besser jedoch 20 Zeichen lang sein und willkürlich gewählte Zahlen und Buchstaben in Groß- und Kleinschreibung beinhalten. Sonderzeichen könnten problematisch werden, man sollte sich bei deren Verwendung überlegen, ob man eventuell an einer Tastatur mit anderem Tastaturlayout das Passwort verwenden will und ob die verwendeten Sonderzeichen dort überhaupt vorhanden sind. Auch akzeptieren manche Programme keine Sonderzeichen oder funktionieren damit nicht richtig.

### 5.1 Passwörter aufbewahren

In einem etwas älteren Handbuch habe ich mal gelesen, dass man sich einen Zettel mit dem Passwort unter die Tastatur legen sollte, für den Fall dass man das Passwort mal vergisst – das ist natürlich völliger Blödsinn und nicht zu empfehlen.

Genau wie andere sensible Daten sollte man Passwörter in einem verschlüsselten Container (Siehe auch „[3.1.1: Partition/Container](#)“, Seite [6](#)) ablegen oder ein Programm verwenden, das dies automatisch übernimmt, wie zum Beispiel `Keepass`<sup>8</sup> (leider nur für Windows und Linux verfügbar, die Mac OS X-Version funktionierte zum Zeitpunkt des Schreibens nicht richtig). Man kann natürlich auch beide Methoden gleichzeitig verwenden. `Keepass` ist Open-source-Software, von Closed-source-Software rate ich in diesem Fall ab, da ein Hersteller sehr leicht eine Hintertür einbauen könnte, vor allem, wenn die verwendete Verschlüsselung kein allgemein bekanntes, mathematisch geprüftes Verfahren ist.

Grundsätzlich sollte man sich das Masterpasswort für den verschlüsselten Container oder die Passwortdatenbank gut merken können und *nicht* zusätzlich irgendwo aufbewahren.

---

<sup>8</sup><http://keepass.sourceforge.net/>

## 5.2 Passwörter selbst erstellen

Eine gängige Möglichkeit, Passwörter zu erstellen, ist, sich einen oder mehrere Sätze zu überlegen und anhand der Anfangsbuchstaben der Wörter sowie der Satzzeichen das Passwort zu erstellen.

Nehmen wir uns zum Beispiel diese zwei Sätze:

Der 43. Präsident der Vereinigten Staaten ist George W. Bush. Mein Gott,  
wie ich ihn hasse!

Wir belassen die Groß- und Kleinschreibung sowie die Zahlen und Satzzeichen wie sie sind (dies sind die Zeichen, von denen in Passwörtern in der Regel zu wenig vorkommen) und betrachten nur noch die Anfangsbuchstaben der Wörter:

D 43. P d V S i G W. B. M G, w i i h!

Das Ganze noch mal ohne die zur besseren Lesbarkeit belassenen Leerzeichen:

D43.PdVSiGW.B.MG,wiih!

Und schon haben wir ein 22-stelliges Passwort, das zudem noch leicht zu merken ist ;-)<sup>9</sup>. Mit der Zeit entwickelt man ein Gefühl dafür, aus Sätzen die Anfangsbuchstaben und „relevanten“ Zeichen herauszufiltern.

---

<sup>9</sup>Passwörter aus Papers oder Büchern sind definitiv *nicht* für den tatsächlichen Einsatz zu empfehlen!

## 5.3 Passwörter erzeugen

Für die nicht ganz so Kreativen haben wir natürlich auch gesorgt: Programme wie `Keepass` bieten zum Beispiel die Möglichkeit, automatisch ein 20 Zeichen langes Passwort zu erstellen. Für Linux (getestet mit `bash 2.05b.0(1)-release`) und Mac OS X (10.4.5, ebenfalls `bash 2.05b.0(1)-release`) habe ich folgendes Script geschrieben:

```
#!/bin/sh

SPECIALCHARS=1
NUMBERS=1
UPPERLETTERS=1

if [[ ("${1}" == "--help") || ("${1}" == "-help") ||
      ("${1}" == "-h") ]]; then
    echo -e "\nSyntax: ${0} [-snu] [length]\n"
    echo "Creates password of [length]"
    echo "Options modify the table of available
          characters for password."
    echo -e "All options are turned ON by default
            for security.\n"
    echo -e "\t-s\tTurn off special characters"
    echo -e "\t-n\tTurn off numbers"
    echo -e "\t-u\tTurn off lower-/uppercase letters\n"
    exit 0
fi

while getopts snu var; do
    case $var in
        s) SPECIALCHARS=0 ;;
        n) NUMBERS=0 ;;
        u) UPPERLETTERS=0 ;;
    esac
done
shift $(( $OPTIND - 1 ))

if [ "${#1}" -gt "0" ]; then
    CHARS="${1}"
else
    CHARS=20
fi

OUTPUT=""
TABLE="abcdefghijklmnopqrstuvwxyz"
if [ "${UPPERLETTERS}" == "1" ]; then
    TABLE="${TABLE}ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

```
fi
if [ "${NUMBERS}" == "1" ]; then
    TABLE="${TABLE}0123456789"
fi
if [ "${SPECIALCHARS}" == "1" ]; then
    TABLE="${TABLE}#+*@!$%&/()=?°"
fi

while [ "${#OUTPUT}" -lt "${CHARS}" ]; do
    RND=$RANDOM
    let "RND %= ${#TABLE}"
    OUTPUT="${OUTPUT}${TABLE:$RND:1}"
done

echo "${OUTPUT}"
```

Das Script ist auch auf <http://www.high-society.at/> verfügbar.

## 6 Programme absichern

Einige Programme werden ohne sich groß mit der Konfiguration zu beschäftigen installiert, obwohl diese Standardkonfiguration eventuell unsicher ist, oder zumindest für unsere Zwecke nicht sicher genug ;-).

### 6.1 glftpd

glftpd trägt sich nach der Installation via `installgl.sh` standardmäßig mit den Parametern „-l -o -i“ ein, sodass das Erstellen, Löschen, Nuken und Unnuken von Verzeichnissen sowie Logins und Logouts in `glftpd.log` und Transfers in `xferlog` aufgezeichnet werden. Prinzipiell ist das Loggen von Transfers für manche Scripts notwendig, sodass man das nicht unbedingt deaktivieren sollte. Allerdings ist „-l“ nicht unbedingt nötig und das Hinzufügen von „-X“ sicher keine schlechte Idee (dann werden *gar keine* IPs mehr gespeichert in Logfiles).

Die Option, glftpd in ein separates Verzeichnis zu installieren, dem keine Zugriffsrechte für Benutzer gegeben werden, die nicht in der glftpd-Gruppe sind, sollte genutzt werden (so genanntes „jail“). Im Optimalfall sollte zwar ohnehin keiner am System angemeldet sein, außer der Eigentümer, aber nun gut...

Außerdem sollte das `ftp-data`-Verzeichnis auf jeden Fall auf einer verschlüsselten Partition liegen, da dort die Logdateien sowie die Benutzerdateien gespeichert werden.

### 6.2 Linux-Server

Bei jedem Login auf einem Linuxserver hinterlässt man Spuren. Wenn man nun einen Server verwaltet, der riskante Daten enthält, sollte man eventuell seine Loginspuren verwischen.

Die Logins werden in „`/var/log/wtmp`“ gespeichert, ein „`ls /var/log/wtmp*`“ zeigt auch ältere Dateien, die durch `logrotate` verschoben wurden. Außerdem werden alle abgesetzten Befehle in einer `history`-Datei gespeichert, die je nach Shell anders heißt (bei der `bash` heißt sie `~/.bash_history`), ein „`ls -a ~/*history*`“ sollte eine entsprechende Datei zeigen. Mit dem Befehl `touch` kann man die Datei bequem leeren, zum sicheren Löschen siehe Seite 13.

### 6.3 IRC-Clients

Bei IRC-Clients ist weniger der Client selbst, sondern vielmehr sind die Convenience-Funktionen das Problem. Viele Benutzer richten ihre IRC-Clients so ein, dass sie sich automatisch mit dem Bouncer/IRC-Server verbinden und sich in die eigentlich privaten Channels einladen (über Bots). Wenn man den Client nicht auf einer verschlüsselten Partition abgelegt hat, ist das natürlich ein gefundenes Fressen für Ermittler. Bei einigen Hausdurchsuchungen wurde berichtet, dass die Ermittler das IRC noch einige Zeit

überwacht haben, da der/die Verdächtige zum Zeitpunkt der Durchsuchung ihren Client offen hatten.

Aufpassen sollte man auch bei Clients oder Scripts, die Logging standardmäßig aktiviert haben. Auch hier gilt jedoch: Auf einer verschlüsselten Partition ist das eigentlich kein Problem.

## **6.4 FlashFXP**

FlashFXP bietet die Möglichkeit, ein Startpasswort zu setzen, mit dem dann die `Sites.dat` und die anderen Dateien, die eigenen Daten enthalten, verschlüsselt werden. Laut dem FlashFXP-Entwickler basiert dieses Verfahren auf Blowfish, was prinzipiell sicher ist – da FlashFXP jedoch Closedsource ist, muss man hier dem Entwickler vertrauen, dass er die Implementierung fehlerfrei durchgeführt hat.

Das Aktivieren des Startpassworts (nicht nur bei) FlashFXP ist also nicht schlecht, man sollte sich jedoch, da man nicht den Quelltext der Anwendung hat, nicht darauf verlassen, also FlashFXP zusätzlich auf eine verschlüsselte Partition legen.

## 7 Sicherheit im echten Leben

Je weniger Menschen von deinen Tätigkeiten wissen, desto weniger können ein Risiko für dich darstellen. Zudem wird es – in den meisten Fällen – deine Familienmitglieder und enge Freunde ohnehin nicht interessieren, warum du nun im Einzelnen kaum Zeit für andere Dinge als den blöden Computer hast ;-).

### 7.1 Kontakt

Ein Interessenkonflikt tritt natürlich dann auf, wenn du dich mit anderen Leuten im echten Leben austauschen willst, die du Online kennen gelernt hast. Vielleicht kann man sich darauf einigen, hier lieber über „normale“ Sachen zu reden anstatt über Computer, Kopien, Programmieren und so weiter, dann steht einem Treffen in der Stammkneipe nichts im Wege. Was du privat in deinen eigenen vier Wänden machst, ist selbstverständlich deine Sache. Wenn die Wahrscheinlichkeit sehr hoch ist, *nicht* verwandt zu sein, könnte man hier ausnahmsweise mal offen reden... ;-)

Von Telefonaten ist dringend abzuraten, da Telefone sehr leicht überwacht werden können und auch effektiv werden<sup>10</sup>. Benutz doch stattdessen **Skype** und telefoniere damit verschlüsselt über das Internet.

### 7.2 Keine unnötigen Blicke riskieren...

Des Weiteren solltest du nicht zusammen mit anderen Leuten, die dir über die Schulter schauen, im IRC unterwegs sein oder Texte, die vertrauliche Informationen beinhalten, verfassen oder bearbeiten. Wenn du also vom Computer weggehst, minimiere deinen IRC-Client und dein E-Mail-Fenster. Viel praktischer: ein Desktopmanager. Diese Programme bieten virtuelle Arbeitsflächen (bei Linux bei fast allen Distributionen standardmäßig dabei). Man kann dann zum Beispiel in einer Arbeitsfläche seine Chat-, Mail- und Messagingprogramme benutzen und in der anderen Arbeitsfläche einen Webbrowser, MP3-Player etc...

Eine gute Sache ist es auch, den Drang zu unterdrücken, immer und von überall aus in den Stamm-IRC-Channels sein zu wollen (Schule, Arbeitsplatz, Freunde...) – man kennt die Netzstruktur in fremden Räumlichkeiten meist nicht genau und sollte sich daher auf das IRCen von Zuhause aus beschränken.

**Vorschlag:** Remote-Administrations-Programme wie VNC, **freenx** oder **Apple Remote Desktop** erlauben den Zugriff durch verschlüsselte Verbindungen auf den eigenen Rechner von beliebigen Orten.

**Nachteil:** Wenn vor Ort jemand den Monitor eingeschaltet hat, kann er genau sehen, was du machst. Du musst dich also immer beobachtet fühlen...

---

<sup>10</sup><http://www.heise.de/newsticker/meldung/36563>

## 8 Hausdurchsuchungen

### 8.1 Was darf die Polizei / Welche Rechte habe ich?

**Hinweis:** Die Punkte beziehen sich auf deutsches Recht, in Österreich sieht es aber fast genauso aus.

- Ihr habt das Recht, bei der Durchsuchung anwesend zu sein (§ 106 Abs. 1 Satz 1 StPO). Wenn der Inhaber der zu durchsuchenden Räume nicht zu Hause ist, ist sein Vertreter oder ein erwachsener Angehöriger, Hausgenosse oder Nachbar hinzuzuziehen.
- Sollte man gegenüber den Beamten ausfällig werden, droht ein Verfahren wegen Widerstandes gegen Vollstreckungsbeamte (§ 113 StGB).
- Die Rechtmäßigkeit der Beschlagnahme einzelner Gegenstände kann im Nachhinein durch einen Rechtsanwalt geklärt werden, nicht während der Durchsuchung.
- Alles was man sagt, kann gegen einen verwendet werden – auch während der Durchsuchung. Bis auf Angaben zur Identität sollte man also keine Angaben machen.
- Wenn es um Unterlagen geht, kann es schnell passieren, dass die ermittelnden Beamten in Ihren Papieren herumwühlen. Gemäß § 110 Abs. 1 StPO ist die Durchsicht von Papieren nur dem Staatsanwalt selbst erlaubt.
- Von beschlagnahmten Gegenständen oder Papieren ist gemäß § 107 StPO ein Protokoll zu erstellen, aus dem sich die mitgenommenen Gegenstände ergeben.

### 8.2 Wie verhalte ich mich?

Prinzipiell sollte man eher kooperativ handeln, den durchsuchenden Polizisten also (zumindest einen) Rechner zeigen und eventuell beim Abbauen helfen. Vielleicht könnt ihr sie ja davon überzeugen, dass Komponenten wie der teure Bildschirm oder dieser komische Kasten zwischen Modem und Switch (externe Festplatte/NAS ;-)) unnötig sind... ;-).

Damit ihr eure Rechner relativ schnell wiederbekommt, solltet ihr euch alle Dokumente genau durchlesen, die ihr unterschreiben sollt und alle Dokumente, bei denen ihr euch mit irgendetwas einverstanden erklären sollt, am besten *nicht unterschreiben* – schließlich muss hier ohnehin ein Missverständnis vorliegen, ihr habt ja keinerlei Raubkopien oder ähnliches :-).

Wenn euch die Ermittler nach Passwörtern fragen, rückt sie ruhig heraus – aber nur die, für die erste Stufe der Verschlüsselung (vorausgesetzt, dass ihr auf Plausible Deniability geachtet habt und hidden Volumes erstellt habt). Falls ihr Passwörter nicht herausgeben wollt, gebt ihnen doch ein falsches – der jeweilige Administrator kann das Passwort ja in der Zwischenzeit geändert haben. Außerdem steht man bei so einer Durchsuchung natürlich auch unter Schock und kann so lange Kombinationen schon mal vergessen ;-).

Schaut euch unbedingt im Anhang die Dokumente zu Hausdurchsuchungen an, auch wenn sie schon etwas älter sind, sind sie durchaus lesenswert.

## 9 Anhang

Hier findest du Links und Verweise zu verwandten Themen und/oder im Text erwähnten Programmen.

### IRC-Clients

- mIRC (Windows): <http://www.mirc.co.uk/>
- irssi (Windows, Linux, Mac OS X): <http://www.irssi.org/>
- XChat (Windows, Linux, Mac OS X): <http://www.xchat.org/>

### IRC-Verschlüsselungs-Plugins

- FiSH (für mIRC, irssi und XChat): <http://fish.sekure.us/>
- Mirccryption (für mIRC, irssi, XChat und eggdrop/psyBNC): <http://www.donationcoder.com/Software/Mouser/mirccryption/>

### IRC-Netze

- EFNet (Eris Free Network, keine Nickname-/Channelregistrierung, kein SSL): <http://www.efnet.org/>, <http://www.efnet.net/>
- Linknet (keine Nickname-/Channelregistrierung, SSL netzwerkweit): <http://www.link-net.org/>

### Daten vernichten

- Shred (Linux): <http://www.gnu.org/software/coreutils/>
- Wipe (Linux): <http://wipe.sourceforge.net>
- SRM, SFILL, SSWAP, SMEM (Linux): <http://www.thc.org/release.php?q=delete>

### Hausdurchsuchungen

- thc.org: Hausdurchsuchungen: <http://www.thc.org/papers/HAUSDURC.TXT>
- <http://unglaublichkeiten.com/unglaublichkeiten/htmlphp/medienhausdurchsuchung.html>
- <http://haus.durchsuchungen.de/>
- <http://www.mindpower.com/wps/bsa.htm>
- <http://www.internetrecht-rostock.de/hausdurchsuchung.htm>

### Quellen

- SystemrescueCD Manual: <http://www.sysresccd.org/Manual>
- Wikipedia: Security through obscurity: [http://de.wikipedia.org/wiki/Security\\_through\\_obscurity](http://de.wikipedia.org/wiki/Security_through_obscurity)
- [http://www.unizh.ch/RZU/services/pc-mac-support/mac-sys/security/osx\\_hardening.pdf](http://www.unizh.ch/RZU/services/pc-mac-support/mac-sys/security/osx_hardening.pdf)
- <http://www.glftpd.com/files/docs/glftpd.docs>