Dokumentation zu Quiddi-github

Quiddi

Inhaltsverzeichnis

1. Git

1.0.1 neues Projekt

In den Ordner im Terminal reinnavigieren und mit

```
git init
```

das Git-Projekt initialisieren. Danach:

```
git add *.*
git add README
git commit –m 'Projekt neu angelegt'
```

1.0.2 GitHub Upload

Danach kann auf der GitHub Homepage ein neues Projekt angelegt werden. Um dies pushen zu können muss man

```
git remote add origin https://github.com/quiddi-github/linux.bibel.git
```

origin adden. Danach kann man das Projekt auf origin pushen, mit:

```
git push –u origin master
```

Heirmit wird es auf dem Master-Branch gepushed. Zuvor müssen aber noch alle Dateien reingeladen werden mit:

```
git add .
```

und ein Kommentar angelegt werden mit

```
git commit –m 'Projekt neu angelegt'
```

Die remote Reposities können mit

```
git remote show
```

bzw. wenn ein remote Reposity namens origin vorliegt, dies mit

```
git remote show origin
```

anzeigen lassen.

2. Verschlüsseln von Passwörtern

2.1 GPG Schlüssel importieren exportieren erstellen

GPG verwaltet unter Linux die Schlüssel.
Man kann sich die **öffentlichen** Schlüssel anzeigen lassen mit:

```
gpg --list-keys
```

oder die **privaten** mit:

```
gpg --list-private-keys
```

2.1.1 Export Keys

Zum exportieren bedient man sich folgendem Befehl:
öffentlich

```
gpg -a --output public.gpg --export <Key-ID>
```

für den **privaten**:

```
gpg --export-secret-keys -a <Key-ID> > secret.asc
```

2.1.2 Import Keys

Die Schlüssel müssen nun auf dem anderen PC importiert werden, Dies passiert für den **öffentlichen** Schlüsssel mit

```
gpg --import public.gpg
```

und für den **privaten** Schlüssel mit

```
gpg --allow-secret-key-import --import secret.asc
```

Um mit dem öffentlischen Schlüssel abreiten zu können, muss diesem noch vertraut werdem. Andernfalls erhält man z.B. in dem Programm pass eine Fehlermeldung. Zum vertrauen des Schlüssels geht man wie folgt vor:

```
gpg --edit-key <Key-ID>
```

Danach öffent sich ein seperates Programm im Terminal. Hier gibt man „trust“ ein und bestätigt dies mit „5 “ für „Ich vertraue ihm absolut“. Danach kann man es wieder schließen. Danach kann mit dem öffentlichen Schlüssel gearbeitet werden. Der private Schlüssel wird nur zum entschlüsseln gebraucht und ihm muss daher nicht vertraut werden.

2.1.3 neues Key-Paar erstellen

Ein neues GPG-Key-Paar, also der öffentliche- mit einem provatem Schlüssel, wird erzeugt mit

```
gpg --full-generate-key
```

Danach muss „1“ gedrückt werden um einen RSA-Key zu erhalten. Danach wird nach der bit-Größe gefragt. Hier kann z.B. „1024“ eingegeben werden. Danach wird nach dem Ablaufdatum gefragt. Ist das Ablaufdatum erreicht so kann mit dem öffentlichen Schlüssel nichts neues mehr verschlüsselt werden. Es können aber noch 100 Jahre nach dem Ablaufdatum vorhandene verschlüsselte Dateien entschlüsselt werden. Er frägt dann weiter ob die Daten korrekkt sind, Nach einem Namen, e-Mail, Kommentar. Danach wird nach einem Passphrase gefragt. Der Passphrase ist notwendig um den Key nachher benutzen zu können. Ohne den Passphrase zu wissen kann er später nicht benutzt werden. Danach ist der Key erstellt.

2.2 pass

Das Programm pass verschlüsselt, mit dem zuvor defineirten öffentlichen Schlüssel, Text, und entschlüsselt dies mit dem dazugehörigen private Key wieder. Hat man ein Schlüsselpaar, bestehend aus öffentlichem- und privatem Schlüssel erstellt so kann man dieses mit

```
pass init <Key-ID>
```

für pass defineren. Achtung, wurde der Key zuvor importiert, so muss dem öffentlichen Key erst vertraut werden, ansonsten kann ein verschlüsseln mit pass nicht stattfinden. Sollte man nun alte verschlüsselte Dateien haben, so können diese nun in „*/home/USER/.password-store*“ kopiert werden. Das entschlüsseln sollte nun mit

```
pass <verschluesselte Datei>
```

funktionieren. Mit „pass“ werden alle verschlüsselten Inhalte angezeigt. Mit „pass insert <verschlüsselte Datei>“ kann dies dann wieder im Klartext dargestellt werden. Mit „pass rm <verschlüsselte Datei>“ kann das Teil wieder gelöscht werden. Mehrere Zeilen können mit

```
pass insert -m <verschluesselte Datei>
```

angelegt werden.

3. Microsoft Office 2013

Vorraussetzungen:

- Office 2013 als 32-bit Version
- Die Pakete: wine winetricks mono-devel samba-winbind

3.1 Wine installieren

Wine wird in Debian wie folgt installiert:
32-Bit Architektur hinzufügen:

```
sudo dpkg --add-architecture i386
```

Den entsprechenden Key hinzufügen

```
wget -nc https://dl.winehq.org/wine-builds/Release.key
sudo apt-key add Release.key
```

Danach kann in die */etc/apt/sources.list* folgendes hinzugefügt werden:

```
deb https://dl.winehq.org/wine-builds/debian/ DISTRO main
```

DISTRO ist hierbei urch das Debian System zu erssetzen. Danach das ganze updaten:

```
sudo apt-get update
```

Nun die StableVersion von Wine installieren:

```
sudo apt-get install --install-recommends winehq-stable
```

3.2 andere Pakete

Es fehlern nun noch
winetricks mono-devel samba-winbind
Diese bitte noch installieren. Am einfachsten mit

```
sudo apt-get install winetricks mono-devel winbind
```

Das war soweit die Vorbereitung. Nun kann Wine konfiguriert werden

3.3 Konfiguration Wine

Für Office 2013 einen neuen Ordner anlegen:

```
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 winecfg
```

Es muss Windows 7 verwendet werden. Nun noch folgendes nachinstallieren

```
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 winetricks corefonts
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 winetricks msxml4
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 winetricks msxml6
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 winetricks riched30
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 winetricks vb6run
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 winetricks d3dx11_42
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 winetricks d3dx11_43
```

Es soll nun in der Konfig bei Bibliotheken unter Bestehende ein Eintrag „*riched20 (Native, Builtin)“ sein. Die kOnfig kann mit

```
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 winecfg
```

aufgerufen werden. Nun kann Office 2013 installiert werden

```
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 wine /PATH_TO_MOUNTPOINT/setup.e
```

Bei mir bleib die Installation irgendwann hängen, starten tut mein Word,... trotzdem. Word kann nun gestartet werden mit

```
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 wine "C:\\Program Files\\Microso
```

Hierbei wird es wahrscheinlich zu einem schwarzen Bildschirm kommen. Dies kann man folgendermaßen gerichtet werden. Wir machen zunächst die Regedit auf

```
WINEPREFIX=~/.wine/Office2013 WINEARCH=win32 wine regedit
```

Unter „HKEY_CURRENT_USER{}Software{}Wine“ legen wir den neuen Schlüssel „Direct3D“ an. Danach wird ein neuer DWORD Wert mit dem Namen „MaxVersionGL“ angelegt. Als Hexwert schreiben wir hier „30002“ hinein. Sollte dies immer noch nicht funktionieren soo wird noch unter „HKEY_CURRENT_USER{}Software{}Wine“ ein Schlüssel mit „Direct2D“ angelegt. Auch hier muss wieder ein DWORD angelegt werden mit dem Namen „max_version_factory“. Hier sollte dann der Wert „0“ drinnenstehen. Das war’s