Steffen Probst

Alle hier erstellten Inhalt dieser Seiten und deren Unterseite stelle ich unter der GPLv3. Ihr dürft die Inhalte gerne im Sinne dieser Lizenz weiterverwenden.
Alle auf dieser Seite und Unterseiten veröffentlichten Bilder stelle ich unter der CC BY-NC-ND[1)] Lizenz, wenn dieses nicht anders angegeben wurde. Ihr dürft die Bilder im Sinne der Lizenz verwenden.

Blog

Firefox die 2.: /etc/firefox/policies/policies.json

Ich habe jetzt eine Standardkonfiguration für den Firefox zu bauen. Vorteil ist, dass man diverse Funktionen sich so einstellen kann, das nach einer Neueinrichtung des Systems, gleich wieder alle Funktionen zur Verfügung stehen und, zum Beispiel, die Telemetriedaten deaktiviert sind. Unabhängig davon muss trotzdem, wenn es erwünscht ist, das alte Profile von `$HOME/.mozilla` umgezogen werden, wenn nicht die Synchronisierung mit Firefox[2)] aktiviert wurde. Das muss jeder selbst entscheiden, inwieweit er der Mozilla Fondation seine Daten anvertraut und soll hier nicht Thema sein. Solltet ihr die Synchronisierungsfunktion verwenden, müsst ihr euch ggf. die ersten zwei Zeilen `policies` auskommentieren oder löschen.

Ich habe mir die `policies.json` unter nixOS bauen lassen, da es hier gut Vorlagen gibt. Dafür habe ich mir ein flake erstellt. Die entsprechende Konfiguration findest du hier → auf Codeberg.

Ein Beispiel wie die `policies.json` aussehen kann, siehst du hier.

policies.json

```
{
  "policies": {
    "DisableAccounts": true,
    "DisableFirefoxAccounts": true,
    "DisableFirefoxScreenshots": true,
    "DisableFirefoxStudies": true,
    "DisablePocket": true,
    "DisableTelemetry": true,
    "DisplayBookmarksToolbar": "never",
    "DisplayMenuBar": "default-off",
    "DontCheckDefaultBrowser": true,
    "EnableTrackingProtection": {
      "Cryptomining": true,
      "Fingerprinting": true,
      "Locked": true,
      "Value": true
    },
    "ExtensionSettings": {
      "floccus@handmadeideas.org": {
        "install_url":
"https://addons.mozilla.org/firefox/downloads/latest/floccus/latest.xpi
```

```
",
        "installation_mode": "force_installed"
      },
      "idcac-pub@guus.ninja": {
        "install_url":
"https://addons.mozilla.org/firefox/downloads/latest/istilldontcareabou
tcookies/latest.xpi",
        "installation_mode": "force_installed"
      },
      "jid1-DNc5AXAyVmgNjQ@jetpack": {
        "install_url":
"https://addons.mozilla.org/firefox/downloads/latest/fxqrl/latest.xpi",
        "installation_mode": "force_installed"
      },
      "langpack-de@firefox.mozilla.org": {
        "install_url":
"https://releases.mozilla.org/pub/firefox/releases/123.0.1/linux-x86_64
/xpi/de.xpi",
        "installation_mode": "normal_installed"
      },
      "langpack-en-US@firefox.mozilla.org": {
        "install_url":
"https://releases.mozilla.org/pub/firefox/releases/123.0.1/linux-x86_64
/xpi/en-US.xpi",
        "installation_mode": "normal_installed"
      },
      "languagetool-webextension@languagetool.org": {
        "install_url":
"https://addons.mozilla.org/firefox/downloads/latest/languagetool/lates
t.xpi",
        "installation_mode": "force_installed"
      },
      "ncpasswords@mdns.eu": {
        "install_url":
"https://addons.mozilla.org/firefox/downloads/latest/nextcloud_password
s/latest.xpi",
        "installation_mode": "force_installed"
      },
      "qwantcomforfirefox@jetpack": {
        "install_url":
"https://addons.mozilla.org/firefox/downloads/file/3996872/qwantcom_for
_firefox-7.0.4.9.xpi",
        "installation_mode": "force_installed"
      },
      "uBlock0@raymondhill.net": {
        "install_url":
"https://addons.mozilla.org/firefox/downloads/latest/ublock-origin/late
st.xpi",
        "installation_mode": "force_installed"
      },
      "{e4a12b8a-ab12-449a-b70e-4f54ccaf235e}": {
```

```
        "install_url":
"https://addons.mozilla.org/firefox/downloads/latest/proxy_switcher_and
_manager/latest.xpi",
        "installation_mode": "force_installed"
      }
    },
    "OverrideFirstRunPage": "",
    "OverridePostUpdatePage": "",
    "Preferences": {
      "browser.contentblocking.category": {
        "Status": "locked",
        "Value": "strict"
      },
      "browser.formfill.enable": {
        "Status": "locked",
        "Value": false
      },
      "browser.newtabpage.activity-stream.feeds.section.topstories": {
        "Status": "locked",
        "Value": false
      },
      "browser.newtabpage.activity-stream.feeds.snippets": {
        "Status": "locked",
        "Value": false
      },
      "browser.newtabpage.activity-
stream.section.highlights.includeBookmarks": {
        "Status": "locked",
        "Value": false
      },
      "browser.newtabpage.activity-
stream.section.highlights.includeDownloads": {
        "Status": "locked",
        "Value": false
      },
      "browser.newtabpage.activity-
stream.section.highlights.includePocket": {
        "Status": "locked",
        "Value": false
      },
      "browser.newtabpage.activity-
stream.section.highlights.includeVisited": {
        "Status": "locked",
        "Value": false
      },
      "browser.newtabpage.activity-stream.showSponsored": {
        "Status": "locked",
        "Value": false
      },
      "browser.newtabpage.activity-stream.showSponsoredTopSites": {
        "Status": "locked",
```

```
        "Value": false
      },
      "browser.newtabpage.activity-stream.system.showSponsored": {
        "Status": "locked",
        "Value": false
      },
      "browser.search.suggest.enabled": {
        "Status": "locked",
        "Value": false
      },
      "browser.search.suggest.enabled.private": {
        "Status": "locked",
        "Value": false
      },
      "browser.topsites.contile.enabled": {
        "Status": "locked",
        "Value": false
      },
      "browser.urlbar.showSearchSuggestionsFirst": {
        "Status": "locked",
        "Value": false
      },
      "browser.urlbar.suggest.searches": {
        "Status": "locked",
        "Value": false
      },
      "extensions.pocket.enabled": {
        "Status": "locked",
        "Value": false
      },
      "extensions.screenshots.disabled": {
        "Status": "locked",
        "Value": true
      }
    },
    "SearchBar": "unified"
  }
}
```

Die ganzen Optionsschalter sind gut dokumentiert und lassen sich über die einschlägigen Seiten herausfinden. Optional sei auch hier wieder auf die `about:config` vervwiesen.

2024/04/07 18:02 · SProbst

Firefox: default Einstellungen global (Win)

Ab und zu kommt man in Verlegenheit, ein Windows für einen anderen Nutzer zu konfigurieren. Da möchte ich, dass der Firefox entsprechende Standardeinstellungen[3)] hat, die nicht mit dem Firefox ausgeliefert werden. Hierfür lässt sich eine entsprechende Einstellung unter dem Firefox vornehmen.

Im Hauptverzeichnis der Firefox Installation unter Windows legt man `local-settings.js` in `defaults/pref` an. Die Javascriptdatei verweist dann auf die `mozilla.cfg`. Diese ist auch eine Javascriptdatei mit den Einstellungen. In der folgenden Zeile, in der `local-settings.js` gibt man dann noch mit, dass die Datei n PlainText zu lesen ist.

local-settings.js

```
/* This Source Code Form is subject to the terms of the Mozilla Public
 * License, v. 2.0. If a copy of the MPL was not distributed with this
 * file, You can obtain one at http://mozilla.org/MPL/2.0/. */
//
// This pref is in its own file for complex reasons. See the comment in
// browser/app/Makefile.in, bug 756325, and bug 1431342 for details. Do
not add
// other prefs to this file.

pref("general.config.filename", "mozilla.cfg");
pref("general.config.obscure_value", 0);
```

Wichtig ist, dass in der `mozilla.cfg` die erste Zeile nicht interpretiert wird und diese mit einem Kommentar aufgefüllt wird. Die `mozilla.cfg` muss im Hauptverzeichnis der Windows-Installation liegen.

Unter Linux muss entsprechend in der jeweiligen Distribution geschaut werden, wie die Konfigurationsdateien abgelegt werden. [4] Alternativ kann ein `/etc/firefox/policies.json`[5] angelegt werden, wo die entsprechenden globalen Einstellungen für Firefox unter Linux abgelegt werden. Das sollte unter allen Linux Distributionen funktionieren[6].

Meine Konfiguration hier setzt die Startseite auf die Suchmaschine Qwant. Zusätzlich wird das überprüfen, ob der Firefox der Standardbrowser ist, deaktiviert und das Caching vom Firefox deaktivert. Dabei habe ich mich an den Einstellungsoptionen aus den Fussnoten bedient und in der about:config vom Firefox nachgeschaut. In jeden Fall sollte die Datei mit einem Linux Editor erzeugt werden, dass der Firefox entsprechend LF (LineFeed) Umbrüche seit Version 60 erwartet. Hier kann es beim Erstellen mit dem **Editor(notepad)** von Windows probleme geben, wenn jemand auf die kommt die Dateien unter Windows zu erstellen. Aber wer kommt schon auf solche Ideen. ;)

mozilla.cfg

```
// IMPORTANT: Start your code on the 2nd line
pref("browser.startup.homepage", "https://www.qwant.com");
pref("browser.shell.checkDefaultBrowser", false);
pref("browser.cache.disk.enable", false);
pref("browser.cache.disk.smart_size.enabled", false);
pref("browser.cache.disk.capacity", 0);
```

2024/04/02 19:48 · SProbst
firefox, einstellung, standard

Debian - apt pinning

In diesem kleinen Beitrag arbeite ich das LUG Thema vom 06.03.2024 auf. Wir hatten dort das Thema Pinning von deb Paketen unter Debian. Eine weiterführende Dokumentation ist auch direkt auf der Debian Seite [7)] zu finden.

Was ist apt

`apt` oder auch `apt-get` ist die Paketverwaltung unter Debian basierte Linux Distribution [8)]

Was ist das Ziel?

Mit `apt` pinning möchte man in der Paketverwaltung bestimmte Programme bei der Installation priorisieren.
Beispiel:
Du möchtest als Standarddistribution Debian 12 bookworm verwenden, aber bestimmt Pakete aus einer anderen Version von Debian (oder einer anderen Distribution) dazu installieren.

Wichtig: Wenn du Pakete aus anderen Distributionen oder Versionen mischst, kann es zu ungewollten Seiteneffekte kommen. Daher gebe ich auf die Anleitung keine Funktionsgarantie und jeder führt diese auf eigenes Risiko durch.

Wie gehe ich vor?

- Erstelle unter `/etc/apt/source.list.d` eine `.list` mit dem entsprechenden Repo, was du verwenden willst. → Als Vorlage kann die `/etc/apt/source.list` dienen. → Kopiere diese in den Ordner `/etc/apt/source.list.d/` und bennen diese vorzugsweise um. Beispiel: `cp /etc/apt/source.list /etc/apt/source.list.d/foo.list` → Ändere die Einträge in der Datei mit einem Editor deiner Wahl auf das passende Repo ab.
- Lege unter `/etc/apt/preferences.d/` eine Datei an, in der du das Pinning konfigurierst. → Ich gebe der Datei immer voran gestellt eine Nummer und dann, was diese enthalten soll. → Beispiel: `99-debian`
- Öffne die Datei mit dem Editor Deiner Wahl und ändere die Einträge wie folgt.

99-debian

```
Package: *
Pin: release a=unstable
Pin-Priority: 500

Package: *
Pin: release a=stable
Pin-Priority: 900

Package: firefox*
Pin: release a=unstable
Pin-Priority: 1000
```

```
Package: keepass*
Pin: release a=unstable
Pin-Priority: 1000

Package: flameshot*
Pin: release a=unstable
Pin-Priority: 1000

Package: filius*
Pin: release a=unstable
Pin-Priority: 1000

Package: remmina*
Pin: release a=unstable
Pin-Priority: 1000

Package: libreoffi*
Pin: release a=unstable
Pin-Priority: 1000

Package: podman*
Pin: release a=unstable
Pin-Priority: 1000

Package: docker*
Pin: release a=unstable
Pin-Priority: 1000

Package: distrobox*
Pin: release a=unstable
Pin-Priority: 1000

Package: remmina*
Pin: release a=unstable
Pin-Priority: 1000

Package: qflipper*
Pin: release a=unstable
Pin-Priority: 1000

Package: mfcuk
Pin: release a=unstable
Pin-Priority: 1000

Package: mfoc
Pin: release a=unstable
Pin-Priority: 1000

Package: fritzing*
Pin: release a=unstable
```

```
Pin-Priority: 1000

Package: arduino*
Pin: release a=unstable
Pin-Priority: 1000
```

- Die ersten dreier Block setzen die Priorität für `sid/unstable` auf den Wert 500. Das bedeutet, dass die Pakete aus diesem Repo installiert werden, wenn Sie benötigt werden.
- Der zweite dreier Block setzte `stable` auf die Priorität 900. Dadurch liegt die Priorität für stable höher als 500 von `unstable` und dem Paket aus dem stabilen Zweck vorrangig installiert, wenn die dort Pakete mit einer niedrigeren Versionsnummer vorhanden sind.
- Die nachfolgenden Blöcke priorisieren die Pakete aus `unstable` mit einer 1000. Die Priorität 1000 sagt, dass die Pakete explizit aus `unstable` installiert werden sollen und die nötigen Abhängigkeiten zwingend nach gezogen werden sollen. Dabei prüft apt ob er Abhängigkeiten aus dem `stable` und `unstable` Zweig der Repositorys auflösen kann.

Schlusswort

Wenn jemand einen Fehler findet, bitte sendet mir diesen zu. Ich korrigiere dann den Eintrag. Falls ich noch auf etwas eingehen soll, dann bitte auch eine Info. Ansonsten findet man sehr gute Anleitungen im Netz dazu, Ich hoffe ich konnte etwas Licht ins Dunkel bringen.

2024/03/07 07:33 · SProbst
debian, linux, apt, pinning

openwrt in docker -> Das geht?

Ja. Das geht. Sogar ziemlich gut!

Was benötigt wird, ist einfach ein `rootfs.tar.gz` von openwrt. Aktuell sind die für **x86/amd64** und **arm** verfügbar. Ab Version **23.5.x** steht in dem Sinne kein **armvirt** mehr zu Verfügung. Aber der aufmerksame Leser der Release Notes findet den Hinweis das, dass entsprechende Release in **armsr/armv7** und **armsr/armv8** gewandert ist.

Als erste lädt man sich entsprechend der Architektur das `rootfs.tar.gz` herunter. Idealerweise mit einem Download wie `wget` oder `aria2c`.

Das Dockerfile sieht wie folgt aus:

Dockerfile

```
FROM scratch

ADD openwrt-23.05.2-armsr-armv8-rootfs.tar.gz /

RUN     mkdir -p /var/lock/ && opkg update && \
        mkdir -p /var/run

EXPOSE 80/tcp
```

```
USER root

CMD ["/sbin/init"]
```

Der Rest sollte plausibel sein und funktioniert wie bei anderen Dockerfiles/-containern. Das Ergebnis ist ein verdammt schmaler Container.

2023/12/16 17:50 · SProbst
docker, dockerfile, shell, bash, openwrt

post-checkout mit git

Ich pflege meinen `.ssh` Ordner mit git. Im Klartext, ich habe mehr **Branches** für unterschiedliche Szenarien. Der Hintergrund: egal, weil es geht. Dabei habe ich immer das Problem gehabt, dass die Keys nicht die korrekten Zugriffsberechtigungen hatten (`0644` auf Dateien). Nach etwas Recherche stellt sich heraus, dass im git Repo, es möglicht ist, Script nach bestimmten Positionen/Stadien automatisch ausführen zu lassen. In meinem Fall ist das `post-checkout` Script unter `$REPODIR/.git/hooks/` spannend. Kurzerhand dort eine Shellscript mit dem Namen angelegt. Ein passenden Find/While Befehl gebaut und schwups, sind die Berechtigung zukünftig korrekt. Hier das Beispiel:

post-checkout

```
#!/usr/bin/bash

#find pubkey and set file.key permissions
find $HOME/.ssh -name '*.pub' -print0  | while IFS= read -r -d '' line;
do chmod 600 $HOME/.$(echo $line | cut -f 2 -d ".") || true; done
```

2023/12/16 17:07 · SProbst
ssh, pub, key, shell, bash, git

Ältere Einträge >>

Index

L

- Linksammlung

[1]
Namensnennung - Nicht-kommerziell - Keine Bearbeitung
[2]
Firefox Cloud/Konto → https://www.mozilla.org/de/account/
[3]
https://support.mozilla.org/en-US/kb/customizing-firefox-using-autoconfig
[4]
https://linuxconfig.org/how-to-customize-firefox-using-the-policies-json-file
[5]
https://mozilla.github.io/policy-templates/
[6]
von mir noch nicht getestet → wird aber bei Gelegenheit nachgeholt
[7]
https://wiki.debian.org/AptConfiguration
[8]
Wie Linux Mint der Ubuntu zum Beispiel.

From:
http://wiki.lug-wr.de/wiki/ - **Wiki der Linux User Group Wernigerode**

Permanent link:
http://wiki.lug-wr.de/wiki/doku.php?id=user:sprobst:start&rev=1591814339

Last update: **2020/06/10 20:38**